
Nithin Gopalakrishnan Nair\textsuperscript{1}\textsuperscript{*} Anoop Cherian\textsuperscript{2} Suhas Lohit\textsuperscript{2} Ye Wang\textsuperscript{2} 
Toshiaki Koike-Akino\textsuperscript{2} Vishal M. Patel\textsuperscript{1} Tim K. Marks\textsuperscript{2}

\textsuperscript{1}Johns Hopkins University \textsuperscript{2}Mitsubishi Electric Research Laboratories (MERL)

\{ngopala2,vpatel36\}@jhu.edu \{acherian, slohit, yewang, koike, tmarks\}@merl.com

https://merl.com/demos/steered-diffusion

Figure 1. An illustration of various applications of our method. We use a diffusion model trained unconditionally and condition using our proposed algorithm only during the test time. We present the results on six tasks: (a) image inpainting, (b) colorization, (c) image super-resolution, (d) semantic generation, (e) identity replication, and (f) text-based image editing. In part (f), the text prompts for the the first and second columns, respectively, are “This person has blonde hair” and “This person has wavy hair.”

Abstract

Conditional generative models typically demand large annotated training sets to achieve high-quality synthesis. As a result, there has been significant interest in designing models that perform plug-and-play generation, i.e., to use a predefined or pretrained model, which is not explicitly trained on the generative task, to guide the generative process (e.g., using language). However, such guidance is typically useful only towards synthesizing high-level semantics rather than editing fine-grained details as in image-to-image translation tasks. To this end, and capitalizing on the powerful fine-grained generative control offered by the recent diffusion-based generative models, we introduce Steered Diffusion, a generalized framework for photorealistic zero-shot conditional image generation using a diffusion model trained for unconditional generation. The key idea is to steer the image generation of the diffusion model at inference time via designing a loss using a pre-trained inverse model that characterizes the conditional task. This loss modulates the sampling trajectory of the diffusion process. Our framework allows for easy incorporation of multiple conditions during inference. We present experiments using steered diffusion on several tasks including inpainting, colorization, text-guided semantic editing, and image super-resolution. Our results demonstrate clear qualitative and quantitative improvements over state-of-the-art diffusion-based plug-and-play models while adding negligible additional computational cost.

* Work done during internship at MERL.
1. Introduction

Deep diffusion-based probabilistic generative models [8, 14, 40] are quickly emerging as one of the most powerful methods to synthesize high-quality content and have shown the potential to revolutionize content creation not only in computer vision, but also in many other areas including speech, audio, and language. Such models (e.g., ImagGen [36], Stable Diffusion [34]) have demonstrated outstanding synthesis results in conditional generation tasks, such as text-conditioned image synthesis [2, 33] and image reconstruction [31, 35, 38]. However, these models do not typically possess zero-shot conditional generative abilities when used directly (zero-shot capabilities as are commonly seen in language foundation models such as GPT-3 [4]), and often demand large amounts of annotated and paired (multimodal) data for conditional generation, which may be challenging to obtain [15].

One way to circumvent this need for large annotated training sets is to leverage predefined models as plug-and-play modules [12, 24, 25] in an otherwise unconditionally trained diffusion model. Specifically, in such plug-and-play models, a model is first trained in an unconditional setting (without labels). During inference, the plug-and-play modules (networks separately trained for a particular conditional task, e.g., image captioning) are incorporated in the reverse diffusion process to produce intermediate samples guided in the Markov chain in specific directions to satisfy the desired condition. Prior works, such as [25, 26], have proposed similar methods in which the authors derive text- or class-conditioned samples from Generative Adversarial Networks (GANs) [11] that were trained without labels. To achieve this, they iteratively refine the noise input of the GAN until the desired sample satisfies the condition. Very recently, Grakios et al. [12] proposed a diffusion-based plug-and-play method that enables using unconditional diffusion models for conditional generation utilizing class labels. Both these methods are specifically designed for tasks involving label-level semantics. However, these methods do not address the usage of unconditional models for general image-to-image translation tasks, which require synthesizing visual content conditioned on fine-grained details in the source image. There are also works that propose diffusion models for image-to-image translation, such as for image super-resolution and inpainting [5, 20]; however, these methods are task-specific and do not generalize well to new tasks or new types of inverse problems (as demonstrated in Section 5). In this work, we present a generic framework that can generalize to any image-to-image translation task.

In this paper, we derive the necessary theory and formulate an algorithm, which we call Steered Diffusion, for diffusion-based image editing and image-to-image translation; our model is subsequently validated on a wide range of tasks. Steered diffusion is motivated by the energy-based formulation of diffusion probabilistic models [10]. In general, inference in a generative model can be thought of as deriving samples from a learned distribution. Recall that every probability density function can be formulated as an energy field that describes an unnormalized estimate of how the distribution density varies in space [13, 25]. If one needs to find points in space that are the closest match to a given condition, one can utilize gradient-based optimization algorithms to find points in the field that have the highest density value for the condition. The gradient-based optimization scheme can be viewed as a modulation of the energy toward the desired direction. Previous work has utilized this idea on GANs [25, 26] and obtained reasonable results for label-based generation tasks. Due to their model structure, diffusion models are ideal candidates for such an energy modulation. One key challenge remains to design a good energy estimator that is robust to all noise levels. Previously, classifier-based guidance [8, 27] has been proposed and thought of as an energy modulation utilizing a pretrained classifier trained on noisy images. This poses a limitation that the guiding function should be noise-robust. In this work, we propose an alternative solution that does not need noise-robust networks but could use any network by utilizing the diffusion model as an implicit denoiser. Figure 2 gives a brief overview of how our approach is different from existing methods.

We present experiments using steered diffusion on multiple conditional generative tasks on faces as well as generic images as portrayed in Figure 1. We present results on (i) identity replication [7], (ii) semantic image generation [29], (iii) linear inverse problems [21], and (iv) text-conditioned
image editing. Although our method is generic, for evaluations we perform experiments on faces. Before presenting our framework in detail, we now summarize the key contributions of our work:

- We propose steered diffusion, a general plug-and-play framework that can utilize various pre-existing models to steer an unconditional diffusion model.
- We present the first work applicable to both label-level synthesis and image-to-image translation tasks and demonstrate its effectiveness for various applications.
- We propose an implicit conditioning-based sampling strategy that significantly boosts the performance of conditional sampling from unconditional diffusion models compared with previous methods.
- We introduce a new strategy that uses multiple steps of projected gradient descent to improve sample quality.

2. Background

2.1. Related Work

Early works on unpaired image-to-image translation utilize a cycle consistency loss between the input and the target domains [9,46]. Newer works, such as [16], have introduced a contrastive learning-based approach where a contrastive loss between corresponding patches of the input and target domains is minimized. The consistency-based method often fails to generate photorealistic images; hence, conditional generative models are preferred when labeled data are available. A few works [37,38] utilize diffusion models for conditional image-to-image translation because of their photorealistic generation quality.

Guiding diffusion models during inference time has been explored by several works, such as [24]. The first method that proposed inference-time conditioning [8] uses a pretrained noise-robust classifier to guide the inference of an unconditional model. GLIDE [27] proposed a method for conditioning using text. Earlier work in plug-and-play modelling for generative models utilized GANs and performed iterative refinement on the latent space of GANs [25]. This method uses a predefined classifier or text captioning network to estimate a loss between the desired label output or text caption and the one generated from the GAN generator. This loss is backpropagated to refine the noise input of the GAN iteratively until the generator predicts the desired output. Recently, [12] proposed a method that uses diffusion models as a plug-and-play prior for class-conditioned generation. Several works have addressed the task of image-to-image translation using unconditional diffusion models [1,5,18,20], but each of these proposes a task-specific inference scheme. For example, ILVR [5] performs image super-resolution, and Reinpaint [20] performs image inpainting. Blended diffusion [1] proposes a method for text-conditioned image editing. DDRM [18] proposes an inference-time scheme offering a general solution for linear inverse problems such as colorization and super-resolution.

2.2. Concurrent Work

In concurrent work to ours that also explores zero-shot conditional generation using diffusion models, [3] used a text to image model [34] and a two-step forward and backward universal guidance process, but it works well only after heavy optimization on network-based inverse problems such as semantic generation and identity generation. Another concurrent work [42] explored the usage of unconditional diffusion models for linear inverse problems using a pseudo-inverse model. In contrast to all these prior works, our steered diffusion algorithm generalizes well to both image-to-image translation tasks and high-level label-based generation tasks.

2.3. Denoising Diffusion Probabilistic Models

Denoising diffusion probabilistic models (DDPMs) [14,40] belong to a class of generative models in which the model learns the distribution of data through a Markovian sampling process. DDPMs consist of a forward process and a reverse process. Let \(x_t\) denote the latent state of an input image at timestep \(t\) in a diffusion process. The sampling operation \(q(\cdot)\) for the forward process in DDPM is defined as:

\[
q(x_{t-1}|x_t) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I),
\]

where \(\{\beta_t\}\) is a predefined variance schedule and \(I\) is the identity matrix.

The forward process can be considered as a noising operation, where the next state \(x_t\) is obtained from the current state \(x_{t-1}\) by adding a small amount of Gaussian noise according to the sampled timestep. The state \(x_t\) at timestep \(t\) can also be sampled directly from the initial state \(x_0\), using:

\[
q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I),
\]

or equivalently,

\[
x_t = x_0\sqrt{\alpha_t} + \epsilon\sqrt{1-\alpha_t}, \quad \epsilon \sim \mathcal{N}(0, I),
\]

where \(\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s\) and \(\alpha_t = 1 - \beta_t\).

In [40], it is shown that if the number of time steps is large and the increment in \(\{\beta_t\}\) is small, then each step in the reverse sampling process can also be approximated by a Gaussian. If \(\mu_\theta\) and \(\Sigma_\theta\) respectively denote the mean and the covariance of this Gaussian, modelled via neural networks with parameters \(\theta\), then each reverse step samples the state \(x_{t-1}\) according to:

\[
p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)).
\]

The parameters \(\theta\) are obtained by minimizing the variational lower bound of the negative log-likelihood of the data distribution.
The term $p_0$ is the density of probability density in the EBM formulations (5) of the individual densities and can be represented as

$$
\nabla_x \log p_0(x_{t-1}|x_t, c) = \nabla_x \log p_0(x_{t-1}|x_t) - \nabla_x V_1(x_{t-1}, c) + \nabla_x V_2(x_{t-1}, c),
$$

(8)

where $V_1$ and $V_2$ are the corresponding energy functions that model the conditional distributions of $x_t$ and $x_{t-1}$ given a condition $c$. Specifically, they project the higher dimensional $x_t$ to the lower dimensional space of $c$ and measure the distance between the mapped value and $c$. The better this particular measure, the more effectively it can be used to generate conditional samples from an unconditional model. Using Eq. (8), the the conditional sampling equation for the reverse process is

$$
x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon(t) (x_t, t) \right) - \nabla_x V_1(x_{t-1}, c) + \nabla_x V_2(x_{t-1}, c) + \sigma_t \epsilon.
$$

(9)

Here $\epsilon(t)$ is the network prediction at a timestep $t$, and $\sigma_t$ is the corresponding variance of the reverse step. The formulation in Eq. (9) shows that the energy function requires a functional mapping from a noisy $x_t$ to $c$.

In many applications of interest, the mapping function from $x_t$ to $c$ is complex and can be modeled effectively using deep networks. For example, in image-to-image translation tasks such as image generation from semantic maps [29], the image is $x_t$ and the semantic map is the condition $c$. Similarly, for text to image generation, the image is $x_t$ and $c$ corresponds to the text. Ideally, we would like to employ an existing (pre-trained) deep neural network for the mapping from $x_t$ to $c$. However, deep networks are usually trained on clean images, which limits the usability of existing pre-trained deep networks for mapping directly from the noisy image $x_t$ to $c$. One workaround would be to use noise-robust networks to map from $x_t$ to $c$, but training noise-robust networks for conditional mapping can be computationally
where we define the implicit step prediction $x_{0|t}$ as:

$$x_{0|t} = \frac{x_t - \sqrt{1 - \alpha_t \epsilon_\theta(t)}(x_t)}{\sqrt{\alpha_t}}.$$  

(11)

Here, we assume $x_t$ and $x_{t-1}$ are first denoised to $x_{0|t}$ and $x_{0|t-1}$, respectively. The terms $\delta_1$ and $\delta_2$ capture the errors arising from the shift in the domain from $x_t$ to $x_0$ and from $x_{t-1}$ to $x_0$; for large $t$, $\delta_1 \approx \delta_2$ as the implicit predictions at nearby steps tend to be similar.

As shown in our experiments, the energy function should be selected according to the task. An easy way to choose the energy function is by looking at the training loss of the mapping network. For example, in the case of semantic generation, a good energy function is the cross-entropy loss between the predicted semantic map at any timestep and the input semantic map. In the case of identity replication, a good choice of regularization would be the negative cosine similarity score between the embeddings from a recognition network for the input and target image. In the case of text-to-image generation, it would be CLIP loss [32]. As a rule of thumb, an energy function could be chosen easily by looking at the loss function used to train the pre-trained network (or an inverse function) that maps from the image $x$ to the condition $c$.

### 3.2. Revisiting Sampling in Diffusion Models

To obtain a closed-form expression for plugging the energy-based formulation into the reverse sampling process efficiently, we take inspiration from DDIM [41] and revisit the reverse sampling operation of diffusion models. From $p_u(x_{1:T})$, one can generate a sample $x_{t-1}$ from a sample $x_t$ by:

$$\nabla_x \log p_\theta(x_{t-1}|x_t, c) = \nabla_x \log p_\theta(x_{t-1}|x_t) - \nabla_x V_1(x_{0|t}, c) - \delta_1 + \nabla_x V_2(x_{0|t-1}, c) + \delta_2,$$

(10)

where, we define the implicit step prediction $x_{0|t}$ as:

$$x_{0|t} = x_t - \sqrt{1 - \alpha_t \epsilon_\theta(t)}(x_t) \frac{x_t - \sqrt{1 - \alpha_t \epsilon_\theta(t)}(x_t)}{\sqrt{\alpha_t}}.$$  

(11)

Hence, we modify our original energy expression (8) to:

$$V(x_t) = \nabla_x \log p_\theta(x_t) = \nabla_x \log p_\theta(x_{t-1}|x_t) - \nabla_x V_1(x_{0|t}, c) - \delta_1 + \nabla_x V_2(x_{0|t-1}, c) + \delta_2,$$

(10)

Algorithm 1 Steered Diffusion

**Input:** Energy function $V$, condition $c$

1. $x_T \sim \mathcal{N}(x_T; 0, I)$
2. for $t = T-1, \ldots, 1$ do
3.     for $n = N_c, \ldots, 1$ do
4.         $\epsilon \sim \mathcal{N}(\epsilon; 0, I)$
5.         $x_{0|t} = \frac{x_t - \sqrt{1 - \alpha_t \epsilon_\theta(t)}(x_t)}{\sqrt{\alpha_t}}$
6.         Compute $x_{0|t}^{uc}$ with $V$, $\epsilon$ using Eq. (15)
7.         if $(n > 1)$ then
8.             Compute $x_{t|c}^{uc}$ using Eq. (13)
9.         else
10.        Compute $x_{t|c}^{uc}$ using Eq. (13)
11. end if
12. end for
13. end for
14. return $x_0$

as in Song et al. [41]. Using Eq. (11), we can rewrite the unconditional sampling step Eq. (12) in terms of $x_{0|t}$:

$$x_{t|c}^{uc} = \sqrt{\alpha_{t-1}} x_{0|t} + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(t) + \sigma_t \epsilon,$$

(13)

Here the superscript $uc$ denotes the unconditional sample, which is obtained without any steering while transitioning from $x_t$ to $x_{t-1}$. The conditional sampling step (8) can then be rewritten as

$$x_{t-1} = x_{t|c}^{uc} - \nabla_x V_1(x_{0|t}, c) + \nabla_x V_2(x_{0|t-1}, c).$$  

(14)

Through this, we can modulate $x_0$ directly, as $x_{0|t}$ is also a function of $\epsilon_\theta(t)$. Following Eq. (14), a rough estimate of the desired $x_0$ for conditional sampling, represented by $x_{0|t}^{uc}$, can be obtained using

$$x_{0|t}^{uc} = x_{0|t} - k(t) \nabla_x \left( V_1(x_{0|t}, c) - V_2(x_{0|t-1}, c) \right),$$

(15)

where $k(t)$ is a scaling factor defining the strength of regularization. We call the process of finding $x_{0|t}^{uc}$ from $x_{0|t}$ Implicit Steering Control (ISC), and we call the new sampling process steered diffusion. The exact algorithm is illustrated in Fig. 3 and explained in Algorithm 1.
prediction step becomes
\[
x_{0|t}^{\text{feas}} = x_{0|t} + k(t)(D(y) - D(x_{0|t})), \quad \text{where } c = D(y).
\]  
(16)

Here, \( y \) is the clean image and \( D \) is the known degradation model. Our sampling procedure ensures that the series of operations preserve the consistency of domains of \( x_{t-1} \) and \( x_{0|t}^{\text{feas}} \) to the original data distribution at the corresponding timesteps. An illustration is shown in Figure 4.

4.2. Multi-Step Implicit Modulation

Our experiments show (see Fig. 7) that performing refinement using Eq. (15) on the implicit step prediction multiple times for each timestep significantly boosts the conditioning quality for more ill-posed conditions such as image inpainting and colorization. A similar observation was also found by [20]. Specifically, at a particular timestep \( t \), we iterate the procedure of steering towards the next sampling step \( x_{t-1} \) and then adding noise to return to \( x_t \). We present the corresponding algorithm in Algorithm 1. An example is shown in Fig. 7, where more realistic images are generated using the multiple-step sampling scheme row labeled “OURS multi.”

Effectively, the \( V_2 \) term in Eq. (12) can be thought of as enabling a multistep sampling in which we modulate the current step by looking ahead to the next sampling step. On a careful analysis of Eq. (7), i.e. from the score contribution due to the different regularization functions, we can see that the term for \( \nabla_{x_t} V_1(x_t, c) \) modulates \( x_t \) based on its current state, and the term from \( \nabla_{x_t} V_2(x_{t-1}, c) \) is a look-ahead correction where the derivative with respect to the future prediction is found. This is exactly what happens in the case of looping back from \( x_{t-1} \) to \( x_t \), where \( x_t \) is modulated iteratively by looking forward to what the future prediction would be.

4.3. Choosing the Scaling Factor \( k(t) \)

In Eq. (15), \( k(t) \) denotes the strength of the regularization constraint. A very small \( k(t) \) would denote no effective regularization, and a large \( k \) would lead to the diffusion process going out of the latent space manifold. Since the derivative of the regularization function by itself is a score value, similar to the normal scaling value of the score function, the appropriate time-varying normalization factor is \( \sqrt{1 - \bar{\alpha}_t} \).

The exact value for \( k(t) \) for each task is defined in Table 1. For linear inverse problems we use a constant \( k(t) = 1 \), which provided the best results.

5. Experiments

We evaluate the performance of our network qualitatively and quantitatively using four image-to-image translation tasks—semantic layout to face image translation, face inpainting, face colorization, and face super-resolution—as well as two high-level vision tasks: identity-based image generation and text-guided image editing. Unlike existing approaches, our method is completely zero-shot and applies to a wide variety of tasks. We compare with other diffusion-based approaches best applicable for each task for a fair evaluation. We also compare the semantic layout to image translation performance with that of task-specific unsupervised methods. We choose the unconditional model released by [5] as the unconditional pretrained diffusion model in all of our experiments with faces. Note that the sampling scheme in ILVR [5] and Repaint [20] can be thought of as happening at time \( t \) rather than at the implicit step as in our method. Hence, comparing these methods for super-resolution and inpainting in our experiments below can be considered an additional ablation study highlighting the improvement from our implicit sampling.

5.1. Implementation Details

For our experiments, we utilize pixel-level unconditional diffusion models. For faces, we utilize the model trained on the FFHQ dataset [17] that was released by [5]. For generic images, we use the model trained on ImageNet [6] released in ADM [8]. All our experiments use 100 steps of sampling.
Table 1. Parameter set for each application.

![Figure 5. Qualitative comparisons for semantic generation.](image)

![Figure 6. Results on 8x super resolution.](image)

### 5.2. Semantic Face Generation

To evaluate how our method performs in generic image-to-image translation tasks, we evaluate our method’s performance for the task of semantic layout to face generation. We utilize the CelebA dataset for this. To generate the semantic labels, we use facer [45] and create 11 label classes for each face. Since there is no other unconditional model that can perform fully test-time semantic generation, to evaluate the performance of our method, we compare with fully unsupervised image translation methods: CycleGAN [46], CUT [28], and ILVR [5].

The corresponding qualitative results are shown in Fig. 5. It is clear that CUT, CycleGAN, and ILVR produce unrealistic facial images with huge artifacts or create low-resolution faces. In contrast, our method always creates good-quality realistic faces. We present the quantitative results in Table 2. The table shows that our method obtains the best FID scores of all methods and obtains the best mIOU score among the inference-time techniques.

### 5.3. Face Super-Resolution

We evaluate the performance for the face super-resolution task using the CelebA dataset [19]. As baselines, we utilize fully inference-time methods in which no task-specific training is used. As the first baseline, we choose PULSE [22], a self-supervised upsampling technique utilizing GANs. As the next comparison method, we choose ILVR [5] which, like our method, performs super-resolution utilizing an unconditional pre-trained diffusion model. However, in ILVR, sampling happens at timestep \( t \) rather than at the implicit step in our algorithm. In total, we utilize 300 images for evaluations. We present some qualitative results in Fig. 6. For ILVR [5], we utilize 100 timesteps of sampling, the same as in our case. As we can see, PULSE [22] and ILVR [5] are unable to restore the correct identity and also contain blur artifacts after restoration. On the other hand, steered diffusion (our method) is able to restore photorealistic facial images.
images. The qualitative evaluations are presented in Table 3; our method yields a 0.18 improvement in perceptual similarity, 6.95 dB improvement in PSNR, and 0.24 improvement in SSIM versus all of the other comparison methods.

5.4. Face Colorization

As a baseline method, we modify ILVR [5] to suit the task of colorization. For this, rather than performing the constraint at every step, we start the sampling process from a noised grayscale image and enforce consistency between the generated and original grayscale images. In total, we utilize 300 images for evaluation. The corresponding results can be seen in Fig. 7. As we can see, our method is able to reconstruct photorealistic faces with naturalistic colours compared to ILVR [5]. The corresponding quantitative metrics are presented in Table 4. We get a significant boost in performance, with an FID score of 19, LPIPS [44] score of 0.19, and NIQE [23] score of 1.5 for our method.

5.5. Inpainting, Image Editing, Identity replication

Our method can also utilize multiple conditions simultaneously; we provide an illustration in Figure 9 where we condition with an identity-preserving network and a text caption simultaneously. From the figure, one can see that our method can generalize well to a diverse range of captions. For preserving identity, we used the VGGFace network [30]. We utilize FARL [45], which is pre-trained with face and corresponding text pairs to enforce the captions. For generic identity replication as in Fig. 1 we use FARL face embedder.

For our image inpainting experiments, we use the subset released by [43] and evaluate three different kinds of masks. Our method obtains better results than existing baselines across all mask variations. Qualitative and quantitative results are shown in Fig. 8 and Table 5, respectively.
Table 5. Quantitative results for inpainting.

<table>
<thead>
<tr>
<th>Method</th>
<th>Thin</th>
<th>Medium</th>
<th>Thick</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID ↓</td>
<td>LPIPS ↓</td>
<td>PSNR ↑</td>
</tr>
<tr>
<td>Degraded</td>
<td>371.03</td>
<td>0.676</td>
<td>7.847</td>
</tr>
<tr>
<td>Reinpaint [20]</td>
<td>43.35</td>
<td>0.304</td>
<td>17.99</td>
</tr>
<tr>
<td>Ours</td>
<td>30.85</td>
<td>0.183</td>
<td>24.92</td>
</tr>
<tr>
<td></td>
<td>FID ↓</td>
<td>LPIPS ↓</td>
<td>PSNR ↑</td>
</tr>
<tr>
<td>Degraded</td>
<td>258.86</td>
<td>0.624</td>
<td>7.527</td>
</tr>
<tr>
<td>Reinpaint [20]</td>
<td>53.71</td>
<td>0.399</td>
<td>13.71</td>
</tr>
<tr>
<td>Ours</td>
<td>35.39</td>
<td>0.220</td>
<td>21.55</td>
</tr>
<tr>
<td></td>
<td>FID ↓</td>
<td>LPIPS ↓</td>
<td>PSNR ↑</td>
</tr>
<tr>
<td>Degraded</td>
<td>231.93</td>
<td>0.585</td>
<td>7.606</td>
</tr>
<tr>
<td>Reinpaint [20]</td>
<td>52.40</td>
<td>0.407</td>
<td>12.78</td>
</tr>
<tr>
<td>Ours</td>
<td>40.12</td>
<td>0.242</td>
<td>18.87</td>
</tr>
</tbody>
</table>

Table 5. Quantitative results for inpainting.

5.6. Ablation Study

Effect of scaling factor $k(t)$ for semantic generation: In this section, we analyze how the scaling factor affects the quality of the sample in the case of complex conditioning of semantic generation. Fig. 10 shows the variation of sample quality starting from the same initial noise with different scaling factors $k(t) = K \sqrt{1 - \bar{\alpha}_t}$. The sample quality is bad for very low scaling factors, and for very high scaling factors, the diffusion process escapes the manifold of natural face images. We show the variation in sample quality for a fixed scaling factor versus a time-varying scaling factor in Fig. 12, which demonstrates that a time-varying scale factor produces more realistic samples. This is because the effective variance of the noise scheduling, which effectively controls the amount of regularization possible at a particular timestep, reduces as the generation process proceeds. Hence a larger tweak is permissible at the early steps of diffusion, and only very small tweaks are permitted in the later steps.

Noise-Robust classifier: To validate the claims in section 3.1, we train a noise-robust inverse mapper for the task of colorization and show the output of the noise-robust classifier and the diffusion outputs for different noise levels in Fig. 11. The noise-robust classifier fails to preserve key details that our approach preserves.

Limitations Although our method can generalize to a wide series of tasks, one limitation that persists is the value of the scaling factor $k(t)$. The value of $k(t)$ has to be empirically found based on the task, but once a few images are used to tune the value of $k(t)$, the model generalizes well to other conditioning images of the same task. Like any other conditional generation models that can perform image editing, our method also has societal impacts, and care must be taken in applying these methods.

6. Conclusion

In this paper, we propose the first framework for plug-and-play conditional generation that can generalize well to both image-to-image translation tasks and label-based generation tasks. For this, we use the energy-based formulation of diffusion models and modulate the inference process using a task-specific predefined network or other preexisting function. Furthermore, we introduce a novel implicit sampling-based technique that improves the sampling quality across multiple tasks. We performed experiments on various tasks to show that our method can generalize across multiple tasks and outperforms existing methods that do not require additional training.
References


