CL-DPS: A Contrastive Learning Approach to Blind Inverse Problem Solving via Diffusion Posterior Sampling

Linfeng Ye
University of Toronto
27 King's College Cir, Toronto
linfeng.ye@mail.utoronto.ca

Pallavi Ferrao
University of Toronto
27 King's College Cir, Toronto
pallavi.ferrao@mail.utoronto.ca

Figure 1. Results of blind rotation deblurring, a challenging **non-linear** inverse problem: (a) ground truth image, (b) rotation blurred measurement, and restored images using (c) BlindDPS [8], (d) FastEM [23], (e) GibbsDDRM [27], and (f) CL-DPS (ours). Notably, all methods fail catastrophically except for CL-DPS.

Abstract

*Diffusion models (DMs) have recently shown great promise in solving inverse problems. While most research in this area addresses non-blind inverse problems, where the measurement operator is assumed to be known, real-world applications frequently involve blind inverse problems with unknown measurements. Existing DM-based methods for blind inverse problems are limited, primarily addressing only linear measurements and thus lacking applicability to real-life scenarios that often involve non-linear operations. To overcome these limitations, we propose **CL-DPS**, a novel approach based on **c**ontrastive **l**earning for solving blind inverse problems via **d**iffusion **p**osterior **s**ampling. In CL-DPS, we first train an auxiliary deep neural network (DNN) offline using a modified version of MoCo [16], a contrastive learning technique. This auxiliary DNN serves as a likelihood estimator, enabling estimation of $p(\boldsymbol{y}|\boldsymbol{x})$ without prior knowledge of the measurement operator, thereby adjusting the reverse path of the diffusion process for inverse problem solving. Additionally, we introduce an overlapped patch-wise inference method to improve the accuracy of likelihood estimation. Extensive qualitative and quantitative experiments demonstrate that CL-DPS effectively addresses non-linear inverse problems, such as rotational deblurring, which previous methods could not solve. Code available: https://github.com/cldps/cldps.*

1. Introduction

Inverse problems are pervasive across many fields, with significant applications in areas such as medical imaging [20, 26], computational photography [28, 38], and seismic imaging in geophysics [19, 45], among others. Particularly, the objective of the inverse problems is to recover the original signal $\boldsymbol{x}$ from the corrupted measurement $\boldsymbol{y}$ which is generated by the forward operation/measurement $\mathcal{A}_{\psi}(\cdot)$.

Inverse problems are typically divided into two major categories based on the availability of $\mathcal{A}_{\psi}$: non-blind and blind inverse problems. Non-blind inverse problems assume that $\mathcal{A}_{\psi}$ is known. In contrast, blind inverse problems arise when $\mathcal{A}_{\psi}$ is unknown, requiring the simultaneous estimation of both $\mathcal{A}_{\psi}$ and $\boldsymbol{x}$, which presents a significantly greater challenge.

Inverse problems are inherently ill-posed, often rely heavily on data priors $p(\boldsymbol{x})$ for accurate computation. Recently, diffusion models (DMs) have emerged as powerful tools for solving inverse problems due to their remarkable ability to capture complex data distributions $p(\boldsymbol{x})$ [9, 10, 13, 34]. A straightforward approach to leveraging DMs for solving inverse problem involves training a conditional DM to directly estimate the posterior $p(\boldsymbol{x}|\boldsymbol{y})$ via supervised learning. However, this method can be computationally intensive, as it requires training separate DMs for each distinct measurement operator $\mathcal{A}_{\psi}$.

To overcome this limitation, recent work has focused on approximating the posterior by leveraging pre-trained, unconditional DMs that estimate the prior $p(\boldsymbol{x})$, thus bypassing the need for additional model training. In this approach, the prior $p(\boldsymbol{x})$ provided by the DMs is combined with the likelihood $p(\boldsymbol{y}|\boldsymbol{x})$ to sample from the posterior distribution in inverse problems. These methods rely on approximating the likelihood term $p(\boldsymbol{y}|\boldsymbol{x})$, as it is analytically intractable [9, 34].

Nevertheless, most inverse-problem solvers proposed in the literature are strictly limited to scenarios in which the measurement operator $\mathcal{A}_{\psi}$ is known and fixed [9, 34]. To address this issue, we propose **CL-DPS**, a method based on **c**ontrastive **l**earning for solving blind inverse problems via **d**iffusion **p**osterior **s**ampling. Specifically, in CL-DPS, first an auxiliary deep neural network (DNN) is trained offline using a modified version of MoCo [16], a contrastive learning (CL) technique. The role of this auxiliary DNN is to estimate the likelihood $p(\boldsymbol{y}|\boldsymbol{x})$ without knowing the measurement $\mathcal{A}_{\psi}$. Then, during inverse problem solving, we perform inference with this auxiliary DNN to estimate $p(\boldsymbol{y}|\boldsymbol{x})$, which is then used to adjust the reverse path of the diffusion process. To further improve the auxiliary DNN's accuracy in estimating $p(\boldsymbol{y}|\boldsymbol{x})$, we introduce a novel overlapped patch-wise inference method that divides the images into patches during the inference stage.

To evaluate the effectiveness of CL-DPS, we conduct experiments on two well-known datasets named FFHQ [21] and AFHQ, [5] under both blind linear and non-linear measurements. Notably, in the non-linear measurement setting, such as rotation blur, all benchmark methods fail, whereas CL-DPS successfully restores the images (see Fig. 1 for restored images from rotation blur). In summary, the contributions of the paper are as follows:

- We propose CL-DPS, an inverse problem solver using diffusion models for the blind setting. CL-DPS incorporates an auxiliary DNN, trained using MoCo, to serve as a likelihood estimator. Unlike previous blind solvers, which are limited to recovering images only under linear measurements, CL-DPS is capable of recovering images for both linear and non-linear measurements.
- To increase the accuracy of the auxiliary DNN in estimating the likelihood, we introduce overlapped patch-wise inference, an information theoretically certified method that increases mutual information between the DNN's input and output, allowing it to capture richer semantic information.
- Through extensive quantitative and qualitative experiments, we demonstrate that CL-DPS effectively addresses both linear and non-linear blind inverse problems.

2. Related Works and Notation

2.1. Diffusion Model for Inverse Problem

The use of diffusion models to solve inverse problems through posterior sampling has recently attracted considerable attention across various domains. For blind inverse problems, alongside the approaches discussed in Sec. 1 [8, 27, 32], [1] introduced Blind RED-Dif, an extension of the RED-diff framework [25]. This method employs variational inference to jointly estimate both the latent image and the unknown forward model parameters, addressing the challenges of unknown measurement operators.

2.2. Contrastive Learning

As a versatile semi-supervised learning framework, contrastive learning learns useful feature representation by clustering positive samples and dispersing negative samples. It achieves great success since instance discrimination has been proposed in [41]. For interested readers seeking further information, please refer to the survey paper [14].

2.3. Notation

For a positive integer C, let $[C] \triangleq \{1, \dots, C\}$, and $[C_1, \dots, C_2] \triangleq \{C_1, \dots, C_2\}$. Denote by $P[i]$ the i-th element of vector P. Scalars are denoted by lowercase letters (e.g. u), vectors by boldface lowercase letters (e.g. $\boldsymbol{u}$). For two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, denote by $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ their inner product. We use $|\mathcal{C}|$ to denote the cardinality of a set $\mathcal{C}$. $(\cdot)^{\mathsf{T}}$ denotes the transpose operation. We denote a closed interval by $[A, B]$, an open interval by (A, B), and a half-

open interval by $(A, B]$ or $[A, B)$. The mutual information between two random variables X and Y is given by $\mathrm{I}(X, Y) = H(X) - H(X|Y)$, where $H(\cdot)$ denotes the entropy function. Let $f_{\boldsymbol{\theta}}$ denote a DNN parameterized by $\boldsymbol{\theta}$, with $f(\cdot)$ representing the output of the DNN.

3. Background and Preliminaries

3.1. Diffusion Models

Diffusion models define a generative process as the reverse of a noise addition process. Specifically, [35] introduced the Itô-stochastic differential equation (SDE) to describe this noise addition process—referred to as the forward SDE—for the data $\boldsymbol{x}_t$ over a continuous time interval $t \in [0, T]$, where $\boldsymbol{x}_t \in \mathbb{R}^d$ for all t.

In this paper, we adopt the variance-preserving form of the SDE (VP-SDE) [35], which is equivalent to the DDPM framework [18] whose equation is given as follows:

$$d\boldsymbol{x} = -\frac{\beta_t}{2}\boldsymbol{x}\, dt + \sqrt{\beta_t}\, d\boldsymbol{w}, \tag{1}$$

where $\beta_t : \mathbb{R} \to \mathbb{R}^+$ represents the noise schedule of the process, which is typically chosen as a monotonically increasing linear function of t [18]. The term $\boldsymbol{w}$ represents the standard d-dimensional Wiener process. The data distribution is specified at $t = 0$, i.e., $\boldsymbol{x}_0 \sim p_{\text{data}}$, while at $t = T$, the process reaches a simple, tractable distribution, such as an isotropic Gaussian: $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

The goal is to recover the data-generating distribution from the tractable distribution. This can be accomplished by formulating the corresponding reverse SDE for Eq. (1), as derived in [3]:

$$d\boldsymbol{x} = \left[-\frac{\beta_t}{2}\boldsymbol{x} - \beta_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right] dt + \sqrt{\beta_t} d\bar{\boldsymbol{w}}, \tag{2}$$

where dt represents time flowing backward, and $d\bar{\boldsymbol{w}}$ corresponds to the standard Wiener process in reverse. The drift function now depends on the time-dependent score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$, which is approximated by a neural network $\boldsymbol{s}_\theta$ trained via denoising score matching [39]:

$$\theta^* = \arg\min_\theta \mathbb{E}_{t\sim U(\varepsilon,1), \boldsymbol{x}_t \sim p(\boldsymbol{x}_t|\boldsymbol{x}_0), \boldsymbol{x}_0 \sim p_{\text{data}}} \tag{3}$$

$$\left[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2\right], \tag{4}$$

where $\varepsilon \simeq 0$ represents a small positive constant. Once the optimal parameters θ^* are obtained through Eq. (3), the approximation $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) \simeq \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t)$ can be used as a plug-in estimate to replace the score function in Eq. (2).

Discretizing Eq. (2) and solving it produces samples from the data distribution $p(\boldsymbol{x}_0)$, which is the ultimate goal of generative modeling. In addition, following [18], we introduce $\alpha_i \triangleq 1 - \beta_i$ and $\bar{\alpha}_i \triangleq \prod_{j=1}^{i} \alpha_j$.

3.2. Diffusion Models for Solving Inverse Problems

We consider the problem of reconstructing an unknown signal $\boldsymbol{x}_0 \in \mathbb{R}^d$ from noisy measurements $\boldsymbol{y} \in \mathbb{R}^m$:

$$\boldsymbol{y} = \mathcal{A}_{\boldsymbol{\psi}}(\boldsymbol{x}_0) + \boldsymbol{n}, \tag{5}$$

where $\mathcal{A}_{\boldsymbol{\psi}}(\cdot) : \mathbb{R}^d \to \mathbb{R}^m$ represents a measurement operator (can be linear or nonlinear) with parameters $\boldsymbol{\psi}$, which we assume to be unknown in our setting—an approach we refer to as "blind". Additionally, $\boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is i.i.d. additive Gaussian noise with a known standard deviation σ. This leads to a likelihood function $p(\boldsymbol{y}|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{y}|\mathcal{A}_{\boldsymbol{\psi}}(\boldsymbol{x}_0), \sigma^2\mathbf{I})$.

Typically, we are interested in the case where $m < d$, which aligns with many real-world scenarios. When $m < d$, the problem becomes ill-posed, requiring some form of *prior* to obtain a meaningful solution. In the Bayesian framework, a prior distribution $p(\boldsymbol{x}_0)$ is employed, with samples drawn from the *posterior* $p(\boldsymbol{x}_0|\boldsymbol{y})$. The relationship is formally defined by Bayes' rule: $p(\boldsymbol{x}_0|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x}_0)p(\boldsymbol{x}_0)}{p(\boldsymbol{y})}$. By using a diffusion model as the prior, we can directly modify Eq. (2) to obtain the reverse diffusion sampler for sampling from the posterior distribution:

$$d\boldsymbol{x} = \Big[-\frac{\beta_t}{2}\boldsymbol{x} - \beta_t(\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t))\Big] dt + \sqrt{\beta_t} d\bar{\boldsymbol{w}}, \tag{6}$$

where we have used the fact that

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t). \tag{7}$$

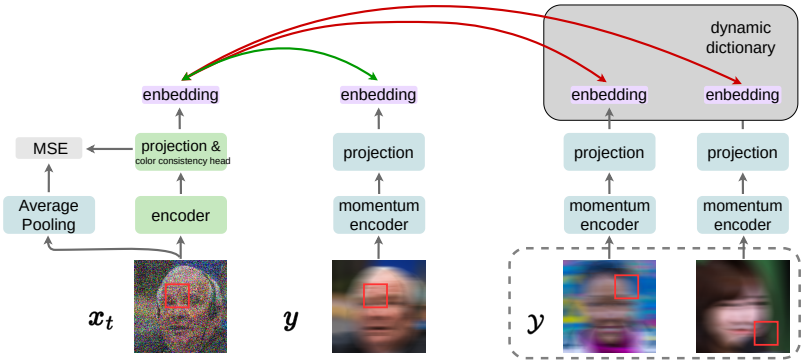
In Eq. (6), two terms need to be computed: the score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ and the likelihood $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$. To compute the former, $p_t(\boldsymbol{x}_t)$, we can directly use the pretrained score function $\boldsymbol{s}_{\theta^*}$. However, obtaining the latter term in closed form is challenging due to its time dependence, as only an explicit relationship between $\boldsymbol{y}$ and $\boldsymbol{x}_0$ exists. Thus, the likelihood $p_t(\boldsymbol{y}|\boldsymbol{x}_t)$ must be estimated.

To estimate $p_t(\boldsymbol{y}|\boldsymbol{x}_t)$, some prior studies assume that the measurement $\boldsymbol{A}(\boldsymbol{x}_0)$ is known [6, 9]. However, this assumption often diverges significantly from real-world scenarios. Alternatively, other research focuses on cases where $\boldsymbol{A}(\boldsymbol{x}_0)$ is unknown, addressing what is commonly referred to as the "blind inverse problem" [2, 12]. Our work follows this latter approach, with a particular emphasis on leveraging contrastive learning, as detailed in Sec. 4.

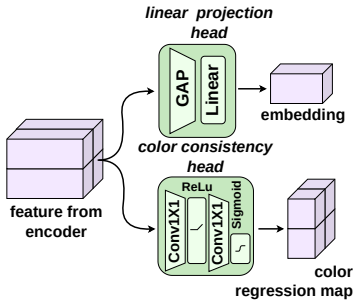
We differ the momentum contrastive learning to the appendix.

4. Methodology

As discussed in Sec. 3, estimating the posterior $p_t(\boldsymbol{x}_t|\boldsymbol{y})$ requires an estimation of the likelihood $p_t(\boldsymbol{y}|\boldsymbol{x}_t)$. To achieve



(a) Training the auxiliary DNN utilizing the MoCo framework.



(b) A detailed view of the linear projection head and color consistency head.

Figure 2. (a) Illustration of the training process for the auxiliary DNN. Small patches from $\boldsymbol{x}_t$ and $\boldsymbol{y}$ are used as keys and queries, respectively. Green and red arrows indicate positive and negative pairs, respectively. (b) Structure of the linear projection head and the color consistency head.

this, we aim to train an auxiliary DNN offline (prior to applying diffusion models for inverse problem-solving) which is able to estimate the likelihood $p_t(\boldsymbol{y}|\boldsymbol{x}_t)$. Note that at this the time the measurement parameters $\boldsymbol{\psi}$ are unknown. This auxiliary DNN will then be employed during the diffusion-based inverse problem-solving process to adjust the reverse diffusion path accordingly.

In the following sections, we fully discuss how to train each of the components.

4.1. Training the Auxiliary DNN

To train the auxiliary DNN for likelihood estimation, we begin by establishing a connection between CL and likelihood estimation in the following subsection.

4.1.1 CL as Likelihood Estimation

First, note that using Bayes' formula, the likelihood $p_t(\boldsymbol{y}|\boldsymbol{x}_t)$ can be expressed as

$$p(\boldsymbol{y}|\boldsymbol{x}_t) = \frac{p(\boldsymbol{y}, \boldsymbol{x}_t)}{p(\boldsymbol{x}_t)} = \frac{p(\boldsymbol{y}, \boldsymbol{x}_t)}{\int p(\tilde{\boldsymbol{y}}, \boldsymbol{x}_t) d\tilde{\boldsymbol{y}}}. \tag{8}$$

To compute Eq. (8), we first obtain a numerical representation of its numerator, $p(\boldsymbol{y}, \boldsymbol{x}_t)$. Specifically, following [24, 29], we approximate $p(\boldsymbol{y}, \boldsymbol{x}_t) \propto \exp(\langle f(\boldsymbol{x}_t), f(\boldsymbol{y})\rangle/\tau)$, where the neural network f produces a feature representation in a transformed space.

The denominator in Eq. (8), $\int p(\tilde{\boldsymbol{y}}, \boldsymbol{x}_t) d\tilde{\boldsymbol{y}}$, is generally intractable. Thus, we rely on an approximation method, using a summation as follows: $\int p(\tilde{\boldsymbol{y}}, \boldsymbol{x}_t) d\tilde{\boldsymbol{y}} \approx \sum_{\tilde{\boldsymbol{y}} \in \mathcal{Y}} p(\tilde{\boldsymbol{y}}, \boldsymbol{x}_t)$, where $\mathcal{Y}$ is a sufficiently large set. This allows us to numerically approximate $p(\boldsymbol{y}|\boldsymbol{x}_t)$ as follows:

$$p(\boldsymbol{y}|\boldsymbol{x}_t) \approx \frac{\exp(\langle f(\boldsymbol{x}_t), f(\boldsymbol{y})\rangle/\tau)}{\sum_{\tilde{\boldsymbol{y}} \in \mathcal{Y}} \exp(\langle f(\boldsymbol{x}_t), f(\tilde{\boldsymbol{y}})\rangle/\tau)}. \tag{9}$$

Now the question is how the the DNN f should be trained such that Eq. (9) is a good approximation for the likelihood $p(\boldsymbol{y}|\boldsymbol{x}_t)$? A natural method to this aim is to train a DNN to directly maximize the log-likelihood $\log\left(p(\boldsymbol{y}|\boldsymbol{x}_t)\right)$ or equivalently minimizes the negative log-likelihood loss:

$$\mathcal{L}_{p(\boldsymbol{y}|\boldsymbol{x}_t)} = -\log \frac{\exp(\langle f(\boldsymbol{x}_t), f(\boldsymbol{y})\rangle/\tau)}{\sum_{\tilde{\boldsymbol{y}} \in \mathcal{Y}} \exp(\langle f(\boldsymbol{x}_t), f(\tilde{\boldsymbol{y}})\rangle/\tau)}. \tag{10}$$

Comparing the loss function in Eq. (10) with the InfoNCE loss in Eq. (21), we observe a resemblance: by setting $q = f(\boldsymbol{x}_t)$ and $\{k_i\}_{i\in[K]} = \mathcal{Y}$, we can use the CL loss to train a DNN to estimate the likelihood $p(\boldsymbol{y}|\boldsymbol{x}_t)$. Note that with a sufficiently large number of keys K (as is typical in MoCo, where $K = 4096$), the set $\{k_i\}_{i\in[K]}$ serves as an effective approximation for $\mathcal{Y}$.

4.1.2 Incorporating Color Consistency Loss

As discussed in Sec. 4.1.1, the loss function in Eq. (10) can be utilized to train a DNN for likelihood estimation. However, when using the objective function in Eq. (10) to train the auxiliary DNN, we observe that the color information in images is often lost, resulting in images with colors that differ from the original. To address this, we introduce a two-layer convolutional neural network head, referred to as the *color consistency head*. This head ensures that features extracted from the encoder retain sufficient color information from the input, which is essential for the inverse problem. It does so by regressing the average color of the input image using the mean squared error (MSE) loss, penalizing the DNN as follows: $\mathsf{MSE}(\mathsf{H}_c(\boldsymbol{x}_t); \mathsf{AP}(\boldsymbol{x}_t))$, where $\mathsf{H}_c(\boldsymbol{x}_t)$ represents the output of the color consistency head, and $\mathsf{AP}(\cdot)$ denotes the average pooling operation (Fig. 2b depicts an overview of the color consistency head). Conse-

quently, we use the loss function for CL-DPS becomes

$$\mathcal{L}_{\text{CL-DPS}} = -\log \frac{\exp(\langle f(\boldsymbol{x}_t), f(\boldsymbol{y})\rangle/\tau)}{\sum_{\tilde{\boldsymbol{y}}\in\mathcal{Y}} \exp(\langle f(\boldsymbol{x}_t), f(\tilde{\boldsymbol{y}})\rangle/\tau)} + \lambda\, \mathsf{MSE}(\mathsf{H}_c(\boldsymbol{x}_t); \mathsf{AP}(\boldsymbol{x}_t)), \tag{11}$$

where the hyper-parameter λ balances the importance of likelihood estimation and color consistency.

4.1.3 Training the Auxiliary DNN for DPS

To use Eq. (11) for training the auxiliary DNN, we assign random parameters $\boldsymbol{\psi}$ to the measurement operator $\mathcal{A}_{\boldsymbol{\psi}}(\cdot)$ to generate $\boldsymbol{y}$.

To obtain $\{\boldsymbol{x}_t\}_{t\in[T]}$, we use the forward diffusion representation for models like VP-SDE or DDPM (the focus of this paper), where $\boldsymbol{x}_t = \sqrt{\bar{\alpha}}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}}\boldsymbol{n}$, with $\bar{\alpha}_i \triangleq \prod_{j=1}^{i} \alpha_j$ (where $0 < \alpha_j < 1$ denotes the noise schedule) and $\boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ as standard Gaussian noise. During auxiliary DNN training, we randomly sample an $\boldsymbol{x}_t$ from $\{\boldsymbol{x}_t\}_{t\in[T]}$. Additionally, only a small patch is cropped from the original image, encouraging the model to focus on learning low-level features (see Fig. 2a for more details).

4.2. Likelihood Estimation Using the Auxiliary DNN

Once the auxiliary DNN is trained using the loss function Eq. (11), it is used to estimate the likelihood $p(\boldsymbol{y}|\boldsymbol{x}_t)$ during inverse problem solving. However, we know that in general, the convolutional neural networks (CNNs) can impair low-level vision details. Specifically, CNNs are well-known for effectively compressing information from the input layer to the output layer [4, 37, 42, 43]. In the next section we purpose a new method to further improve the likelihood estimation.

4.2.1 Overlapped Patch-Wise Inference

Here, we propose a post-training inference method, which we refer to as overlapped patch-wise inference, which encourages the DNN to retain more information about the image $\boldsymbol{x}$ in its output.

Specifically, given an image $\boldsymbol{x} \in R^{(N_1,N_2)}$, we first patchify it into L_s overlapped $n\times n$ patches $\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[L_s]}$ with stride $s < n$, which yields $L_s = \lfloor \frac{N_1-n}{s} + 1 \rfloor \lfloor \frac{N_2-n}{s} + 1 \rfloor$.

Next, instead of performing inference over the image $\boldsymbol{x}$, we parallely perform inference over the L_s patches, and then concatenate the output of the DNN for these patches as $f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[L_s]}) = [f^{\mathsf{T}}(\boldsymbol{p}_1^{\boldsymbol{x}}), \ldots, f^{\mathsf{T}}(\boldsymbol{p}_{L_s}^{\boldsymbol{x}})]^{\mathsf{T}}$. In order to analytically show that such patchifying increases the information of the DNN's output about the image $\boldsymbol{x}$, we first need to quantify the information of DNN's output $f(\boldsymbol{x})$ about its input $\boldsymbol{x}$. To this end, we deploy mutual information quantity $\mathrm{I}(\boldsymbol{x}; f(\boldsymbol{x}))$ [11, 33], in that the higher this value is, the more information $f(\boldsymbol{x})$ has about $\boldsymbol{x}$. Using this metric, the following theorem discusses that $f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[L_s]})$ contains more information about $\boldsymbol{x}$ compared to the output $f(\boldsymbol{x})$.

Theorem 1 *For a given DNN f, and independent and identically distributed (i.i.d.) sampled input $\boldsymbol{x} \in \mathbb{R}^{(N,N)}$, we patchify it into some overlapped patches. For $U, V \in \mathbb{N}$, assume that we patchify $\boldsymbol{x}$ one time to U patches $f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[U]})$, and one time to V patches $f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]})$. Then, if $U < V$, we have*

$$\mathrm{I}(\boldsymbol{x}, f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[U]})) \leq \mathrm{I}(\boldsymbol{x}, f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]})). \tag{12}$$

The proof for Theorem 1 is differed to the supplementary material. Theorem 1 states that if $\boldsymbol{x}$ is patchified to more overlapping patches, then the output of the DNN would have more information about the input $\boldsymbol{x}$. We examine the effectiveness of overlapped patch-wise inference in Sec. 8.1.

Algorithm 1 CL-DPS

1: **Input:** The number of iterations N, $\boldsymbol{y}$, noise levels $\{\tilde{\sigma}\}$, pre-trained encoder $f(\cdot)$, and $\eta > 0$.
2: $\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
3: **for** $t = N-1, N-2, \ldots, 0$ **do**
4: $\quad \hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$
5: $\quad \tilde{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\hat{\boldsymbol{s}})$
6: $\quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
7: $\quad \boldsymbol{x}'_{t-1} \leftarrow \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\tilde{\boldsymbol{x}}_0 + \tilde{\sigma}_t\boldsymbol{z}$.
8: $\quad$ Patchify $\boldsymbol{x}_t$ and $\boldsymbol{y}$ to $U \geq 2$ patches: $\{\boldsymbol{p}_j^{\boldsymbol{x}_t}\}_{j\in[U]} \leftarrow \boldsymbol{x}_t$; $\{\boldsymbol{p}_j^{\boldsymbol{y}}\}_{j\in[U]} \leftarrow y$.
9: $\quad \boldsymbol{x}_{t-1} \leftarrow \boldsymbol{x}'_{t-1} - \eta\nabla_{\boldsymbol{x}_t} \langle f(\{\boldsymbol{p}_j^{\boldsymbol{x}_t}\}_{j\in[U]}), f(\{\boldsymbol{p}_j^{\boldsymbol{y}}\}_{j\in[U]})\rangle$.
10: **end for**
11: **Output:** $\boldsymbol{x}_0$

Henceforth, during the likelihood estimation, we perform the same patchification for the measurement signal $\boldsymbol{y}$, and compute the likelihood probability as

$$p(\boldsymbol{y}|\boldsymbol{x}_t) \propto \langle f(\{\boldsymbol{p}_j^{\boldsymbol{x}_t}\}_{j\in[U]}), f(\{\boldsymbol{p}_j^{\boldsymbol{y}}\}_{j\in[U]})\rangle, \tag{13}$$

where the proportionality $\propto$ accounts for the denominator in Eq. (9) being nearly constant for large $\mathcal{Y}$.

Finally, it is worth noting that only the encoder is retained after the training phase.

4.3. Algorithm for CL-DPS

After training the auxiliary DNN, only the encoder will be preserved and be used to estimate the posterior distribution. Then, we incorporate the pre-trained encoder as a likelihood estimator in to the DPS method [10] which leads to Algorithm 1. Note that the only distinction between the CL-DPS algorithm and unconditional sampling lies in lines 8 and 9 (highlighted in blue), where conditioning is introduced.

Method	FFHQ (256 × 256)						AFHQ (256 × 256)					
	Rotation			Zoom			Rotation			Zoom		
	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓
CL-DPS (Ours)	**22.74**	**33.66**	**0.302**	**20.68**	**42.61**	**0.435**	**21.46**	**36.96**	**0.319**	**19.63**	**57.54**	**0.468**
BlindDPS [8]	16.87	343.76	0.552	16.39	292.91	0.780	13.25	200.46	0.674	11.75	279.57	0.607
FastEM [23]	15.96	268.43	0.597	18.68	303.25	0.623	11.57	289.19	0.680	15.60	310.06	0.797
GibbsDDRM [27]	18.43	236.55	0.565	15.45	327.42	0.802	15.24	263.49	0.628	14.57	280.54	0.549

Table 1. **Non-linear** blind inverse problems: Blind rotation and zoom deblurring results on the FFHQ and AFHQ datasets. CL-DPS successfully restores the input images with high quality, whereas all other methods fail. **Bold** and underlined values denote the best and second-best results, respectively.

Method	FFHQ (256 × 256)						AFHQ (256 × 256)					
	Motion			Gaussian			Motion			Gaussian		
	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓
CL-DPS (Ours)	22.93	32.44	0.157	**24.82**	**26.64**	0.348	**22.06**	42.25	0.280	**23.76**	20.56	**0.225**
SelfDeblur [31]	10.83	270.0	0.717	11.36	235.4	0.686	9.081	300.5	0.768	11.53	172.2	0.662
DeblurGANv2 [22]	17.75	220.7	0.571	19.69	185.5	0.529	17.64	186.2	0.597	20.29	86.87	0.523
Pan_l0 [30]	15.53	242.6	0.542	19.94	92.70	0.415	15.34	235.0	0.627	21.41	62.76	0.395
BlindDPS [8]	22.24	**29.49**	0.281	24.77	27.36	**0.233**	20.92	**23.89**	0.338	23.63	**20.54**	0.287
FastEM [23]	24.68	-	0.34	-	-	-	-	-	-	-	-	-
LatentDEM [40]	22.65	-	0.167	-	-	-	-	-	-	-	-	-
GibbsDDRM [27]	**25.80**	38.71	**0.115**	-	-	-	22.01	48.00	**0.197**	-	-	-

Table 2. **Linear** blind inverse problems: Blind motion and Gaussian deblurring results on the FFHQ and AFHQ datasets. CL-DPS achieves competitive results compared to other benchmark methods.

5. Experiments

In this section we evaluate CL-DPS under blind inverse settings for both linear and non-linear measurements.

Datasets. For our experiments, we use Flickr-faces-HQ (FFHQ) 256 × 256 dataset [21] and animal faces-HQ (AFHQ) 256 × 256 dataset [5]. Similar to the previous works [8, 23, 27], for FFHQ, we randomly select 50k images for training, and sample 1k images of test data separately. For AFHQ, we train our model using the images in the dog category, which consists of about 5k images. Testing was performed with the held-out validation set of 500 images of the same category.

• **Pre-trained diffusion models.** We leverage pre-trained score functions as those used in [7].

• **Evaluation metrics.** For all experiments, we use Fréchet inception distance (FID) [17], learned perceptual image patch similarity (LPIPS) [44] and peak signal-to-noise ratio (PSNR) between the original image and reconstructed image as the evaluation metrics.

• **Benchmarks.** We compare the performance of CL-DPS with the following seven benchmark methods which are designed for solving blind inverse problems: SelfDeblur [31], DeblurGANv2 [22], Pan_l0 [30], BlindDPS [8], FastEM [23], LatentDEM [40], GibbsDDRM [27]. Notably, the last four methods use diffusion models in their methodology.

Non-linear deblurring. Non-linear deblurring is commonly encountered in real life, often resulting from phenomena such as rotation, rolling shutter effects, and zoom blur during image capturing. Here, we consider rotation blur and zoom deblurring tasks as non-linear inverse problems. In particular, to generate rotation-blurred measurements, we randomly select the center point among the input images and set the rotation angle within the range of $[10° - 30°]$ and applying a random weight to the rotation trajectory. For zoom blur, we set the center of the image as the focal point of the zoom, then apply a zoom factor ranging $[1 - 3]$.

5.1. Results

The qualitative results for the rotation deblurring task using benchmark methods and CL-DPS are shown in Fig. 1. As observed, CL-DPS is the only method capable of accurately recovering the ground truth images, while all benchmark methods fail to do so. Qualitative results for the zoom deblurring task are provided in the Appendix. Additionally, the quantitative results are presented in Tab. 1. The results on both datasets show the significant superiority of CL-DPS over benchmark methods in restoring original images.

Linear deblurring. For linear deblurring, we consider Gaussian and motion deblurring. Specifically, following [23, 27, 40], we apply the Gaussian blur kernel with the size of 61 × 61 and standard deviation of 3.0. Also, the motion blur kernel is generated randomly using an open-source code[1], with kernel size of 61 × 61 and intensity of 0.5. These kernels are convolved with the ground truth image to produce the measurement.

Tab. 2 summarizes the quantitative results for Gaussian and motion deblurring tasks. Compared to state-of-the-art methods, CL-DPS achieves competitive performance across various metrics under blind linear inverse settings. Notably, CL-DPS outperforms all the other methods in terms

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

of PSNR and FID score on the FFHQ dataset when subjected to Gaussian blur.

6. Conclusion and Future Work

In this work, we demonstrated the potential of DMs for solving blind inverse problems with unknown measurements, extending their applicability beyond the limitations of previous methods, which focused primarily on linear measurements. We introduced CL-DPS, where an auxiliary DNN trained offline with a modified MoCo approach as a likelihood estimator. This allowed us to estimate $p(\boldsymbol{y}|\boldsymbol{x})$ without prior knowledge of the measurement operator, thereby adjusting the reverse diffusion process.

References

[1] Cagan Alkan, Julio Oscanoa, Daniel Abraham, Mengze Gao, Aizada Nurdinova, Kawin Setsompop, John M Pauly, Morteza Mardani, and Shreyas Vasanawala. Variational diffusion models for blind mri inverse problems. In *NeurIPS 2023 Workshop on Deep Learning and Inverse Problems*, 2023. 2

[2] Mariana S. C. Almeida and Mario A. T. Figueiredo. Blind image deblurring with unknown boundaries using the alternating direction method of multipliers. In *2013 IEEE International Conference on Image Processing*, pages 586–590, 2013. 3

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

[4] Kwan Ho Ryan Chan, Yaodong Yu, Chong You, Haozhi Qi, John Wright, and Yi Ma. Redunet: A white-box deep network from the principle of maximizing rate reduction. *Journal of machine learning research*, 23(114):1–103, 2022. 5

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 2, 6

[6] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022. 3

[7] 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*, pages 12413–12422, 2022. 6

[8] Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. Parallel diffusion models of operator and image for blind inverse problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6059–6069, 2023. 1, 2, 6, 3

[9] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 3

[10] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 5

[11] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999. 5

[12] N. Damera-Venkata, T.D. Kite, M. Venkataraman, and B.L. Evans. Fast blind inverse halftoning. In *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, pages 64–68 vol.2, 1998. 3

[13] Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[14] Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2024. 2

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 1735–1742. IEEE, 2006. 2

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1, 2

[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. *Advances in neural information processing systems*, 30, 2017. 6

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[19] Mahdi S. Hosseini and Konstantinos N. Plataniotis. Convolutional deblurring for natural imaging. *IEEE Transactions on Image Processing*, 29:250–264, 2020. 2

[20] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE transactions on image processing*, 26(9):4509–4522, 2017. 2

[21] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 43(12):4217–4228, 2021. 2, 6

[22] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8878–8887, 2019. 6

[23] Charles Laroche, Andres Almansa, and Eva Coupete. Fast Diffusion EM: a diffusion model for blind inverse problems

with application to deconvolution . In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5259–5269, Los Alamitos, CA, USA, 2024. IEEE Computer Society. 1, 6, 3

[24] Junjie Li, Yixin Zhang, Zilei Wang, Keyu Tu, and Saihui Hou. Probabilistic contrastive learning for domain adaptation. *arXiv preprint arXiv:2111.06021*, 2021. 4

[25] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *The Twelfth International Conference on Learning Representations*. 2

[26] Michael T McCann, Kyong Hwan Jin, and Michael Unser. Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, 34(6): 85–95, 2017. 2

[27] Naoki Murata, Koichi Saito, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Gibbsddrm: A partially collapsed gibbs sampler for solving blind inverse problems with denoising diffusion restoration. In *International conference on machine learning*, pages 25501–25522. PMLR, 2023. 1, 2, 6, 3

[28] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1 (1):39–56, 2020. 2

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4, 2

[30] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. l_0 -regularized intensity and gradient prior for deblurring text images and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):342–355, 2017. 6

[31] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3341–3350, 2020. 6

[32] Yash Sanghvi. *Kernel Estimation Approaches to Blind Deconvolution*. PhD thesis, Purdue University Graduate School, 2024. 2

[33] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. 5

[34] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. 2

[35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 3

[36] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020. 2

[37] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015. 5

[38] Francesco Tonolini, Jack Radford, Alex Turpin, Daniele Faccio, and Roderick Murray-Smith. Variational inference for computational imaging inverse problems. *Journal of Machine Learning Research*, 21(179):1–46, 2020. 2

[39] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 3

[40] Bai Weimin, Chen Siyi, Chen Wenzheng, and Sun He. Blind inversion using latent diffusion priors. *arXiv preprint arXiv:2406.03184*, 2024. 6

[41] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 2

[42] En-Hui Yang, Shayan Mohajer Hamidi, Linfeng Ye, Renhao Tan, and Beverly Yang. Conditional mutual information constrained deep learning for classification. *arXiv preprint arXiv:2309.09123*, 2023. 5

[43] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in neural information processing systems*, 33: 9422–9434, 2020. 5

[44] 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. 6

[45] Michael S Zhdanov. *Geophysical inverse theory and regularization problems*. Elsevier, 2002. 2

CL-DPS: A Contrastive Learning Approach to Blind Inverse Problem Solving via Diffusion Posterior Sampling

Supplementary Material

7. Proof of Theorem 1

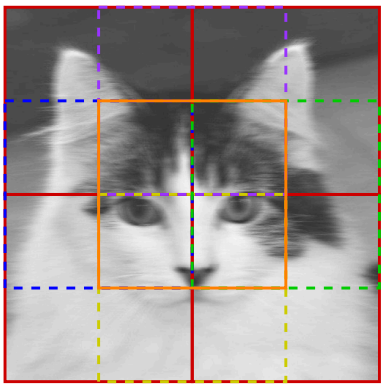


Figure 3. Example of patchifed image of a resolution 256×256, with a stride size of 64 and a patch size of 128×128.

We start the proof by writing the mutual information in terms of entropy:

$$\mathrm{I}(\boldsymbol{x}, f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]})) = \mathrm{H}(\boldsymbol{x}) - \mathrm{H}(\boldsymbol{x}|f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]})) \quad (14)$$

Now, denote by $\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]} \setminus \{\boldsymbol{p}_i^{\boldsymbol{x}}\}_{i\in[U]}$ the set of all elements in $\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]}$ which do not present in $\{\boldsymbol{p}_i^{\boldsymbol{x}}\}_{i\in[U]}$. Note that since $U < V$, $\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]} \setminus \{\boldsymbol{p}_i^{\boldsymbol{x}}\}_{i\in[U]}$ is a non-empty set. Now, we have

$$\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]} = \left\{\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]} \setminus \{\boldsymbol{p}_i^{\boldsymbol{x}}\}_{i\in[U]}\right\} \cup \{\boldsymbol{p}_i^{\boldsymbol{x}}\}_{i\in[U]} \quad (15)$$

Next, using Eq. (15) in Eq. (14) we obtain

$$\begin{aligned}
&\mathrm{I}(\boldsymbol{x}, f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]})) = \mathrm{H}(\boldsymbol{x}) && (16)\\
&- \mathrm{H}\left(\boldsymbol{x}|f\left(\left\{\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[V]} \setminus \{\boldsymbol{p}_i^{\boldsymbol{x}}\}_{i\in[U]}\right\} \cup \{\boldsymbol{p}_i^{\boldsymbol{x}}\}_{i\in[U]}\right)\right) && (17)\\
&\geq \mathrm{H}(\boldsymbol{x}) - \mathrm{H}(\boldsymbol{x}|f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[U]})) && (18)\\
&= \mathrm{I}(\boldsymbol{x}, f(\{\boldsymbol{p}_j^{\boldsymbol{x}}\}_{j\in[U]})), && (19)
\end{aligned}$$

where Eq. (18) holds since conditioning reduces entropy. Hence, the proof is concluded.

8. More Ablation Study

8.1. Overlapped Patch-Wised Inference and Global Average Pooling

In this section, we analyze the impact of the global average pooling (GAP) layer and overlapped patch-wise inference on image restoration. Figure 4 illustrates the four possible configurations combining the presence or absence of GAP and patch-wise inference (refer to the figure caption for details on each setting). Among these, the configuration in Figure 4(c)—where the GAP layer is excluded and overlapped patch-wise inference is applied—produces the most visually accurate restored image.

Figure 4. Ablation study on different inference methods. (a) original images, (b) measurement, and restored image (c) without GAP and with patch-wise inference, (d) with GAP and without patch-wise inference, (e) with GAP and with patch-wise inference, (f) without GAP and without patch-wise inference.

8.2. Color Consistency Head

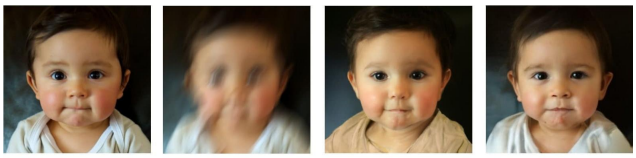


Figure 5. Ablation study on the effect of color consistency head. (a) original image, (b) measurement, (c) restored image from the model trained without color consistency head, (d) restored image from the model trained with color consistency head.

In this section, we qualitatively assess the effect of the color consistency head. Fig. 5 demonstrates its effect: the restored image (Fig. 5 (c)) from the model trained without the color consistency head fails to capture the original color, resulting in a significant color mismatch, especially in the

shirt. This issue is resolved in Fig. 5(d), where the color consistency head is incorporated in the training process.

9. Qualitative results on Zoom deblurring task

Zoom blur, a highly challenging non-linear blur in the context of diffusion-based deblurring, presents substantial difficulties for existing techniques. The results, presented in Figure 7, highlight this complexity. Among the benchmark methods, CL-DPS (ours) emerges as the **sole** approach capable of reliably recovering the original signal without catastrophic failure. This outcome demonstrates the robustness and adaptability of CL-DPS in addressing intricate non-linear blurs where other benchmark methods fall short.

10. Denoising Process of CL-DPS

Here, we visualize the denoising process of CL-DPS over 1000 timesteps. To this end, we select a single image and display the reconstructed images throughout the denoising process, as illustrated in Fig. 7.

10.1. Momentum Contrast Learning

Contrastive learning (CL) is a method that teaches machines to understand which data points are similar or different by contrasting them with each other, helping them learn useful representations without explicit labels [15, 36]. Kaiming *et al*. proposed momentum contrast (MoCo) as an efficient CL method to learn a feature encoder $f_{\boldsymbol{\theta}}$ from an unlabeled dataset [16]. The core idea is to treat CL as a dictionary look-up. To elucidate, imagine a dictionary where "keys" are encoded representations of images. Given a "query" (another image), the goal is to find the most similar key. MoCo trains a model to do this, forcing it to learn meaningful image representations.

MoCo uses two main components: (i) Queue: a large queue stores encoded "keys" (image representations). New keys are enqueued, old ones dequeued, keeping the dictionary diverse and up-to-date. (ii) momentum encoder: instead of directly using the query encoder (the model encoding the "query" image) to encode keys, MoCo uses a separate encoder, updated with a momentum term:

$$\boldsymbol{\theta}_k \leftarrow m\boldsymbol{\theta}_k + (1-m)\boldsymbol{\theta}_q, \tag{20}$$

where $\boldsymbol{\theta}_k$ is the parameters of the key encoder $\boldsymbol{\theta}_q$ is parameters of the query encoder, and $m \in [0, 1)$ is the momentum coefficient. This means the key encoder evolves more slowly, providing more consistent representations for the keys in the dictionary.

Using the same encoder for queries and keys can lead to oscillations in training. The momentum encoder smooths out the updates, making the dictionary more stable. A large dictionary is crucial, but updating all keys with the query encoder for every batch is computationally expensive. The momentum encoder allows for a large dictionary without this overhead.

The model is trained with a contrastive loss function called InfoNCE [29]:

$$\mathcal{L}_q = -\log \frac{\exp(\langle q, k_+\rangle/\tau)}{\sum_{i=0}^{K} \exp(\langle q, k_i\rangle/\tau)}, \tag{21}$$

where q is encoded query representation, k_+ is encoded representation of the positive key (the matching image), k_i is encoded representations of the negative keys (other images in the batch), K is the number of negative samples, and τ is temperature parameter (controls the concentration of the distribution). This loss function encourages the model to maximize similarity between the query and its positive key (q and k_+) and minimize similarity between the query and the negative keys (q and k_-).

10.2. Limitation and Future Work

A limitation of CL-DPS is the need for back-propagating gradients through the likelihood estimator. Developing a more efficient and lightweight estimator is an area for future exploration. Currently, we tested CL-DPS using the MoCo framework, which may not be the optimal choice for CL in the context of blind inverse problems. Investigating alternative CL frameworks better suited to this task is another promising direction for future work.

10.3. Ablation Study

In this section, we conduct an ablation study to evaluate the impact of each component comprising CL-DPS. To do so, we incrementally introduce each component into the blind inverse process and perform qualitative and/or quantitative comparisons. Specifically, we analyze (i) the effect of random patch-size in Sec. 10.3.1, (ii) overlapped patch-wise inference and global average pooling in the Appendix, and (iii) the color consistency head, also in the Appendix.

10.3.1 Effect of the Size of Random Patches on CL-DPS

In this section, we examine the impact of the size of random patches when training the auxiliary DNN (random patches are illustrated by red rectangles in Fig. 2a). For this analysis, we remove the global average pooling and fully connected layer from the model, apply motion blur to corrupt the input images, and train the auxiliary DNN twice: once using a patch size of 128×128 and once with a patch size of 64×64. The results for the motion deblurring task using these two auxiliary DNNs are shown in Fig. 8.

As observed, the restored images using auxiliary DNN trained by large cropped patch size 128×128 (Fig. 8 column c) fail to produce the image with fine details. Such problem can be solved by reduce the patch size to 64×64 (see Fig. 8 column d).

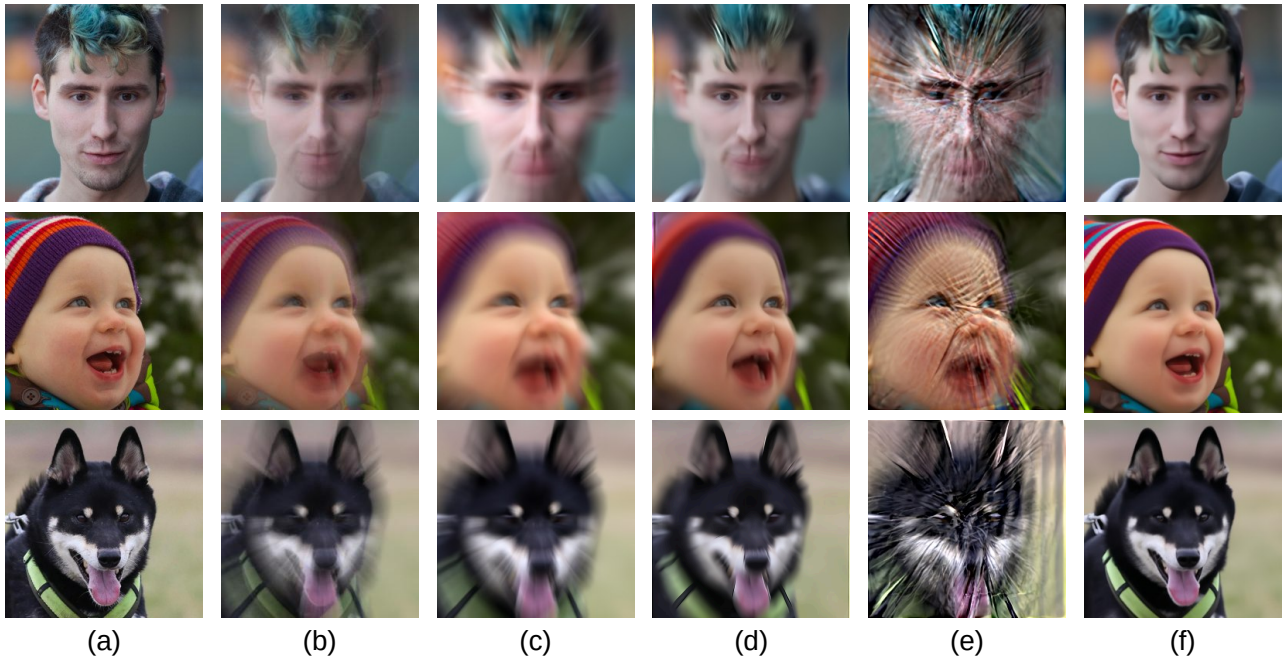


Figure 6. Results of blind zoom deblurring, a challenging **non-linear** inverse problem: (a) ground truth image, (b) zoom blurred measurement, and restored images using (c) BlindDPS [8], (d) FastEM [23], (e) GibbsDDRM [27], and (f) CL-DPS (ours). Notably, all methods fail catastrophically except for CL-DPS.

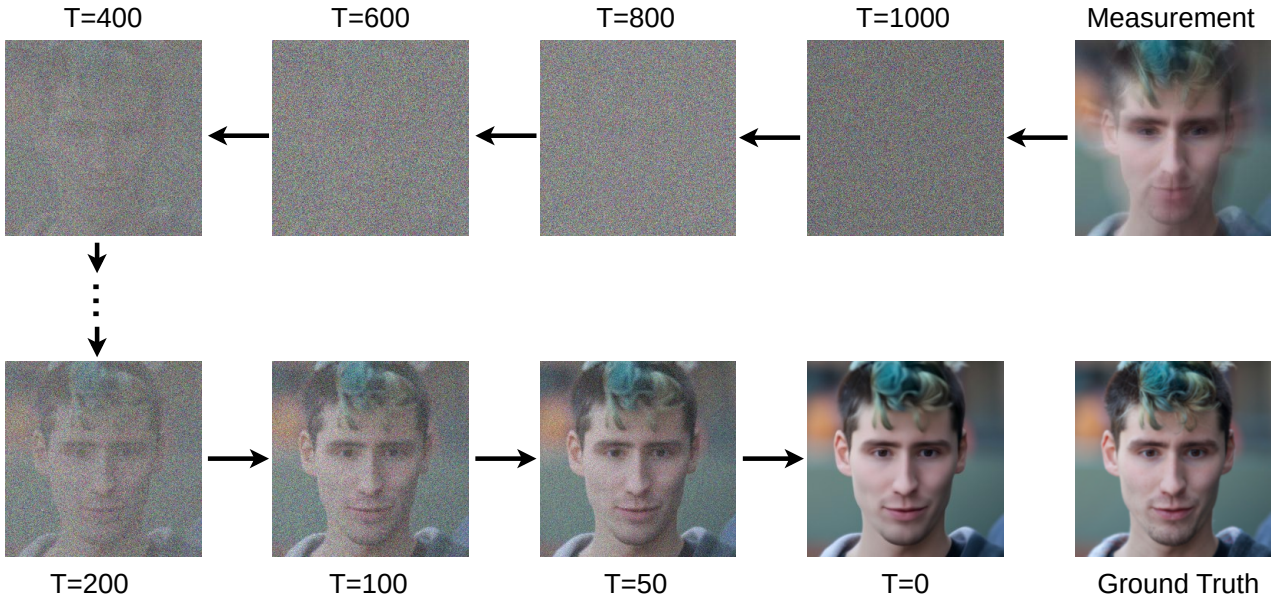


Figure 7. The CL-DPS process of recover the zoom blurred measurement.

Figure 8. Effect of patch size during auxiliary DNN training within the CL-DPS framework for the motion deblurring task: (a) original image, (b) measurement, (c) restored image using an auxiliary DNN trained with a patch size of 128×128, and (d) restored image using an auxiliary DNN trained with a patch size of 64×64.