Towards Coherent Image Inpainting Using Denoising Diffusion Implicit Models

Guanhua Zhang [*1] **Jiabao Ji** [*1] **Yang Zhang** [2] **Mo Yu** [3] **Tommi Jaakkola** [4] **Shiyu Chang** [1]

Abstract

Image inpainting refers to the task of generating a complete, natural image based on a partially revealed reference image. Recently, many research interests have been focused on addressing this problem using fixed diffusion models. These approaches typically directly replace the revealed region of the intermediate or final generated images with that of the reference image or its variants. However, since the unrevealed regions are not directly modified to match the context, it results in *incoherence* between revealed and unrevealed regions. To address the incoherence problem, a small number of methods introduce a rigorous Bayesian framework, but they tend to introduce mismatches between the generated and the reference images due to the approximation errors in computing the posterior distributions. In this paper, we propose CoPaint, which can coherently inpaint the whole image without introducing mismatches. CoPaint also uses the Bayesian framework to jointly modify both revealed and unrevealed regions, but approximates the posterior distribution in a way that allows the errors to gradually drop to zero throughout the denoising steps, thus strongly penalizing any mismatches with the reference image. Our experiments verify that CoPaint can outperform the existing diffusion-based methods under both objective and subjective metrics. The codes are available at https://github.com/UCSB-NLP-Chang/CoPaint/.

1. Introduction

Image inpainting refers to the problem of generating a natural, complete image based on a partially revealed reference image. In recent years, researchers have increasingly focused on using diffusion models, a class of generative models that convert noise images into natural images through a series of denoising steps, to solve this problem. One popular approach is to use a fixed, generic diffusion model that has been pre-trained for image generation. This eliminates the need for retraining the diffusion model, making the process more efficient and versatile.

(a) Input (b) Blended (c) Ddrm (d) CoPaint

Figure 1. Inpainted images by Blended (b), Ddrm (c) and our proposed method CoPaint-TT (d). Image are generated conditioned on the given masked input (a) with a fixed diffusion model.

However, despite their promising performance, such methods are susceptible to the *incoherence problem*. Specifically, these methods often impose the inpainting constraints based on some form of replacement operations, *e.g.*, directly replacing the revealed portion of the predicted image with that of the reference image (Wang et al., 2022), or replacing the revealed portion of the intermediate denoising results with a corrupted version of the reference images (Avrahami et al., 2021; Lugmayr et al., 2022). Yet the pixels of the unrevealed region, which should also be modified to match the context of the revealed region, are not directly modified (Trippe et al., 2022). As a result, these methods can easily lead to discontinuity or incoherence between the revealed and unrevealed regions in the generated images. For example, Figure 1 shows some incoherent inpainting results of a half-masked portrait image. The result in (b) has unmatched hair colors and styles between the left and right halves, and the result in (c) has a clear discontinuity in the middle resulting from different skin tones.

To address the incoherence problem, a small number of methods like Dps (Chung et al., 2022a) and Resampling (Trippe et al., 2022) use a more rigorous Bayesian framework, which casts the inpainting problem as sampling the images from the posterior distributions, conditional on the inpainting constraint. Since the posterior distribution

[*]Equal contribution [1]UC Santa Barbara [2]MIT-IBM Watson AI Lab [3]IBM Research during the project's involvement [4]MIT CSAIL. Correspondence to: Guanhua Zhang <guanhua@ucsb.edu>, Jiabao Ji <jiabaoji@ucsb.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

differs from the prior distribution in both the revealed and unrevealed pixels, these methods can ensure that the entire image is coherently modified. However, since the posterior distribution is often very hard to compute, these methods would resort to approximations or Monte Carlo methods, which would introduce errors in satisfying the inpainting constraints. In short, it remains an unresolved problem how to ensure coherence during generation while strictly enforcing inpainting constraints.

In this paper, we propose CoPaint, a simple inpainting algorithm that addresses the incoherence problem without violating the inpainting constraints. CoPaint also adopts the Bayesian framework to coherently modify the entire images but introduces a new solution to address the challenges in computing and sampling from the posterior distribution. Specifically, CoPaint derives an approximated posterior distribution for the intermediate images, whose maximum a posteriori (MAP) samples become equivalent to directly minimizing the errors in the inpainting constraint, referred to as the *inpainting errors*. To make the computation of the inpainting errors tractable at each intermediate denoising step, we use the one-step estimation of the final generated image instead of directly computing the final generation. Although this would introduce further approximation errors, we can show that the errors would gradually decrease as the denoising process proceeds. Notably, at the final step, all the approximation errors can be made zero.

Our experimental evaluations on `CelebA-HQ` and `ImageNet` with various shapes of the revealed region verify that CoPaint has better inpainting quality and coherence than existing diffusion-model-based approaches under both objective and subjective metrics. For example, CoPaint achieves an average of 19% relative reduction in `LPIPS` compared to RePaint (Lugmayr et al., 2022), our most competitive baseline, while consuming 31% less computation budget on `ImageNet` dataset.

2. Related Work

Image inpainting is a long-lasting research question in computer vision, aiming at completing a degraded image naturally and coherently (Xiang et al., 2022; Shah et al., 2022). In recent years, various deep learning techniques have been suggested for the task of inpainting (Reddy et al., 2022), with a majority of them built upon auto-encoder (Pathak et al., 2016; Vo et al., 2018; Liu et al., 2018; Iizuka et al., 2017; Song et al., 2018; Guo et al., 2019; Xiao et al., 2018; Hong et al., 2019; Nazeri et al., 2019; Liu et al., 2020), VAE (Zheng et al., 2019; Zhao et al., 2020; 2021; Peng et al., 2021), GAN (Pathak et al., 2016; Vo et al., 2018; Liu et al., 2018; Iizuka et al., 2017; Song et al., 2018; Guo et al., 2019; Xiao et al., 2018; Hong et al., 2019; Weng et al., 2022) or auto-regressive transformer (Yu et al., 2021; Wan et al., 2021) structures. Despite achieving notable successes in inpainting, these methods are primarily based on supervised learning, *i.e.*, the networks require to be trained on specific degradation types. As a result, these approaches require large computational resources and may not be well-suited for scenarios that were not encountered during training, leading to poor generalization performance (Xiang et al., 2022). More recently, diffusion model-based approaches are gaining increasing popularity due to their exceptional results in image generation (Sohl-Dickstein et al., 2015; Ho et al., 2020; Yang et al., 2022; Bond-Taylor et al., 2021; Chung et al., 2022b; Batzolis et al., 2022; Bansal et al., 2022; Liu et al., 2022; Ku et al., 2022; Benton et al., 2022; Horwitz & Hoshen, 2022; Horita et al., 2022; Li et al., 2022). Besides, these methods enjoy the advantage of being able to perform inpainting without the need for degradation-specific training (Song & Ermon, 2019a). In this section, we will review the current literature on diffusion-based inpainting. These methods can broadly be divided into two categories: supervised and unsupervised methods (Kawar et al., 2022).

Supervised diffusion inpainting Supervised diffusion inpainting approaches involve training a diffusion model for the specific task of inpainting, taking into account the particular degradation types. Palette (Saharia et al., 2021a;b) feeds the degraded image to the diffusion model at each time step of the diffusion process for training a diffusion inpainting model. Similar methods are also used by Glide (Nichol et al., 2021), where a text-conditional diffusion model is fine-tuned for the inpainting task. Latent Diffusion (Rombach et al., 2021) incorporates an autoencoding model for compressing the image space, and then the spatially aligned conditioning information is concatenated with the input of the model. By contrast, CCDF (Chung et al., 2021) adopts a non-expansive mapping for aggregating the degradation operation during training. A "predict-and-refine" conditional diffusion model is proposed by Whang et al. (2021), where a diffusion model is trained to refine the output of a deterministic predictor. However, all these methods require degradation-specific training, which could be computationally expensive and may not generalize well to unseen degradation operators.

Unsupervised diffusion inpainting Different from supervised methods, unsupervised diffusion inpainting aims at utilizing pre-trained diffusion models for the inpainting task without any model modification. Our proposed method also falls into this category. As an early work, Song & Ermon (2019a) proposes to modify the DDPM sampling process by spatially blending the noisy version of the degraded image in each time step of the denoising process. A similar idea is adopted by BlendedDiffusion for text-driven inpainting (Avrahami et al., 2021). DDRM (Kawar et al., 2022) defines a new posterior diffusion process whose marginal probability is proved to be consistent with DDPM (Ho et al., 2020). Roughly speaking, the proposed denoising process

is equivalent to blending the degraded image in a weighted-sum manner in each time step. Despite the high efficiency of these methods, the images generated by the simple blending-based methods are often not harmonizing in the recovered part (Lugmayr et al., 2022).

To address the issue, the authors of REPAINT (Lugmayr et al., 2022) proposed a resampling strategy. Specifically, a "time travel" operation is introduced, where images from the current t time step are first blended with the noisy version of the degraded image, and then used to generate images in the $t+1$ time step using a one-step forward process, thereby reducing the visual inconsistency caused by blending. Trippe et al. (2022) further proves that a simple blending-based method would introduce irreducible approximation error in the generation process. A particle filtering-based method, named RESAMPLING, is then proposed, where for time step t, each generated image is resampled based on its probability of generating the revealed part of the degraded image in the $t-1$ time step. Pokle et al. (2022) look at diffusion models in a deep equilibrium (DEQ) perspective and propose a DEQ method for inverting DDIM to save memory consumption. DDNM (Wang et al., 2022) introduces a new blending mechanism, where the degraded image is directly incorporated in each time step without noise. Another recent work DPS (Chung et al., 2022a) addresses the inpainting problem via approximation of the posterior sampling in a similar manner with classifier-free guided diffusion (Dhariwal & Nichol, 2021a). Specifically, they use the approximated gradient of the posterior likelihood as a mean shift for images generated at each time step of the denoising process. Different from these methods, we introduce a Bayesian framework to jointly modify both revealed and unrevealed parts of images by maximizing the posterior in each time step along the denoising process and thus enjoying better coherence for the inpainted part.

3. Background and Notations

In this section, we will provide a brief overview of the diffusion model frameworks and notations that will be used in this paper. Note that we will only cover just enough details for the purpose of explaining our proposed approach. We would recommend readers refer to the original papers cited for complete details and derivations.

Denote $\boldsymbol{X}_0$ as a random vector of the natural images (vectorized). DDIMs (Song et al., 2020) try to recover the distribution of $\boldsymbol{X}_0$ through a set of intermediate variables, *e.g.,* $\boldsymbol{X}_{1:T}$, which are progressively corrupted versions of $\boldsymbol{X}_0$. There are two processes in a DDIM framework, a *forward diffusion process*, which defines how $\boldsymbol{X}_0$ is corrupted into $\boldsymbol{X}_T$, and a *reverse denoising process*, which governs how to recover $\boldsymbol{X}_0$ from $\boldsymbol{X}_T$ based on the forward process.

The forward diffusion process of DDIMs follows that of the denoising diffusion probabilistic models (DDPMs) (Ho et al., 2020; Sohl-Dickstein et al., 2015), which is a Markov process that progressively adds Gaussian noises to the intermediate variables, *i.e.,*

$$\begin{aligned} q(\boldsymbol{X}_{1:T}|\boldsymbol{X}_0) &= \prod_{t=1}^{T} q(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}), \\ q(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}) &= \mathcal{N}(\boldsymbol{X}_t; \sqrt{\alpha_t}\boldsymbol{X}_{t-1}, \beta_t \boldsymbol{I}), \end{aligned} \tag{1}$$

where $\alpha_{1:T}$ and $\beta_{1:T}$ define the scaling and variance schedule with $\alpha_t = 1 - \beta_t$. It can be easily shown that, with an appropriate scaling and variance schedule and a sufficiently large T, $\boldsymbol{X}_T$ approaches the standard Gaussian distribution.

For the reverse diffusion process, DDIMs introduce another distribution q_σ, called the inference distribution, that has a matching conditional distribution of each individual intermediate variable to q. Specifically

$$\begin{aligned} q_\sigma(\boldsymbol{X}_{1:T}|\boldsymbol{X}_0) &= q_\sigma(\boldsymbol{X}_T|\boldsymbol{X}_0) \prod_{t=T}^{2} q_\sigma(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t, \boldsymbol{X}_0), \\ q_\sigma(\boldsymbol{X}_T|\boldsymbol{X}_0) &= \mathcal{N}(\boldsymbol{X}_T; \sqrt{\bar{\alpha}_T}\boldsymbol{X}_0, (1-\bar{\alpha}_T)\boldsymbol{I}), \\ q_\sigma(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t, \boldsymbol{X}_0) &= \mathcal{N}(\boldsymbol{X}_{t-1}; \boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}), \end{aligned} \tag{2}$$

where $\bar{\alpha} = \prod_{i=1}^{t} \alpha_i$ and σ_t^2 is a free hyperparameter, and

$$\boldsymbol{\mu}_t = \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{X}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\boldsymbol{X}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{X}_0}{\sqrt{1-\bar{\alpha}_t}}. \tag{3}$$

It can be shown that as long as $\sigma_t^2 \in [0, 1-\bar{\alpha}_t], \forall t$. q_σ and q have matching distributions: $q_\sigma(\boldsymbol{X}_t|\boldsymbol{X}_0) = q(\boldsymbol{X}_t|\boldsymbol{X}_0), \forall t$.

The denoising process is derived from q_σ by replacing $\boldsymbol{X}_0$ with an estimated value of $\boldsymbol{X}_0$, *i.e.,*

$$\begin{aligned} p_\theta(\boldsymbol{X}_T) &= \mathcal{N}(\boldsymbol{X}_T; \boldsymbol{0}, \boldsymbol{I}) \\ p_\theta(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t) &= q_\sigma(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t, \hat{\boldsymbol{X}}_0^{(t)}), \end{aligned} \tag{4}$$

where

$$\hat{\boldsymbol{X}}_0^{(t)} = \boldsymbol{f}_\theta^{(t)}(\boldsymbol{X}_t) \tag{5}$$

is produced by a (reparameterized) neural network that predicts $\boldsymbol{X}_0$ from $\boldsymbol{X}_t$ by minimizing the mean squared error.

Equation 5 provides a way of estimating the final generation as a deterministic function of $\tilde{\boldsymbol{X}}_t$. In particular, $\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)$ is generated by feeding to the inference network once, and thus can be regarded as a compute-efficient approximation of the final generation. We will refer to it as *one-step generation*. As shown in Figure 2 , the gap between $\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)$ and $\tilde{\boldsymbol{X}}_0$ typically gets smaller as t gets smaller. As we will show, one-step generation is central to our algorithm because it permits direct control over the final generation through the intermediate variables.

4. The COPAINT Algorithm

4.1. Problem Formulation

The image inpainting problem aims to generate a natural, complete image given a partially revealed image, such that

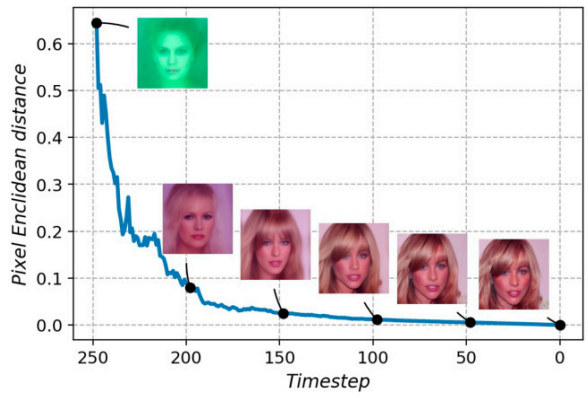

Figure 2. The trajectory of the gap between $\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)$ and $\tilde{\boldsymbol{X}}_0$ along the unconditional diffusion denoising process. We report the pixel-wise averaged Euclidean distance between the two.

the generated image is identical to the given image in the revealed regions. Formally, denote $\boldsymbol{r}(\cdot)$ as an operator that outputs a revealed subset of the input dimensions, and $\boldsymbol{s}_0$ as the revealed portion of the given reference image. Then the goal of image inpainting is to generate a natural image under the following inpainting constraint

$$\mathcal{C} : \boldsymbol{r}(\tilde{\boldsymbol{X}}_0) = \boldsymbol{s}_0, \tag{6}$$

which we denote as event $\mathcal{C}$ for notation brevity. In this paper, we focus on the scenario where the diffusion model is pretrained and fixed, *i.e.,* $\boldsymbol{f}_\theta^{(t)}$ is fixed for all t.

As discussed, many existing diffusion-model-based approaches only replace the revealed region of the generated intermediate or final images *i.e.,* $\boldsymbol{r}(\boldsymbol{X}_t)$ or $\boldsymbol{r}(\boldsymbol{X}_0)$, to directly impose the inpainting constraint, whereas the generation of the remaining unrevealed region is not directly modified to match the context. Thus the resulting generated images could easily suffer from incoherence between the revealed and unrevealed regions. In the following, we will explain how we propose to jointly optimize both regions.

4.2. A Prototype Approach

We will start with a prototype approach. Consider the simplest form of DDIM, where $\sigma_t = 0, \forall t$. In other words, the denoising process becomes a deterministic process with respect to $\tilde{\boldsymbol{X}}_T$. As a result, the inpainting constraint on $\tilde{\boldsymbol{X}}_0$ in Equation 6 can translate to that on $\tilde{\boldsymbol{X}}_T$, so the image inpainting problem boils down to determining an appropriate $\tilde{\boldsymbol{X}}_T$ based on the following posterior distribution:

$$\begin{aligned} p_\theta(\tilde{\boldsymbol{X}}_T|\mathcal{C}) \propto & p_\theta(\tilde{\boldsymbol{X}}_T) \cdot p_\theta\big(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0) = \boldsymbol{s}_0 \big| \tilde{\boldsymbol{X}}_T\big) \\ = & p_\theta(\tilde{\boldsymbol{X}}_T) \cdot \delta\big(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0) = \boldsymbol{s}_0\big). \end{aligned} \tag{7}$$

According to Equations 2 and 4, $p_\theta(\tilde{\boldsymbol{X}}_T)$ is a standard Gaussian distribution. To clarify, $p_\theta\big(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0) = \boldsymbol{s}_0 \big| \tilde{\boldsymbol{X}}_T\big)$ denotes the probability density function of $\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)$ evaluated at $\boldsymbol{s}_0$, conditional on the value of $\tilde{\boldsymbol{X}}_T$. Since $\tilde{\boldsymbol{X}}_T$ is given and $\tilde{\boldsymbol{X}}_0$ is a deterministic function of $\tilde{\boldsymbol{X}}_T$, $p_\theta\big(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0) = \boldsymbol{s}_0 \big| \tilde{\boldsymbol{X}}_T\big)$ becomes a dirac delta function $\delta(\cdot)$, with infinity probability density at where the event holds, and zero density elsewhere. The dirac delta function can be approximated by a Gaussian density function with zero variance. Therefore, Equation 7, after taking the logarithm, can be approximated as

$$\begin{aligned} & \log p_\theta(\tilde{\boldsymbol{X}}_T|\mathcal{C}) \\ \approx & -\frac{1}{2}\|\tilde{\boldsymbol{X}}_T\|_2^2 - \frac{1}{2\xi_T^2}\big\|\boldsymbol{s}_0 - \boldsymbol{r}(\tilde{\boldsymbol{X}}_0)\big\|_2^2 + C \\ \approx & -\frac{1}{2}\|\tilde{\boldsymbol{X}}_T\|_2^2 - \frac{1}{2\xi_T^2}\big\|\boldsymbol{s}_0 - \boldsymbol{r}(\boldsymbol{g}_\theta(\tilde{\boldsymbol{X}}_T))\big\|_2^2 + C, \end{aligned} \tag{8}$$

where we denote $\tilde{\boldsymbol{X}}_0 = \boldsymbol{g}_\theta(\tilde{\boldsymbol{X}}_T)$ to emphasize $\tilde{\boldsymbol{X}}_0$ is a function of $\tilde{\boldsymbol{X}}_T$; C is the normalizing constant; ξ_T is the standard deviation of the second Gaussian distribution. When ξ_T approaches zero, the approximation in Equation 8 becomes exact. In practice, ξ_T can be set to a very small value.

Equation 8 provides a justification for solving $\tilde{\boldsymbol{X}}_T$ using optimization method, because the first term can be regarded as a prior regularization and the second term as a penalty term enforcing the inpainting constraint. One can either perform gradient ascent over $\tilde{\boldsymbol{X}}_T$ to find the maximum a posteriori (MAP) estimate of $\tilde{\boldsymbol{X}}_T$, or apply gradient-based sampling techniques such as Hamiltonian Markov Chain Monte Carlo (MCMC) (Neal, 2011). to draw random samples. Note that the optimization is over the entire $\tilde{\boldsymbol{X}}_T$, not just the revealed regions, so this would ideally resolve the incoherence problem in the existing replacement methods. Since the weight on the second term is very large, we can expect to solve for an $\tilde{\boldsymbol{X}}_T$ that can satisfy the inpainting constraint very well.

4.3. One-Step Approximation

The key limitation of the aforementioned prototype approach is that it is computationally impractical, because evaluating the final generation $\boldsymbol{g}_\theta(\tilde{\boldsymbol{X}}_T)$ and computing its gradient involve performing forward and reverse propagation through the entire DDIM denoising process, which typically consists of tens or even hundreds of denoising steps. We thus need to derive a computationally-feasible algorithm from the prototype approach.

As discussed in Section 3, the one-step generation $\boldsymbol{f}_\theta^{(T)}(\tilde{\boldsymbol{X}}_T)$ offers a fast approximation of the final generation, so a straightforward modification is to replace the $\boldsymbol{g}_\theta(\tilde{\boldsymbol{X}}_T)$ in Equation 8 with $\boldsymbol{f}_\theta^{(T)}(\tilde{\boldsymbol{X}}_T)$.

Formally, we introduce a approximated conditional distribution of $\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)$ given $\tilde{\boldsymbol{X}}_T$, denoted as $p'_\theta(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)|\tilde{\boldsymbol{X}}_T)$, which is centered around the one-step generated value, $\boldsymbol{r}(\boldsymbol{f}_\theta^{(T)}(\tilde{\boldsymbol{X}}_T))$, plus a Gaussian error, *i.e.*,

$$p'_\theta(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)|\tilde{\boldsymbol{X}}_T) = \mathcal{N}\big(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0); \boldsymbol{r}(\boldsymbol{f}_\theta^{(T)}(\tilde{\boldsymbol{X}}_T)), \xi_T'^2 \boldsymbol{I}\big), \tag{9}$$

Algorithm 1 CoPaint-TT

Input: $\boldsymbol{s}_0$, $\{f_\theta^{(t)}(\cdot)\}_{t=1}^T$, time travel interval τ and frequency K, gradient descent number G and learning rate $\{\eta_t\}_{t=1}^T$
Initialize $\tilde{\boldsymbol{X}}_T \sim \mathcal{N}(0, \mathbf{I})$
$t \leftarrow T, k \leftarrow K$
while $t \neq 0$ **do**
  Optimize $\tilde{\boldsymbol{X}}_t$ to maximize Equations 10 14 by G-step gradient descent with learning rate η_t
  Generate $\tilde{\boldsymbol{X}}_{t-1}$ with Equation 4
  $t \leftarrow t-1$
  if $t \bmod \tau = 0$ and $t \leq T - \tau$ **then**
    if $k > 0$ **then**
      // time trave l
      Generate $\tilde{\boldsymbol{X}}_{t+\tau} \sim q(\tilde{\boldsymbol{X}}_{t+\tau}|\tilde{\boldsymbol{X}}_t)$
      $t \leftarrow t+\tau-1, k \leftarrow k-1$
    else
      $k \leftarrow K$
    end if
  end if
end while
Return: $\tilde{\boldsymbol{X}}_0$

where ξ_T' is the standard deviation parameter. Plugged in this approximated distribution, the approximate posterior is

$$\begin{aligned}&\log p'_\theta(\tilde{\boldsymbol{X}}_T|\mathcal{C})\\ =&\log(p_\theta(\tilde{\boldsymbol{X}}_T)) + \log\left(p'_\theta\big(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)=\boldsymbol{s}_0\big|\tilde{\boldsymbol{X}}_T\big)\right) + C'\\ =&-\frac{1}{2}\|\tilde{\boldsymbol{X}}_T\|_2^2 - \frac{1}{2\xi_T'^2}\big\|\boldsymbol{s}_0 - \boldsymbol{r}(\boldsymbol{f}_\theta^{(T)}(\tilde{\boldsymbol{X}}_T))\big\|_2^2 + C',\end{aligned} \tag{10}$$

where C' refers to any normalizing constant, and the last line is derived from Equation 9.

It can be easily shown that in order to minimize the approximation gap, *i.e.,* the KL divergence between $p_\theta(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)|\tilde{\boldsymbol{X}}_T)$ and $p'_\theta(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)|\tilde{\boldsymbol{X}}_T)$, $\xi_T'^2$ should be set to

$$\xi_T'^2 = \frac{1}{N}\mathbb{E}_{p_\theta}\left[\big\|\boldsymbol{r}(\boldsymbol{f}_\theta^{(T)}(\tilde{\boldsymbol{X}}_T)) - \boldsymbol{r}(\tilde{\boldsymbol{X}}_0)\big\|_2^2\right], \tag{11}$$

where N is the dimension of $\boldsymbol{s}_0$. Similar to Equation 8, maximizing Equation 10 over $\tilde{\boldsymbol{X}}_T$ is essentially trying to satisfy the (approximated) inpainting constraint (second term) regularized by its prior (first term). However, in contrast to the exact case in Equation 8, where ξ_T should be as small as possible, ξ_T' should be large enough (Equation 11) to capture the approximation error, which leads to a smaller weight on the approximate inpainting constraint term in Equation 10.

4.4. Denoising Successive Correction

Equation 10 will push revealed part of the one-step approximated generation, $\boldsymbol{r}(\boldsymbol{f}_\theta^{(T)}(\tilde{\boldsymbol{X}}_T))$, towards the reference image $\boldsymbol{s}_0$. However, the actual inpainting constraint requires us to push the actual final generation, $\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)$, to $\boldsymbol{s}_0$. As a result, optimizing Equation 10 cannot exactly satisfy the inpainting constraint. To further enforce the inpainting constraint, we return to the *non-deterministic* DDIM procedure, where $\sigma_t \neq 0$, and apply the optimization technique discussed in Sections 4.2 and 4.3 to all the intermediate variables to successively correct the approximation error.

The proposed DDIM procedure samples $\tilde{\boldsymbol{X}}_{0:T}$ from the approximate posterior $p'_\theta(\tilde{\boldsymbol{X}}_{0:T}|\mathcal{C})$, which is decomposed as

$$p'_\theta(\tilde{\boldsymbol{X}}_{0:T}|\mathcal{C}) = p'_\theta(\tilde{\boldsymbol{X}}_T|\mathcal{C})\prod_{t=1}^T p'_\theta(\tilde{\boldsymbol{X}}_{t-1}|\tilde{\boldsymbol{X}}_t, \mathcal{C}). \tag{12}$$

$p'_\theta(\tilde{\boldsymbol{X}}_T|\mathcal{C}_T)$ is defined in Equation 10. To compute $p'_\theta(\tilde{\boldsymbol{X}}_{t-1}|\tilde{\boldsymbol{X}}_t, \mathcal{C})$, we introduce a set of Gaussian approximated distributions similar to Equation 9 as

$$p'_\theta(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)|\tilde{\boldsymbol{X}}_t) = \mathcal{N}\big(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0); \boldsymbol{r}(\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)), \xi_t'^2\boldsymbol{I}\big), \tag{13}$$

where ξ_t^2 is defined similar to Equation 11 (replacing T with t) to minimize the one-step approximation error. Then $p'_\theta(\tilde{\boldsymbol{X}}_{t-1}|\tilde{\boldsymbol{X}}_t, \mathcal{C})$ can be computed as

$$\begin{aligned}&\log p'_\theta(\tilde{\boldsymbol{X}}_{t-1}|\tilde{\boldsymbol{X}}_t, \mathcal{C})\\ =&\log p_\theta(\tilde{\boldsymbol{X}}_{t-1}|\tilde{\boldsymbol{X}}_t) + \log p'_\theta\big(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)=\boldsymbol{s}_0|\tilde{\boldsymbol{X}}_{t-1}, \tilde{\boldsymbol{X}}_t\big) + C'\\ =&\log p_\theta(\tilde{\boldsymbol{X}}_{t-1}|\tilde{\boldsymbol{X}}_t) + \log p'_\theta\big(\boldsymbol{r}(\tilde{\boldsymbol{X}}_0)=\boldsymbol{s}_0|\tilde{\boldsymbol{X}}_{t-1}\big) + C'\\ =&-\frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{X}}_{t-1} - \tilde{\boldsymbol{\mu}}_t\|_2^2 - \frac{1}{2\xi_{t-1}'^2}\big\|\boldsymbol{s}_0 - \boldsymbol{r}\big(\boldsymbol{f}_\theta^{(t-1)}(\tilde{\boldsymbol{X}}_{t-1})\big)\big\|_2^2\\ &+C',\end{aligned} \tag{14}$$

where the third line follows from the reverse Markov property of the DDIM denoising process. The first term in the last line follows from Equations 2 to 4, with

$$\tilde{\boldsymbol{\mu}}_t = \sqrt{\alpha_{t-1}}f_\theta^{(t)}(\tilde{\boldsymbol{X}}_t) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\frac{\tilde{\boldsymbol{X}}_t - \sqrt{\alpha_t}f_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)}{\sqrt{1-\alpha_t}}. \tag{15}$$

To generate the final inpainting result, we follow the following *greedy* optimization procedure to find samples of $\tilde{\boldsymbol{X}}_{0:T}$ that maximizes the $p'_\theta(\tilde{\boldsymbol{X}}_{0:T}|\mathcal{C})$ in Equation 12. *First*, we sample an $\tilde{\boldsymbol{X}}_T$ by optimizing Equation 10. *Second*, given the generated value of $\tilde{\boldsymbol{X}}_t$, we sample an $\tilde{\boldsymbol{X}}_{t-1}$ by optimizing Equation 14. Both steps are essentially enforcing the approximate inpainting constraints under the DDIM prior regularization. According to Figure 2, the one-step approximation error will gradually reduce as t decreases, so the algorithm would approach the inpainting constraint with increasing levels of exactness, successively correcting the approximation errors made in the previous steps. In particular, when $t = 1$, if we set $\sigma_1 = 0$ and let ξ_1 approach zero, we will have zero approximation error, *i.e.* $\boldsymbol{f}_\theta^{(1)}(\tilde{\boldsymbol{X}}_1) = \tilde{\boldsymbol{X}}_0$, so the generated image can be made to satisfy the inpainting constraint with very small errors.

4.5. Additional Algorithmic Designs

Although our algorithm can eventually eliminate the one-step approximation error in the final denoising step, the error in the early denoising steps can still affect the generation quality because it affects the quality of the prior distribution for subsequent generations. We introduce additional optional designs to reduce the approximation error.

Multi-Step Approximation In the early denoising steps where the approximation error is more significant, we can replace the one-step approximation with multi-step approximation, where $\tilde{\boldsymbol{X}}_0$ is approximated by going through multiple deterministic denoising steps at a subset of time steps.

Time Travel To improve the self-consistency of the intermediate examples, we can apply the time travel technique (Lugmayr et al., 2022; Wang et al., 2022), which periodically returns to the previous denoising steps by corrupting the intermediate images. Specifically, for a set of selected time steps ϕ at denoising time step $T-\tau$, instead of progressing to $T-\tau-1$, we rewind to time $T-1$ by sampling a new $\tilde{\boldsymbol{X}}_{T-1}$ based on $q(\tilde{\boldsymbol{X}}_{T-1}|\tilde{\boldsymbol{X}}_{t-\tau})$, and repeat the denoising steps from there. After K rounds of rewinding and denoising through steps $T-1$ to $T-\tau$, we then enter K rounds of rewinding and denoising loop through steps $T-\tau-1$ to $T-2\tau$. This process progresses until time zero is reached. The algorithm of CoPaint with time travel, abbreviated as CoPaint-TT, is shown in Algorithm 1.

5. Experiments

5.1. Experiment Setup

Datasets and models Following Lugmayr et al. (2022), we validate our method on two commonly used image datasets: `CelebA-HQ` (Liu et al., 2014) and `ImageNet-1K` (Russakovsky et al., 2015). `CelebA-HQ` contains more than 200K celebrity images, and we use the data split provided by Suvorov et al. (2022) following Lugmayr et al. (2022). `ImageNet-1K` is a large-scale image dataset containing 1000 categories, and the original data split is used (Russakovsky et al., 2015). Since not all images in the datasets are square-shaped images that diffusion models accept, we crop all images into 256×256 size to accommodate pretrained diffusion models. For `CelebA-HQ` dataset, we use the diffusion model pretrained by Lugmayr et al. (2022). For `ImageNet`, we use the model pretrained by Dhariwal & Nichol (2021b). We use the first five images in the validation set for hyperparameter selection. The first 100 images in test sets are used for evaluation following Lugmayr et al. (2022). Following Lugmayr et al. (2022); Wang et al. (2022); Suvorov et al. (2022), we consider seven different degradation masks on the original images for recovering: *Expand*, *Half*, *Altern*, *S.R.*, *Narrow*, *Wide*, and *Texts*. Examples of the degraded images are in Figure 4.

Metrics We evaluate the quality of the inpainting results using both *objective* and *subjective* metrics. For the objective metric, we adopt the `LPIPS` used in Lugmayr et al. (2022), which computes the similarity of two images in the feature space of AlexNet (Krizhevsky, 2014). For each reference image, we generate two inpainted images and the overall average `LPIPS` is reported. For the subjective metrics, we conduct a human evaluation on Amazon MTurk, where each subject is presented with a masked reference image and a pair of inpainted images, one by CoPaint-TT and the other by one of the baselines. The subject is then asked to select which one is of better quality according to a set of prespecified criteria. We also introduce a third option, 'cannot tell the difference', if the subject cannot find any noticeable differences between the pair.

We perform two tests where different criteria are specified. In the first test, referred to as `overall`, three criteria are introduced: 1) the inpainted image should be natural and without artifact; 2) the revealed portion should resemble the reference image; and 3) the image should be coherent. In the second test, referred to as `coherence`, only the coherence criterion is introduced. For both tests, we randomly sample 50 images for every mask in `CelebA-HQ` and `ImageNet` and thus result in $2 \times 2 \times 7 \times 50 = 1400$ image pairs for comparison. In each comparison with one baseline, we use the `vote difference (%)`, which is the percentage of the votes for CoPaint-TT subtracted by that for the baseline, as the metric for the relative inpainting quality compared to the baseline. More details about the human evaluation design could be seen in Appendix A.1.

Baselines and implementation details We focus on comparison with diffusion-model-based methods, which have been shown to achieve state-of-the-art performance over methods that do not use diffusion models (Lugmayr et al., 2022). Specifically, the following baselines are introduced: Blended (Song & Ermon, 2019a; Avrahami et al., 2021), DDRM (Kawar et al., 2022), RESAMPLING (Trippe et al., 2022), RePaint (Lugmayr et al., 2022), DPS (Chung et al., 2022a), and DDNM (Wang et al., 2022). A brief introduction about these baselines could be found in Section 2.

For all methods, we set the number of reverse sampling steps as 250 if not specified otherwise. For RePaint, we use their released codes[1] out-of-the-shelf with exactly the same setting as reported in their paper (Lugmayr et al., 2022). We then implement all other methods based on the RePaint code base and keep all hyper-parameters the same as the corresponding papers, details could be seen in Appendix A.2. Specifically, we set gradient descent step number $G = 2$ for both CoPaint and CoPaint-TT. A time-efficient version of our method, CoPaint-Fast is further introduced with $G = 1$ and reverse sampling

[1]With the released code of RePaint in `shorturl.at/AHILU` and the matching configurations, we noticed there is a slight gap between our implemented results and the reported ones in Lugmayr et al. (2022). Nevertheless, we believe our comparison with RePaint is fair because our methods were implemented based on the same code base, so any configuration nuances that can account for the gap are likely to affect the performance of our methods in the same direction.

Table 1. Quantitative results on CelebA-HQ(*top*) and ImageNet (*bottom*). We report the objective metric LPIPS and subjective human vote difference score of each baseline compared with our method CoPaint-TT. Lower is better for both metrics. The *vote difference* scores are calculated as the vote percentage of CoPaint-TT minus vote percentage of certain baseline. We report the results of two human tests, *i.e.*, overall and coherence, in the Vote(%) column separated by /, where overall is based on naturalness, restoration quality and coherence, while coherence is only based coherence of the generated image. vote difference score being lower than zero indicates certain baseline is better than our method CoPaint-TT. Numbers marked in blue are additional results.

Method	*Expand*		*Half*		*Altern*		*S.R.*		*Narrow*		*Wide*		*Text*		**Average**	
	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓
CelebA-HQ																
Blended	0.557	82/80	0.228	64/72	0.047	12/30	0.269	78/86	0.078	54/64	0.102	46/58	0.011	18/12	0.185	51/57
DDRM	0.704	94/98	0.273	86/96	0.151	78/84	0.596	100/100	0.140	76/84	0.125	84/62	0.028	38/42	0.288	79/81
Resampling	0.536	60/66	0.231	68/88	0.050	24/46	0.261	64/72	0.077	50/64	0.102	40/50	0.013	**-12**/8	0.181	42/56
RePaint	0.496	24/18	0.199	2/12	**0.014**	**-32**/38	0.041	10/10	0.039	4/10	0.072	**-16/-32**	**0.006**	4/**-14**	0.124	**0**/6
DPS	**0.449**	**-16/-12**	0.261	28/32	0.166	58/72	0.182	60/82	0.160	72/52	0.181	30/28	0.152	58/60	0.222	41/45
DDNM	0.598	76/94	0.257	84/72	0.015	-2/**-2**	0.046	6/**0**	0.071	14/38	0.111	28/60	0.014	**-12**/10	0.158	27/39
CoPaint-Fast	0.483	10/34	0.203	44/20	0.057	10/2	0.084	20/6	0.068	16/10	0.096	20/4	0.036	14/-4	0.147	13/11
CoPaint	0.472	12/20	0.188	40/24	0.016	-6/-4	0.033	22/-4	0.040	20/14	0.071	24/-2	0.007	-12/-4	0.118	15/6
CoPaint-TT	0.464	0/0	**0.180**	**0/0**	**0.014**	0/0	**0.028**	**0/0**	**0.037**	**0/0**	**0.069**	0/0	**0.006**	0/0	**0.114**	**0/0**
ImageNet																
Blended	0.717	39/36	0.366	72/80	0.277	96/92	0.686	94/96	0.161	76/64	0.194	62/60	0.028	8/26	0.347	64/65
DDRM	0.730	58/44	0.385	78/64	0.439	92/100	0.822	92/100	0.211	84/84	0.231	86/72	0.060	32/44	0.411	75/71
Resampling	0.704	38/40	0.353	58/86	0.259	72/88	0.624	94/98	0.151	66/64	0.183	76/66	0.028	22/26	0.329	61/67
RePaint	0.706	36/36	0.323	4/24	0.103	50/22	0.209	70/66	**0.072**	32/2	0.156	24/36	0.014	22/18	0.226	34/29
DPS	0.673	38/44	0.512	82/72	0.474	100/100	0.511	96/95	0.447	94/98	0.468	96/92	0.438	92/96	0.503	87/86
DDNM	0.805	34/76	0.408	68/64	0.051	12/12	0.107	18/36	0.101	50/70	0.185	48/60	**0.012**	**-2/-20**	0.238	33/44
CoPaint-Fast	0.678	14/26	0.335	22/24	0.075	10/6	0.128	36/28	0.103	26/22	0.167	24/32	0.043	6/-2	0.218	15/19
CoPaint	0.640	-2/8	0.307	6/0	0.041	22/4	**0.069**	20/18	0.078	24/30	0.138	14/16	0.017	2/-10	0.184	12/9
CoPaint-TT	**0.636**	**0/0**	**0.294**	**0/0**	**0.039**	**0/0**	**0.069**	**0/0**	0.074	**0/0**	**0.133**	**0/0**	0.015	0/0	**0.180**	**0/0**

step number as 100. We adopt an adaptive learning rate as $\eta_t = 0.02\sqrt{\bar{\alpha}_t}$ for all our methods. The rationale for such a learning rate setting can be seen in Appendix A.3. For better efficiency, we simply set $\xi_t'^2 = (1/1.012)^{T-t}$ instead of calculating it, inspired by the empirical observation that $\{\xi_t\}$ is increasing along t in Figure 2. For CoPaint-TT, we use time travel interval $\tau = 10$ and travel frequency $K = 1$. The ablation studies for the hyper-parameters could be seen in Section 5.4. Note that all methods use the same pretrained diffusion models without any modification.

5.2. Experiment Results

Quantitative results Table 1 shows the quantitative results of the proposed CoPaint-Fast, CoPaint and CoPaint-TT together with all other baselines on both CelebA-HQ (*top*) and ImageNet (*bottom*) datasets with seven mask types. The results in the Votes (%) column show the two vote difference scores, the first for overall test and the second for the coherence test. Here are our key observations. First, in terms of the objective metric, CoPaint consistently outperforms the other baselines, and reduces the average LPIPS score by 5% and 19% beyond the best-performing baseline RePaint in CelebA-HQ and ImageNet dataset, respectively. Second, when combined with time travel, CoPaint-TT can further bring down the average LPIPS score by another 3% and 1% in the two datasets, respectively. Besides, CoPaint-TT achieves the best performance among eleven out of the fourteen inpainting tasks while achieving comparable performances with the best baseline in the rest. Third, in terms of subjective evaluations, CoPaint-TT consistently produces positive vote difference scores in both the overall and coherence tests in most of the comparisons, indicating that the images generated by our method are not only more coherent, but also considered superior in terms of other aspects as well, including naturalness and meeting the inpainting constraint. Also, notice that the performance advantage of CoPaint-TT is generally more significant on ImageNet, which may be because images in ImageNet are more complex and thus any imperfections in the images, including incoherence, would be more conspicuous.

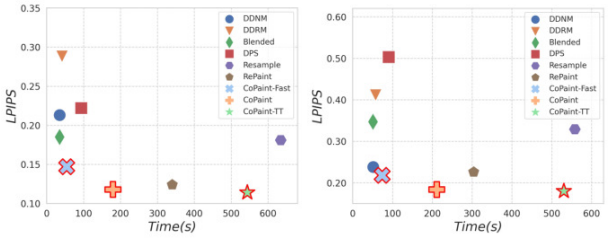

Figure 3. Time-performance trade-off on CelebA-HQ (*left*) and ImageNet (*right*). The x-axis indicates the average time (↓) to process one image, and the y-axis is the average LPIPS (↓).

We further conduct an additional experiment on inpainting high-resolution images in Appendix C, where our methods still achieve the best performance compared with other baselines with competitive time efficiency. Besides, the proposed method could also be used in other image restoration tasks. An additional experiment on the super-resolution task could be seen in Appendix D, where our methods show consistent superiority over other baselines.

Time-performance trade-off Figure 3 shows the running time of the proposed methods with other baselines on both CelebA-HQ (*left*) and ImageNet (*right*). In each subfigure, the x-axis denotes the average running time of each

Figure 4. Qualitative results of baselines and ours (COPAINT, COPAINT-TT) on `CelebA-HQ` with seven degradation masks.

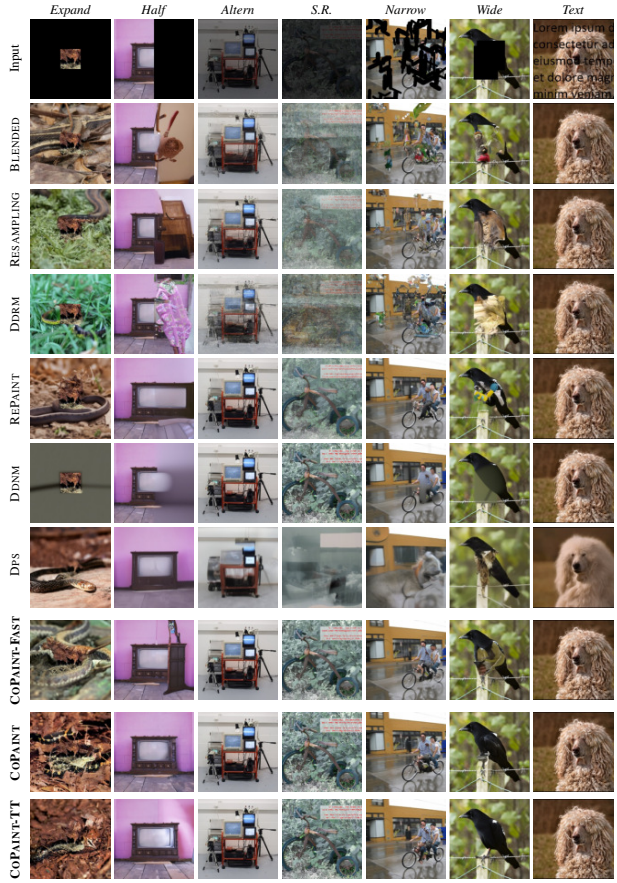

Figure 5. Qualitative results of baselines and ours (COPAINT, COPAINT-TT) on `ImageNet` with seven degradation masks.

method for processing one image, while the y-axis represents the average `LPIPS` score over seven mask types. The position closer to the left-bottom corner of the figure indicates better performance and time efficiency. COPAINT-TT achieves the best performance, although it has a larger computational cost than most baselines. On the other hand, with almost comparable performance, COPAINT reduces the time cost by nearly 60% in both datasets. Compared with the best-performing baseline REPAINT, COPAINT lies to the left-bottom of REPAINT in both datasets, demonstrating its advantage in the time-efficiency tradeoff. Moreover, we show that COPAINT-FAST is four times faster than COPAINT and is comparable to other baseline methods in terms of running time. COPAINT-FAST also achieves competitive performances in both two datasets. Specifically, COPAINT-FAST outperforms other baselines except for REPAINT in `CelebA-HQ` and beats all baselines in `ImageNet`.

Qualitative results We show some example generated images `CelebA-HQ` and `ImageNet` in Figures 4 and 5, respectively. More qualitative results with large size could

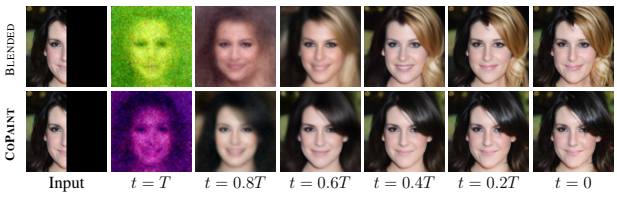

Figure 6. Coherence study of baseline BLENDED and our methods COPAINT on `CelebA-HQ` dataset with *Half* mask.

be seen in Appendix B. There are two key observations. First, our method achieves better coherence compared with other baselines, which is particularly significant with larger masks, such as *Expand* and *Half*. For example, in the second column in Figure 5 with the *Half* mask, the revealed part of the input is half of a television, as shown in the first row. In contrast to the failed completions generated by most baselines, both COPAINT and COPAINT-TT successfully generate a television with matching size and style. COPAINT-FAST shows slight performance degradation due to the limited number of reverse sampling and gradient de-

Table 2. Ablation study of the gradient descent number G, the time travel frequency K, the time travel interval τ, and the step number for approximating $\tilde{\boldsymbol{X}}_0$ in each time step H. The results are based on the testing set of CelebA-HQ dataset with *Half* mask.

Method	G	LPIPS↓	Time (s)↓
CoPaint-TT	1	0.187	326
	2	0.180	562
	5	0.192	1365
Method	K	LPIPS↓	Time (s)↓
CoPaint-TT	1	0.180	562
	2	0.179	721
	5	0.181	1428
Method	τ	LPIPS↓	Time (s)↓
CoPaint-TT	2	0.186	567
	5	0.178	569
	10	0.180	562
	20	0.181	564
Method	H	LPIPS↓	Time (s)↓
CoPaint-TT	1	0.180	562
	2	0.176	1491
	5	0.177	3346

scent. Second, although some baselines, such as DPS, also generate relatively coherent images, our methods produce more realistic images. For example, the televisions generated by our methods have more decorations and grains, while the television generated by DPS appears smooth and lacks details.

5.3. Coherence Study

To show how CoPaint ensures coherence along the denoising process, we present a coherence study, where we plot one-step generations over time steps $t = \{T,\ 0.8T,\ 0.6T, 0.4T, 0.2T, 1\}$ for the baseline Blended and our method CoPaint in Figure 6. As can be observed, although the revealed part is a woman with black hair, Blended keeps generating blond hair for the woman. This is consistent with the known bias in CelebA-HQ dataset, that women are more correlated with blond hair (Liu et al., 2021). The problem is that directly replacing the revealed portion of the image along the denoising process does not require the unrevealed portion to be consistent with the context of the revealed region. By contrast, our method could effectively generate a coherent image with black hair.

5.4. Ablation Study

We investigate the design choices of three hyperparameters, gradient descent step number G, time travel frequency K, time travel interval τ, and the effects of multi-step approximation as mentioned in Section 4. Specifically, we conduct our experiments on the CelebA-HQ with *Half* mask. The results could be seen in Table 2.

As shown in Algorithm 1, a G-step gradient descent method is adopted for optimizing $\tilde{\boldsymbol{X}}_t$ at each time step. In Table 2 (*first*), we see that a larger G would not always introduce better performances CoPaint-TT. As we optimize $\tilde{\boldsymbol{X}}_t$ to minimize the mean square error (corresponding to the second term in Equation 14) only in the revealed region, a larger gradient descent number may introduce an overfitting problem and thus lead the poor performances.

Table 2 (*second* and *third*) shows the effects of time travel frequency K and interval τ. Different from RePaint (Lugmayr et al., 2022) where $K = 9$ is used, we see that $K = 1$ is sufficient for our method, demonstrating that our proposed method is better at imposing the inpainting constraints than the simple replacement operations adopted by RePaint. Besides, we show that the value of time travel interval τ does not have a significant impact on the performance with $\tau \geq 5$.

As we have mentioned in Section 4.5, the one-step approximation for $\tilde{\boldsymbol{X}}_0$ could be replaced with multi-step approximation by going through multiple deterministic denoising steps at a subset of time steps. We denote the approximation step number as H, and its effects could be seen in Table 2 (*fourth*). We see that with a minor decrease in LPIPS, the time cost dramatically increases. With $H = 5$, it takes about six times longer than $H = 1$ for processing one image. We leave it for our future work to explore how to improve computational efficiency for multi-step approximation.

6. Conclusion

In this paper, we proposed a diffusion-based image inpainting method, CoPaint, which introduces a Bayesian framework to jointly modify both revealed and unrevealed parts of intermediate variables in each time step along the denoising process, leading to better coherence in the inpainted image. CoPaint's approximation error of the posterior distribution is designed to gradually drop to zero, thus strongly enforcing the inpainting constraint. Results from extensive experiments showed that CoPaint outperforms existing diffusion-based methods in both objective and subjective metrics in terms of coherence and overall quality. However, there are still some imperfections in CoPaint, due to the suboptimal greedy optimization and one-step approximation error. See the failure case study in Appendix E and the discussion about potential societal impacts in Appendix F. In the next step, we plan to replace our greedy optimization with more plausible sampling methods and investigate ways to further reduce approximation error.

References

Avrahami, O., Lischinski, D., and Fried, O. Blended diffusion for text-driven editing of natural images. *2022*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18187–18197, 2021.

Bansal, A., Borgnia, E., Chu, H.-M., Li, J., Kazemi, H., Huang, F., Goldblum, M., Geiping, J., and Goldstein, T. Cold diffusion: Inverting arbitrary image transforms without noise. *ArXiv*, abs/2208.09392, 2022.

Batzolis, G., Stanczuk, J., Schonlieb, C.-B., and Etmann, C. Non-uniform diffusion models. *ArXiv*, abs/2207.09786, 2022.

Benton, J., Shi, Y., Bortoli, V. D., Deligiannidis, G., and Doucet, A. From denoising diffusions to denoising markov models. *ArXiv*, abs/2211.03595, 2022.

Bond-Taylor, S., Hessey, P., Sasaki, H., Breckon, T., and Willcocks, C. G. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In *European Conference on Computer Vision*, 2021.

Chung, H., Sim, B., and Ye, J.-C. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12403–12412, 2021.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022a.

Chung, H., Sim, B., Ryu, D., and Ye, J. C. Improving diffusion models for inverse problems using manifold constraints. *ArXiv*, abs/2206.00941, 2022b.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021a.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021b.

Guo, Z., Chen, Z., Yu, T., Chen, J., and Liu, S. Progressive image inpainting with full-resolution residual network. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

Hong, X., Xiong, P., Ji, R., and Fan, H. Deep fusion network for image completion. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

Horita, D., Yang, J., Chen, D., Koyama, Y., and Aizawa, K. A structure-guided diffusion model for large-hole diverse image completion. *ArXiv*, abs/2211.10437, 2022.

Horwitz, E. and Hoshen, Y. Conffusion: Confidence intervals for diffusion models. *ArXiv*, abs/2211.09795, 2022.

Iizuka, S., Simo-Serra, E., and Ishikawa, H. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36:1 – 14, 2017.

Kawar, B., Elad, M., Ermon, S., and Song, J. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022.

Krizhevsky, A. One weird trick for parallelizing convolutional neural networks. *ArXiv*, abs/1404.5997, 2014.

Ku, W.-F., Siu, W. C., Cheng, X., and Chan, H. A. Intelligent painter: Picture composition with resampling diffusion model. *ArXiv*, abs/2210.17106, 2022.

Li, W., Yu, X., Zhou, K., Song, Y., Lin, Z., and Jia, J. Sdm: Spatial diffusion model for large hole image inpainting. *ArXiv*, abs/2212.02963, 2022.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021.

Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B. Image inpainting for irregular holes using partial convolutions. In *European Conference on Computer Vision*, 2018.

Liu, H., Jiang, B., Song, Y., Huang, W., and Yang, C. Correction to: Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. *Computer Vision – ECCV 2020*, 12347:C1 – C1, 2020.

Liu, H., Wang, Y., Wang, M., and Rui, Y. Delving globally into texture and structure for image inpainting. *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2014.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Gool, L. V. Repaint: Inpainting using denoising diffusion probabilistic models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461, 2022.

Nazeri, K., Ng, E., Joseph, T., Qureshi, F. Z., and Ebrahimi, M. Edgeconnect: Structure guided image inpainting using edge prediction. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3265–3274, 2019.

Neal, R. M. Mcmc using hamiltonian dynamics. *arXiv: Computation*, pp. 139–188, 2011.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, 2021.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Peng, J., Liu, D., Xu, S., and Li, H. Generating diverse structure for image inpainting with hierarchical vq-vae. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10770–10779, 2021.

Pokle, A., Geng, Z., and Kolter, Z. Deep equilibrium approaches to diffusion models. *ArXiv*, abs/2210.12867, 2022.

Reddy, V. R., Priya, B. L., Vinuthna, P., Reddy, K. P., and Reddy, D. S. Exploration of image inpainting approaches and challenges: A survey. *International Journal of Computer Engineering in Research Trends*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Saharia, C., Chan, W., Chang, H., Lee, C. A., Ho, J., Salimans, T., Fleet, D. J., and Norouzi, M. Palette: Image-to-image diffusion models. *ACM SIGGRAPH 2022 Conference Proceedings*, 2021a.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021b.

Shah, R., Gautam, A., and Singh, S. K. Overview of image inpainting techniques: A survey. *2022 IEEE Region 10 Symposium (TENSYMP)*, pp. 1–6, 2022.

Sohl-Dickstein, J. N., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2020.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *ArXiv*, abs/1907.05600, 2019a.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *ArXiv*, abs/1907.05600, 2019b.

Song, Y., Yang, C., Shen, Y., Wang, P., Huang, Q., and Kuo, C.-C. J. Spg-net: Segmentation prediction and guidance network for image inpainting. *ArXiv*, abs/1805.03356, 2018.

Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., and Lempitsky, V. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2149–2159, 2022.

Trippe, B. L., Yim, J., Tischer, D. K., Broderick, T., Baker, D., Barzilay, R., and Jaakkola, T. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *ArXiv*, abs/2206.04119, 2022.

Vo, H. V., Duong, N. Q. K., and Pérez, P. Structural inpainting. *Proceedings of the 26th ACM international conference on Multimedia*, 2018.

Wan, Z., Zhang, J., Chen, D., and Liao, J. High-fidelity pluralistic image completion with transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4672–4681, 2021.

Wang, Y., Yu, J., and Zhang, J. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022.

Weng, Y., Ding, S., and Zhou, T. A survey on improved gan based image inpainting. *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp. 319–322, 2022.

Whang, J., Delbracio, M., Talebi, H., Saharia, C., Dimakis, A. G., and Milanfar, P. Deblurring via stochastic refinement. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16272–16282, 2021.

Xiang, H., Zou, Q., Nawaz, M. A., Huang, X., Zhang, F., and Yu, H. Deep learning for image inpainting: A survey. *Pattern Recognit.*, 134:109046, 2022.

Xiao, Q., Li, G., and Chen, Q. Deep inception generative network for cognitive image inpainting. *ArXiv*, abs/1812.01458, 2018.

Yang, L., Zhang, Z., Hong, S., Xu, R., Zhao, Y., Shao, Y., Zhang, W., Yang, M.-H., and Cui, B. Diffusion models: A comprehensive survey of methods and applications. *ArXiv*, abs/2209.00796, 2022.

Yu, Y., Zhan, F., Wu, R., Pan, J., Cui, K., Lu, S., Ma, F., Xie, X., and Miao, C. Diverse image inpainting with bidirectional and autoregressive transformers. *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

Zhao, L., Mo, Q., Lin, S., Wang, Z., Zuo, Z., Chen, H., Xing, W., and Lu, D. Uctgan: Diverse image inpainting based on unsupervised cross-space translation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5740–5749, 2020.

Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E. I.-C., and Xu, Y. Large scale image completion via co-modulated generative adversarial networks. *ArXiv*, abs/2103.10428, 2021.

Zheng, C., Cham, T.-J., and Cai, J. Pluralistic image completion. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1438–1447, 2019.

A. Experiment Setup

A.1. Human Evaluation

As described in Section 5.1, we conduct two human evaluations on Amazon Mturk[2] to evaluate the quality of inpainted images. Figures 7 and 8 show the user interface for the human evaluations, where evaluators are asked to select an image of better quality from two candidate images inpainted by different algorithms according to the criteria listed in instructions. To avoid bias, we put the candidate images in random order. As mentioned in Section 5.1, we perform two user studies, with one of them focusing on `overall` quality and the other focusing on `coherence`.

Detailed criteria of `overall` test are:

- It is important that the edited image should look like a **natural image**. It should not contain a lot of artifacts, distortion or non-commonsensical scenes.
- The completed image should **resemble the source image** except in the missing part.
- The completed missing parts should be **visually coherent** with the given parts in the source image.

Detailed criteria of `coherence` test are:

- The completed missing parts should be visually coherent with the given parts in the source image. More specifically, the completed parts should follow the same style as the given parts in source image, for example, the haircut style of a human should be same and the filled parts should not contain irrelevant objects in the source image.

[2] https://www.mturk.com

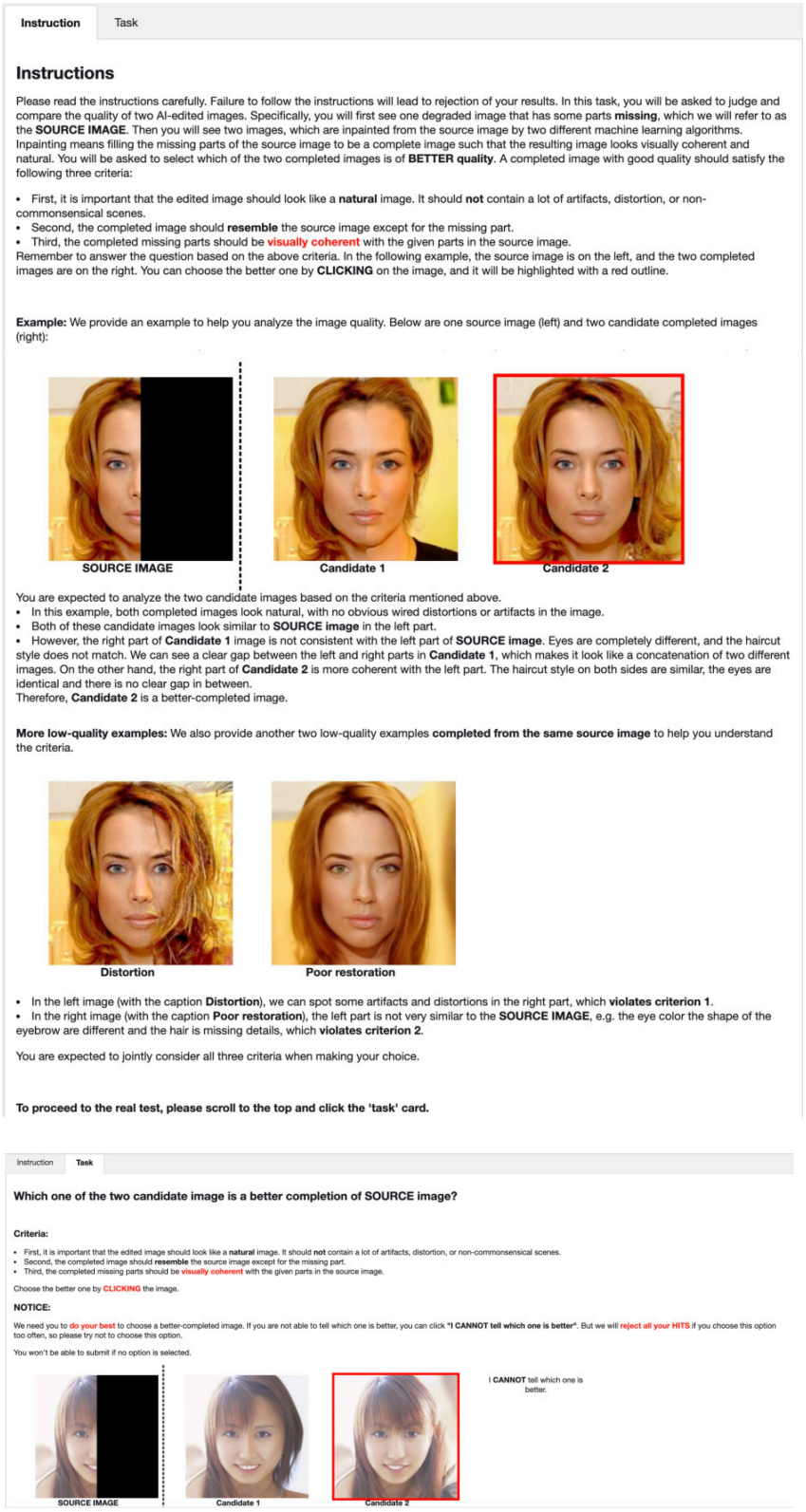

Figure 7. Human evaluation interface for `overall` test. Evaluators are asked to choose an image of better quality from two **Candidate** images following the criteria listed in the instructions.

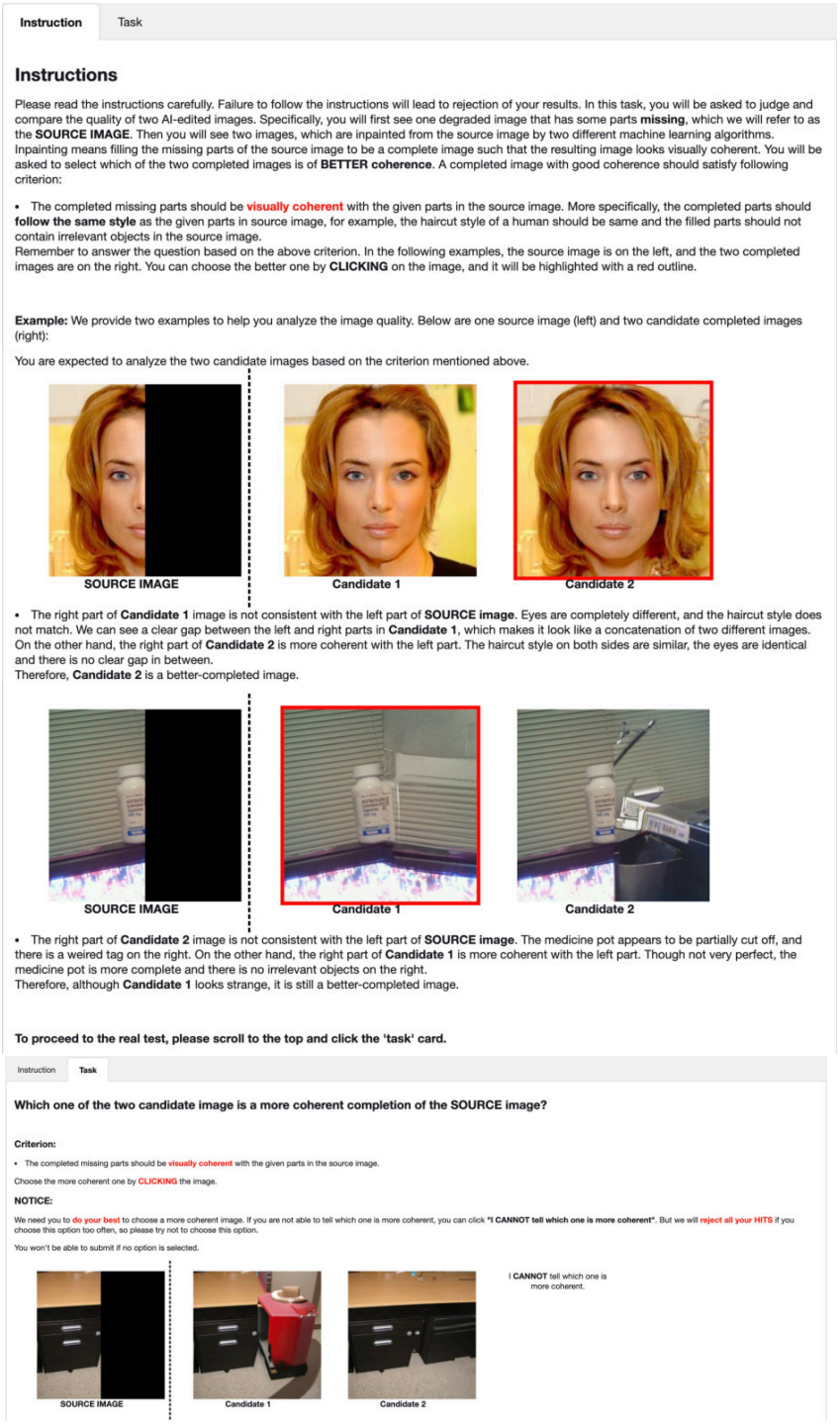

Figure 8. Human evaluation interface for `coherence` test. Evaluators are asked to choose an image of better quality from two **Candidate** images following the criteria listed in the instructions.

A.2. Implementation Details of Baselines

We implement all methods based on the code[3] released by Lugmayr et al. (2022) and generate images with the same pretrained diffusion model. For `CelebA-HQ` dataset, we use the model pretrained by Lugmayr et al. (2022). For `ImageNet`, we use the model pretrained by Dhariwal & Nichol (2021b). For all experiments, we set the number of reverse sampling steps as 250 if not specified otherwise. All experiments are done on an Nvidia-V100-SXM2-32GB GPU. The key hyper-parameters for each baseline method are listed below:

BLENDED, we use DDPM (Song & Ermon, 2019b) sampler with 250 sampling steps.

DDRM, we perform all experiments with the default setting $\eta_B = 1.0, \eta = 0.85$.

RESAMPLING, we generate and resample twenty images[4] in each time step, and select the two with the highest posterior probability when $t = 1$.

REPAINT, we perform all experiments with the default setting, where jump length $j = 10$ and resampling number $n = 10$.

DPS, we perform all experiments following the setting of Gaussian noise measurement in the original paper, where the measurement noise is set to 0 and the step size $\xi_i = 1/\left\|\mathbf{y} - \mathbf{A}(\hat{\mathbf{x}}_\mathbf{0}(\mathbf{x_i}))\right\|$.

DDNM, we perform all experiments with the default setting, where linear degradation operator $\mathbf{A} = r$ and its pseudo-inverse $\mathbf{A}^\dagger = r$.

A.3. Adaptive Learning Rate for Our Method

In our algorithm 1, $\tilde{\boldsymbol{X}}_t$ is optimized to maximize the posterior in each time step. It is equivalent to optimize $\tilde{\boldsymbol{X}}_t$ by minimizing the following loss,

$$\mathcal{L}_t = -\log p'_\theta(\tilde{\boldsymbol{X}}_t|\tilde{\boldsymbol{X}}_{t+1}, \mathcal{C}) = \frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{X}}_t - \tilde{\boldsymbol{\mu}}_t\|_2^2 + \frac{1}{2\xi_t'^2}\left\|\boldsymbol{s}_0 - \boldsymbol{r}\left(\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)\right)\right\|_2^2, \tag{16}$$

and its gradient on $\tilde{\boldsymbol{X}}_t$ could be calculated as follows,

$$\begin{aligned}\nabla_{\tilde{\boldsymbol{X}}_t}\mathcal{L} &= \frac{1}{\sigma_t^2}\left(\tilde{\boldsymbol{X}}_t - \tilde{\boldsymbol{\mu}}_t\right) + \frac{1}{\xi_t'^2}\left[\nabla_{\tilde{\boldsymbol{X}}_t}\boldsymbol{r}\left(\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)\right)\left[\boldsymbol{s}_0 - \boldsymbol{r}\left(\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)\right)\right]\right] \\ &= \frac{1}{\sigma_t^2}\left(\tilde{\boldsymbol{X}}_t - \tilde{\boldsymbol{\mu}}_t\right) + \frac{1}{\xi_t'^2}\frac{\partial \boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)}{\partial \tilde{\boldsymbol{X}}_t}\frac{\partial \boldsymbol{r}\left(\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)\right)}{\partial \boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)}\left[\boldsymbol{s}_0 - \boldsymbol{r}\left(\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)\right)\right].\end{aligned} \tag{17}$$

where we note that $\frac{\partial \boldsymbol{r}\left(\boldsymbol{f}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)\right)}{\partial f_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)}$ is a diagonal matrix with either one or zero. Following the one-step approximation function $f_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)$ in DDIM (Song et al., 2020), we have

$$\begin{aligned}\frac{\partial f_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)}{\partial \tilde{\boldsymbol{X}}_t} &= \frac{\partial\left((\tilde{\boldsymbol{X}}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t))/\sqrt{\bar{\alpha}_t}\right)}{\partial \tilde{\boldsymbol{X}}_t} \\ &= \frac{\mathbf{1} - \sqrt{1-\bar{\alpha}_t}\nabla_{\tilde{\boldsymbol{X}}_t}\boldsymbol{\epsilon}_\theta^{(t)}(\tilde{\boldsymbol{X}}_t)}{\sqrt{\bar{\alpha}_t}}\end{aligned} \tag{18}$$

Given the fact that $\{\bar{\alpha}_t\}$ is strictly decreasing, $1/\sqrt{\bar{\alpha}_t}$ could be very large when t is large and thus lead to large gradient magnitudes for updating $\tilde{\boldsymbol{X}}_t$. In practice, we find that it would easily result in NaN if optimizing $\tilde{\boldsymbol{X}}_t$ directly with the gradient. To alleviate the problem, we multiply the learning rate with an offset term $\sqrt{\bar{\alpha}_t}$. With a base learning rate 0.02, we finally use $0.02\sqrt{\bar{\alpha}_t}$ as our learning rate.

[3] `shorturl.at/AHILU`

[4] It is the maximum affordable number for a 32G GPU.

B. Qualitative Results

We provide the larger size version of Figures 4 and 5 in Figures 9 and 12. More qualitative results are further provided on `CelebA-HQ` in Figure 10, Figure 11 and more qualitative results on `ImageNet` in Figure 13, Figure 14 in this section.

Figure 9. Qualitative results of baseline BLENDED and our methods (COPAINT, COPAINT-TT) on `CelebA-HQ` with seven masks.

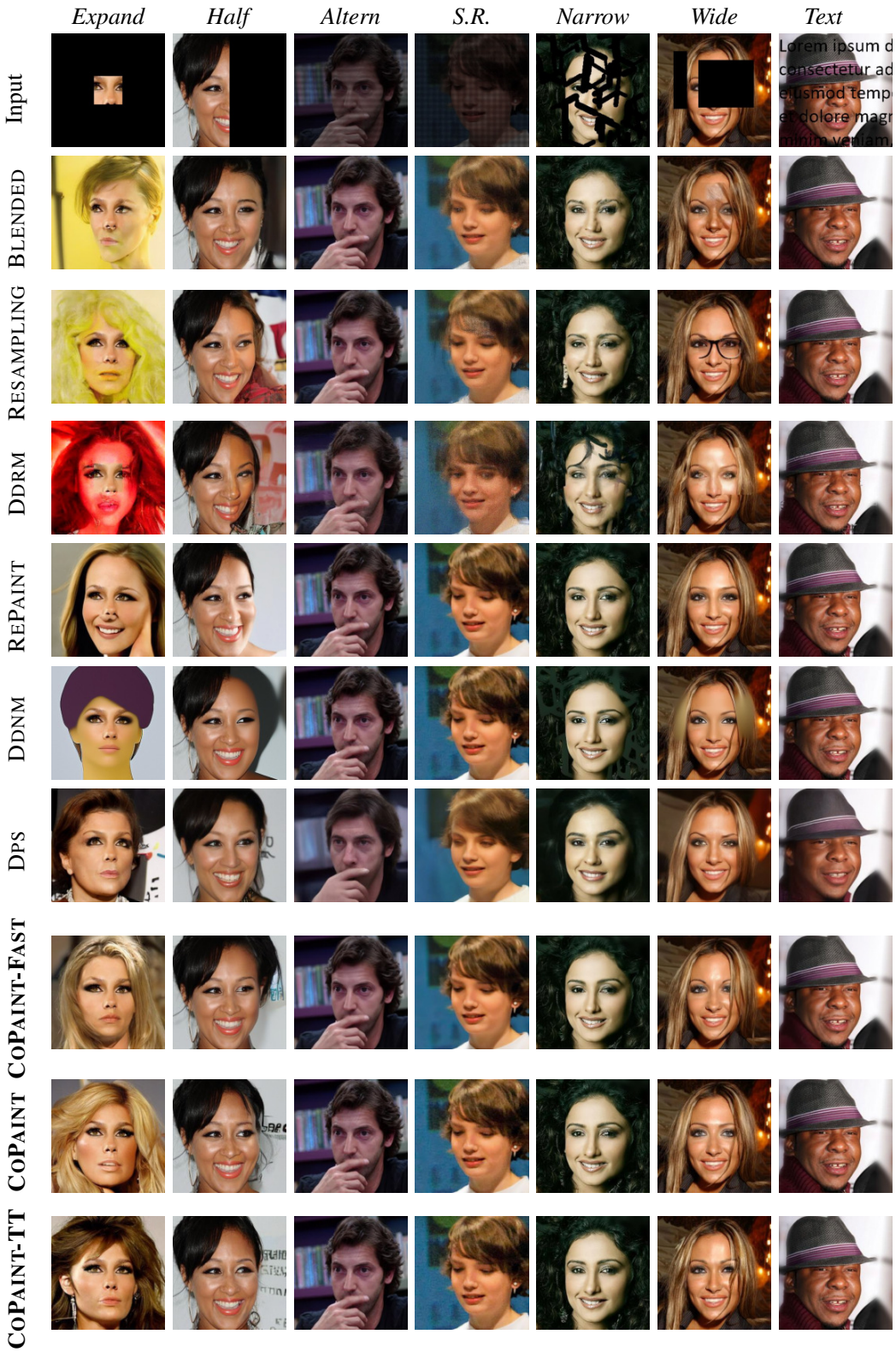

Figure 10. Qualitative results of baseline methods and our methods (COPAINT, COPAINT-TT) on `CelebA-HQ` with seven masks.

Figure 11. Qualitative results of baseline methods and our methods (COPAINT, COPAINT-TT) on `CelebA-HQ` with seven masks.

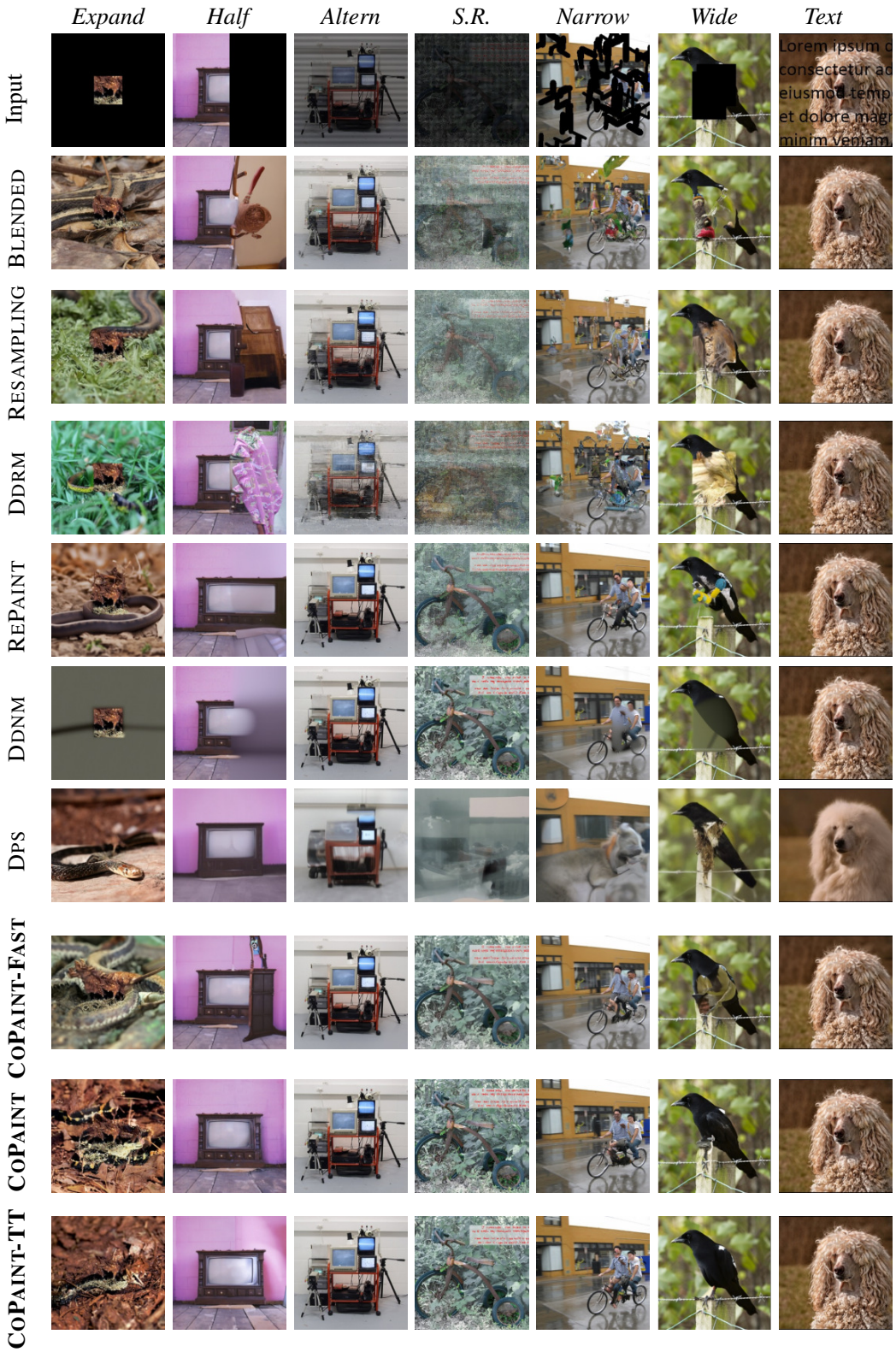

Figure 12. Qualitative results of baseline BLENDED and our methods (COPAINT, COPAINT-TT) on `ImageNet` with seven masks.

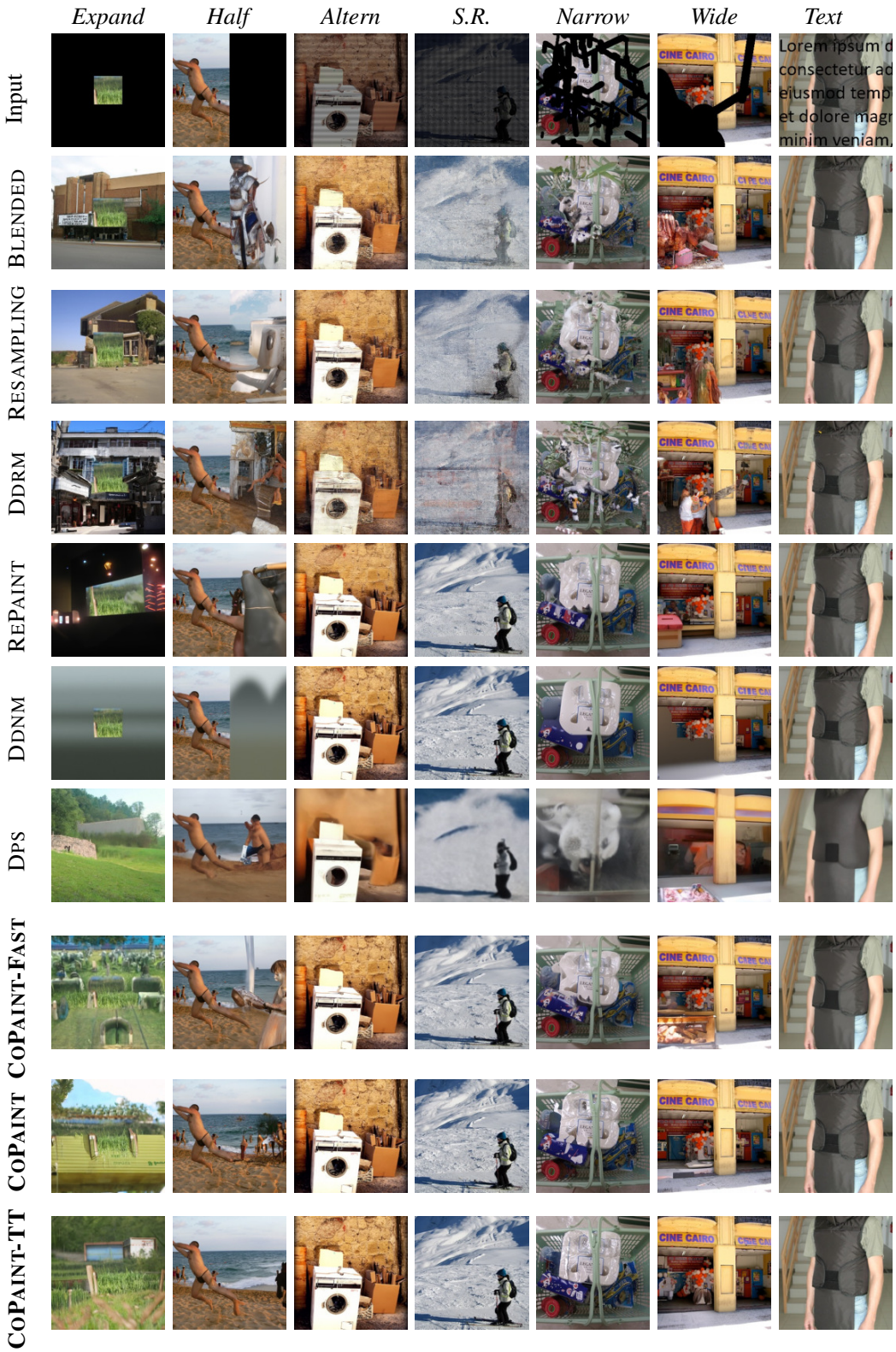

Figure 13. Qualitative results of baseline methods and our methods (COPAINT, COPAINT-TT) on `ImageNet` with seven masks.

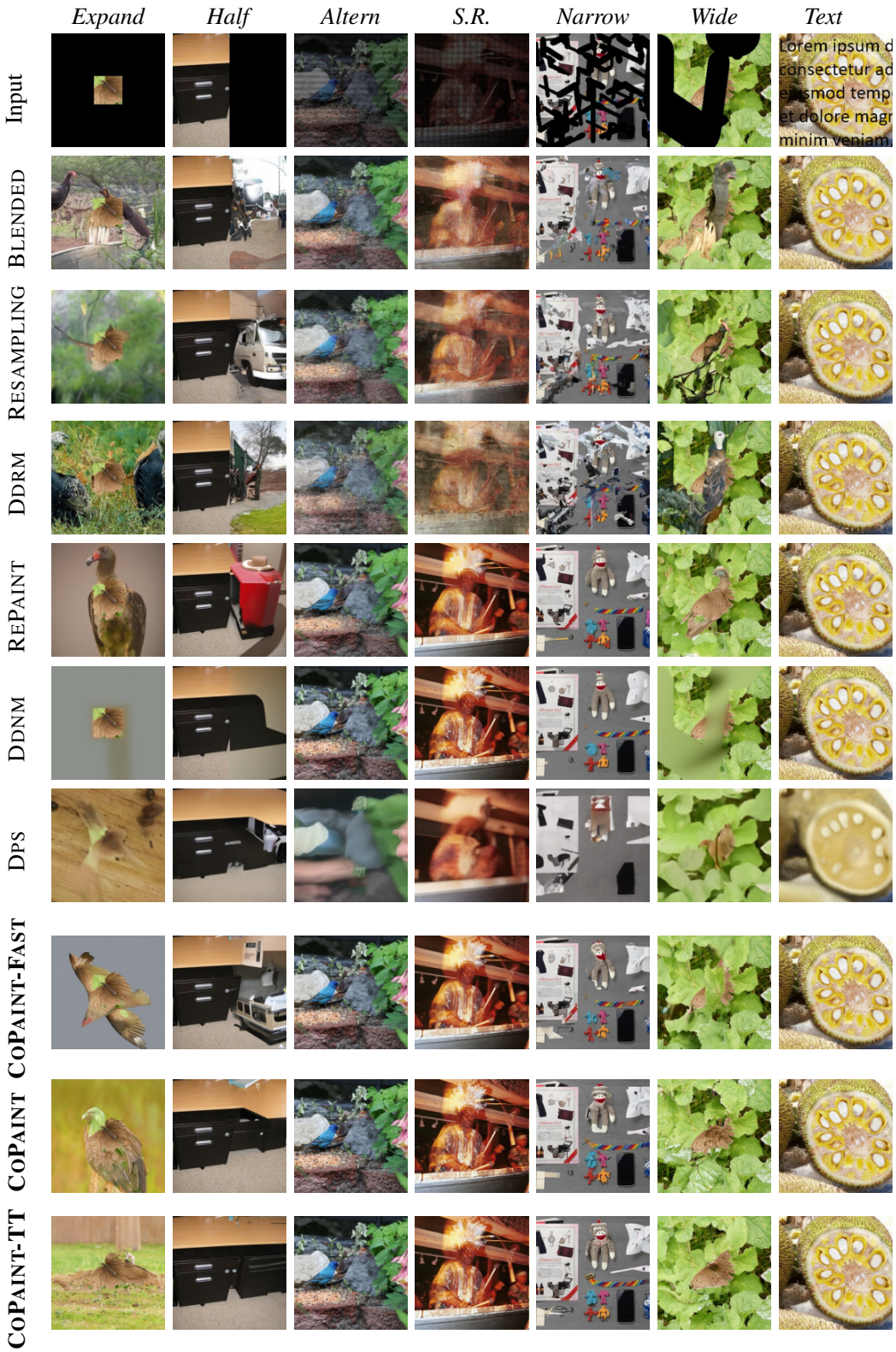

Figure 14. Qualitative results of baseline methods and our methods (COPAINT, COPAINT-TT) on `ImageNet` with seven masks.

C. Additional High-resolution Inpainting Experiments

We conduct an additional experiment on inpainting images, where we use the released 512×512 diffusion model[5] pre-trained on ImageNet dataset as the backbone. The quantitative results could be found in Table 3. As can be observed, our methods still achieve the best *LPIPS* compared with other baselines. For example, CoPaint-TT reduces *LPIPS* by 19.4% compared with the most competing baseline RePaint. In Figure 15 with the time-performance tradeoff, we show that our method is able to outperform other baselines except for RePaint with a comparable computational time budget (CoPaint-Fast), and outperforms all baseline methods given more computational budget (CoPaint and CoPaint-TT).

Table 3. Quantitative results on ImageNet for 512×512 resolution inpainting. Lower is better for LPIPS.

Method	ImageNet-512							
	Expand LPIPS↓	*Half* LPIPS↓	*Altern* LPIPS↓	*S.R.* LPIPS↓	*Narrow* LPIPS↓	*Wide* LPIPS↓	*Text* LPIPS↓	**Average** LPIPS↓
Blended	0.739	0.377	0.210	0.495	0.157	0.179	0.038	0.313
DDRM	0.859	0.391	0.339	0.712	0.204	0.197	0.073	0.396
Resampling	0.799	0.366	0.205	0.482	0.157	0.173	0.039	0.317
RePaint	0.835	0.351	0.066	0.158	**0.083**	0.146	**0.019**	0.237
DPS	0.750	0.575	0.513	0.543	0.496	0.519	0.480	0.554
DDNM	0.850	0.406	0.033	0.079	0.173	0.193	0.044	0.254
CoPaint-Fast	0.678	0.335	0.075	0.128	0.103	0.167	0.043	0.218
CoPaint	0.732	0.310	0.033	0.067	0.100	0.146	0.026	0.202
CoPaint-TT	**0.726**	**0.292**	**0.022**	**0.043**	0.093	**0.136**	0.025	**0.191**

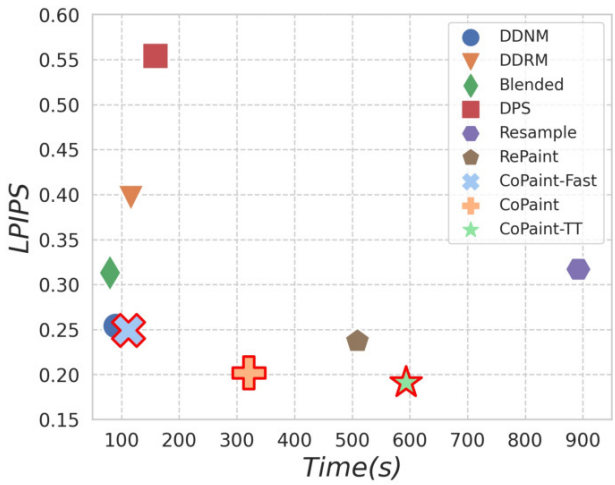

Figure 15. Time-performance trade-off on ImageNet for 512×512 inpainting. The x-axis indicates the average time (↓) to process one image, and the y-axis is the average LPIPS (↓).

[5] https://github.com/openai/guided-diffusion

D. Additional Super-resolution Experiments

We conduct an additional experiment with our method on the super-resolution task. Specifically, we apply average pooling to downsample a 256×256 image to a lower resolution at different scales following DDNM (Wang et al., 2022) and then use different methods to reconstruct the original 256×256 image. We compare our method with DPS (Chung et al., 2022a), DDRM (Kawar et al., 2022), and DDNM (Wang et al., 2022) as they are suitable for the super-resolution task.

The quantitative results in Table 4 demonstrate the consistent superiority of our method compared with other baselines. The qualitative results are shown in Figures 16 and 17. Although the most competing baseline DDNM performs well in $2\times$ and $4\times$ super-resolution, their generated images in $8\times$ super-resolution are more blurry and lack finer details such as hair, as demonstrated in the first `CelebA-HQ` example, and fur, as demonstrated in the second `ImageNet` example. In contrast, our method produces more natural-looking images with better details.

Table 4. Quantitative results of super-resolution task on `CelebA-HQ`(top) and `ImageNet` (*bottom*) datasets. Following (Wang et al., 2022), we apply average-pooling to a 256×256 image to obtain the low-resolution input and then reconstruct the original image using different methods. We perform experiments for three different scales, *i.e.*, $2\times$, $4\times$ and $8\times$, with the image being downsampled at the corresponding scale. We report the objective metric `LPIPS` of each baseline. Lower is better for `LPIPS`.

`ImageNet`					
Scale Factor	DPS `LPIPS`↓	DDRM `LPIPS`↓	DDNM `LPIPS`↓	COPAINT `LPIPS`↓	COPAINT-TT `LPIPS`↓
$2\times$	0.156	0.054	0.031	0.037	**0.025**
$4\times$	0.190	0.228	0.141	0.113	**0.082**
$8\times$	0.235	0.360	0.250	0.293	**0.170**
`CelebA-HQ`					
Scale Factor	DPS `LPIPS`↓	DDRM `LPIPS`↓	DDNM `LPIPS`↓	COPAINT `LPIPS`↓	COPAINT-TT `LPIPS`↓
$2\times$	0.417	0.121	0.113	0.063	**0.042**
$4\times$	0.483	0.345	0.328	0.252	**0.204**
$8\times$	0.531	0.480	0.528	0.511	**0.423**

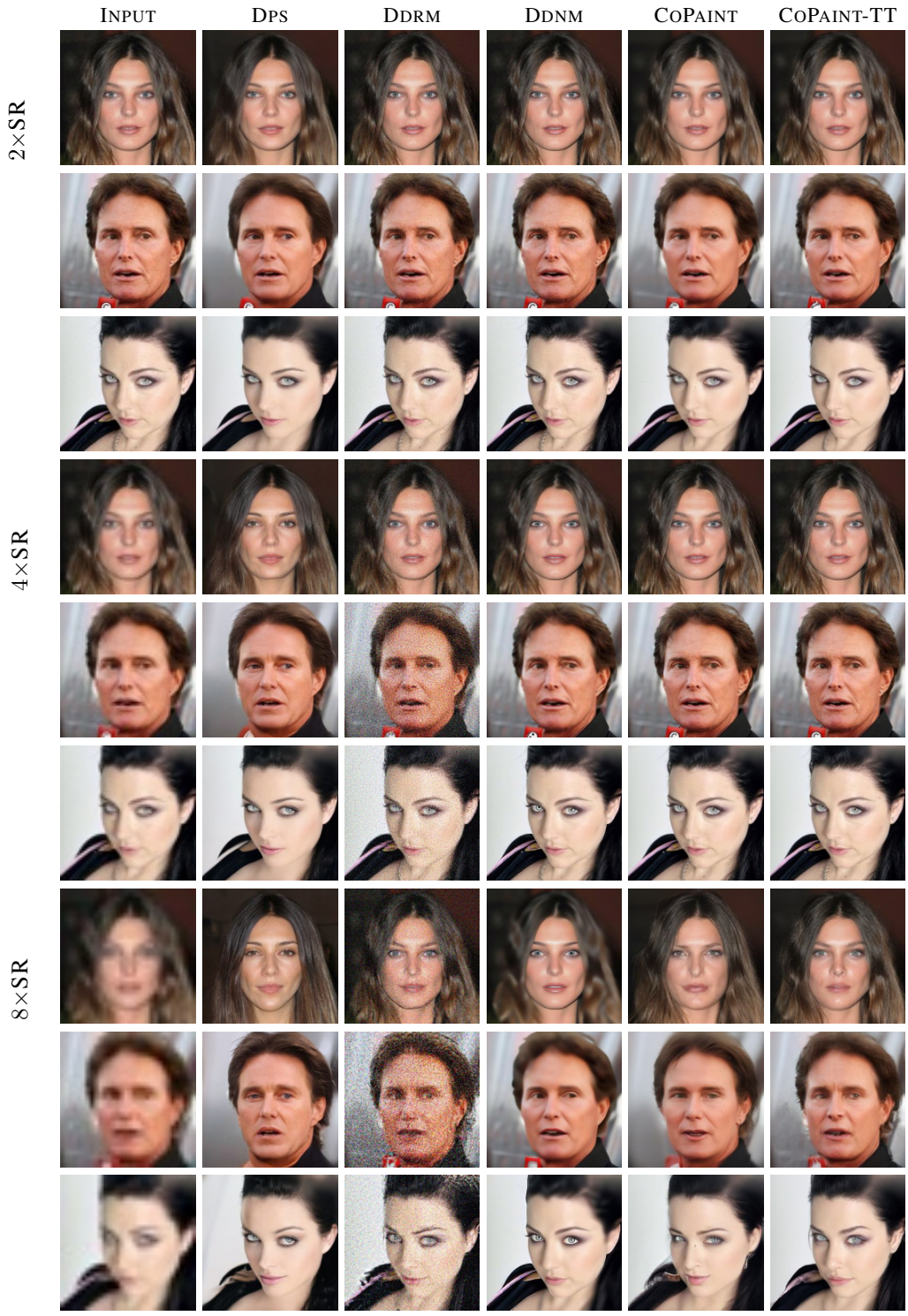

Figure 16. Qualitative results of applying different methods to super-resolution task on `CelebA-HQ` dataset.

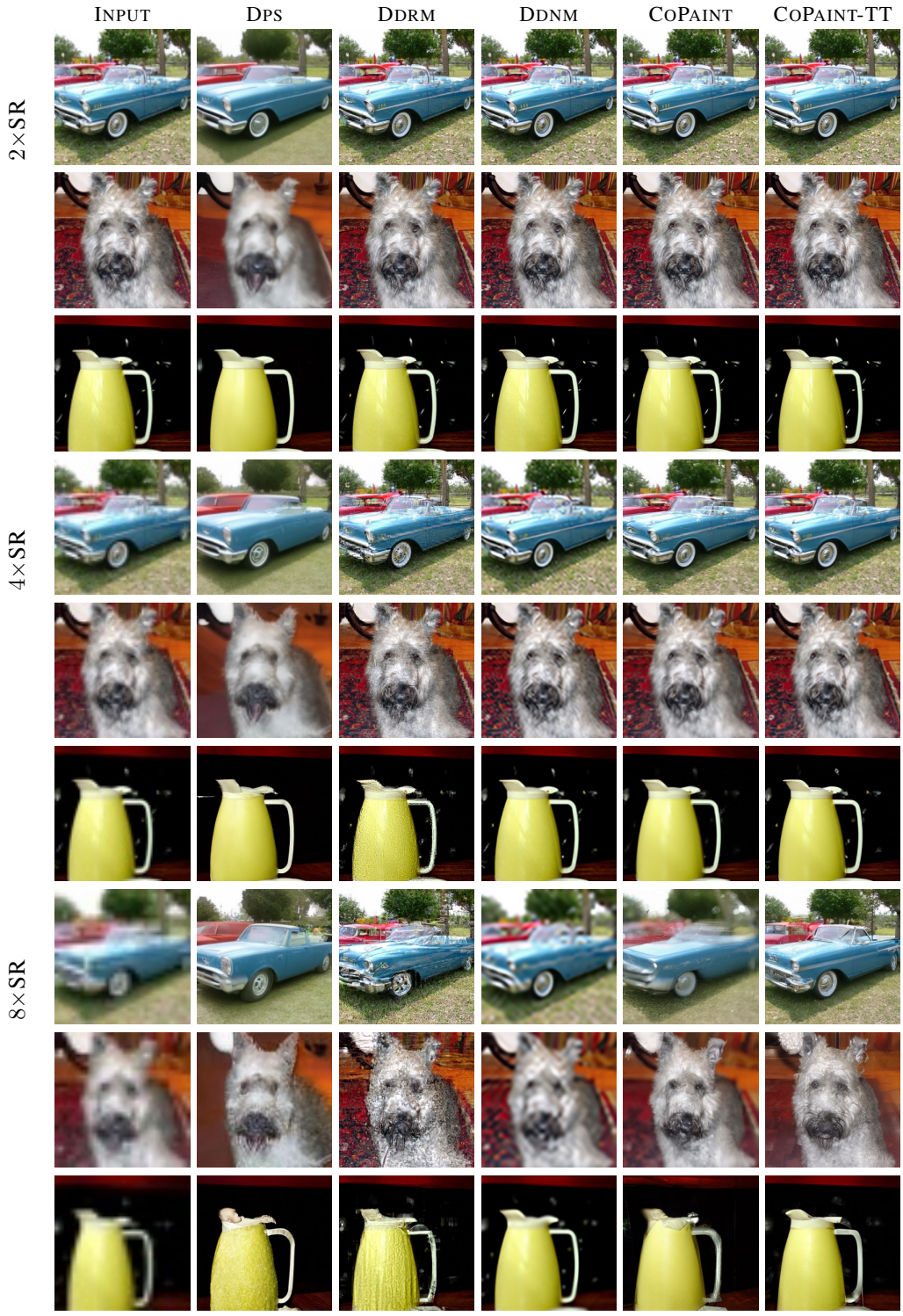

Figure 17. Qualitative results of applying different methods to super-resolution task on `ImageNet` dataset.

E. Failure Case Study

We present a failure case study of our methods, CoPaint and CoPaint-TT, which can be found in Figure 18. Our findings indicate that these methods are susceptible to failure when it comes to inpainting image details. For instance, in the first column, while the inpainted area appears coherent and natural, the text on the hat does not blend well with the surrounding region. Other baselines exhibit similar issues. We attribute this to the deficiency of diffusion models in generating image details, particularly text, and plan to address this in future work. Additionally, we demonstrate that all methods, including ours, are likely to fail for large masked regions where the revealed surrounding information is inadequate for inpainting, resulting in unnatural images. An example of this is shown in the last column.

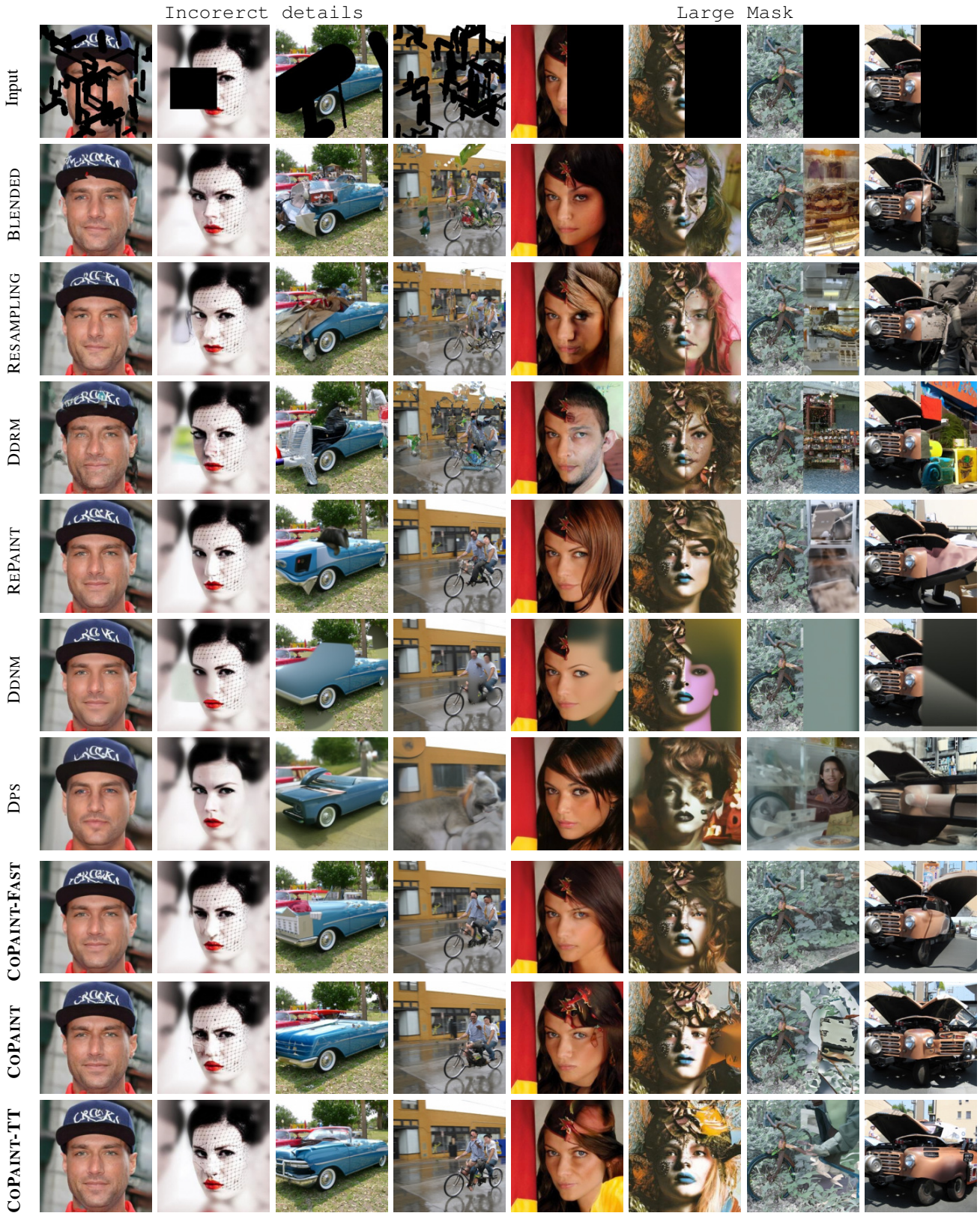

Figure 18. Fail-cases of our method

F. Potential Societal Impacts

Despite the recent success in image generation with diffusion models, these models are prone to the biases exhibited in data (Rombach et al., 2021) and thus could generate biased images for downstream tasks. In line with other diffusion inpainting works, our method heavily relies on the pre-trained diffusion models and thus could exhibit or even amplify the biases existing in the models. For example, as shown in Figure 6, BLENDED (Song & Ermon, 2019a; Avrahami et al., 2021) inpaint a blond-haired woman for the reference image with a black-haired woman, which aligns with a known bias in `CelebA-HQ` dataset (Liu et al., 2021). The underlying reason lies in that, the replacement operation used by BLENDED only enforces the inpainting constraint on the revealed part of the generated image, while the unrevealed part is not directly modified and has to rely more on prior knowledge learned from data. By contrast, in this paper, we introduce a Bayesian framework to jointly modify both the revealed and unrevealed parts of intermediate variables in each time step. This would enforce better coherence between the revealed and unrevealed parts, making our method less susceptible to biases. As shown in Figure 6, our method COPAINT successfully completes the image with a black-haired woman. On the other hand, however, due to the suboptimal greedy optimization and one-step approximation error, we note that there are still some imperfections in our method. Therefore, some bias may still persist, particularly when the revealed part contains too little information. Besides, our method might be used in generating fake content and other malicious images to deceive humans and spread misinformation. In practice, our method should be appropriately used with careful checks on potential risks.