Final report of Qualcomm Intern Project 2023

CameraML

Inpainting and Outpainting with Diffusion Models

Xitong Zhang (CMSE, Michigan State University)

Mentors: Shusil Dangi, Sandesh Ghimire

Manager: Ravi Dayana

Outline

- Motivation
- Diffusion Models for unconditional image generation
- Unsupervised Inpainting and Outpainting:
  - Denoising Diffusion Probabilistic Models (DDPM) with optional constraints
  - Diffusion GAN and Wavelet Diffusion Models
- Supervised Inpainting and Outpainting
- Free-Size Inpainting and Outpainting
  - Denoising Diffusion Null-Space Models (DDNM) with Mask-Shift Restoration and Hierarchical Restoration
  - Future work: Diffusion models + GAN

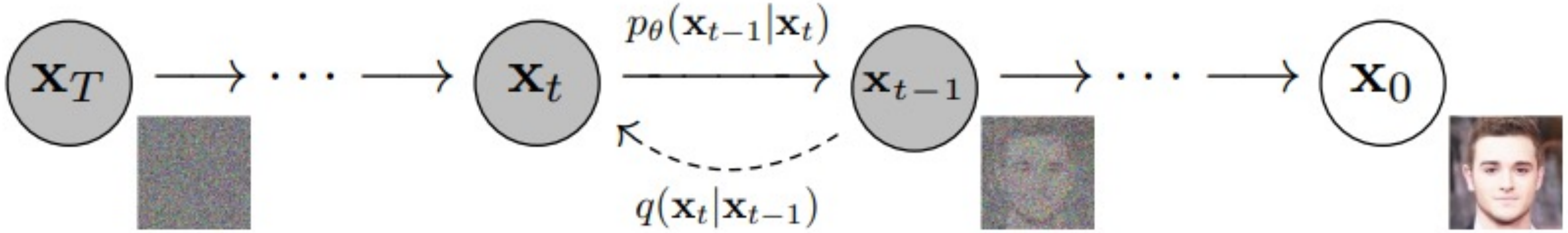
Motivation

- Inpainting and Outpainting are essential to computational photography.
- The goal is to fill the unknown target region of an image.

Diffusion Models

[2006.11239] Denoising Diffusion Probabilistic Models (arxiv.org)

By learning the reverse process of degrading clean image, diffusion models can generate high-quality and diverse images from random noise.

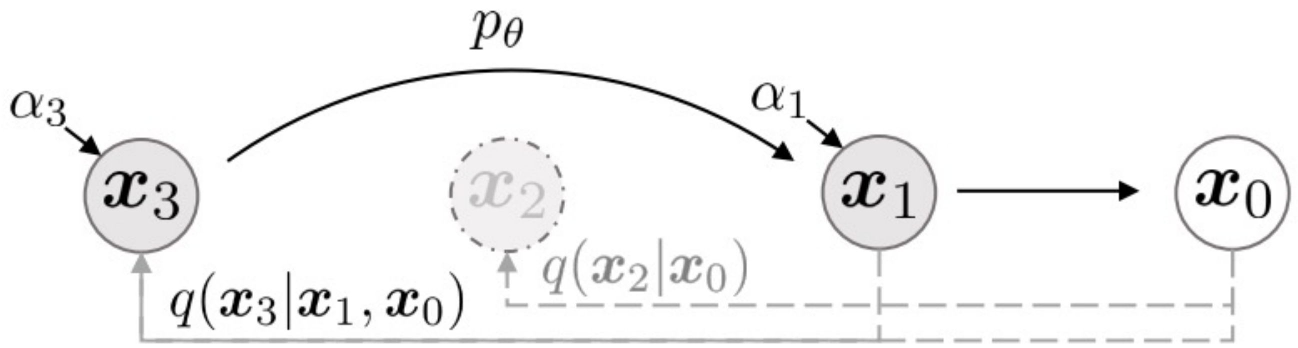


Denoising Diffusion Probabilistic Model (DDPM)

Denoising Diffusion Implicit Models (DDIM)

2010.02502.pdf (arxiv.org)

- A well-trained DDPM with T steps can generalize to sampling with τ steps (a subsequence of T steps), since we only consider the mapping from x_T to x_0.
- Thus, we can skip-sampling by rescaling the denoising strength.
- It also provides a deterministic sampling approach.



Evaluation Metrics for Unconditional Generation

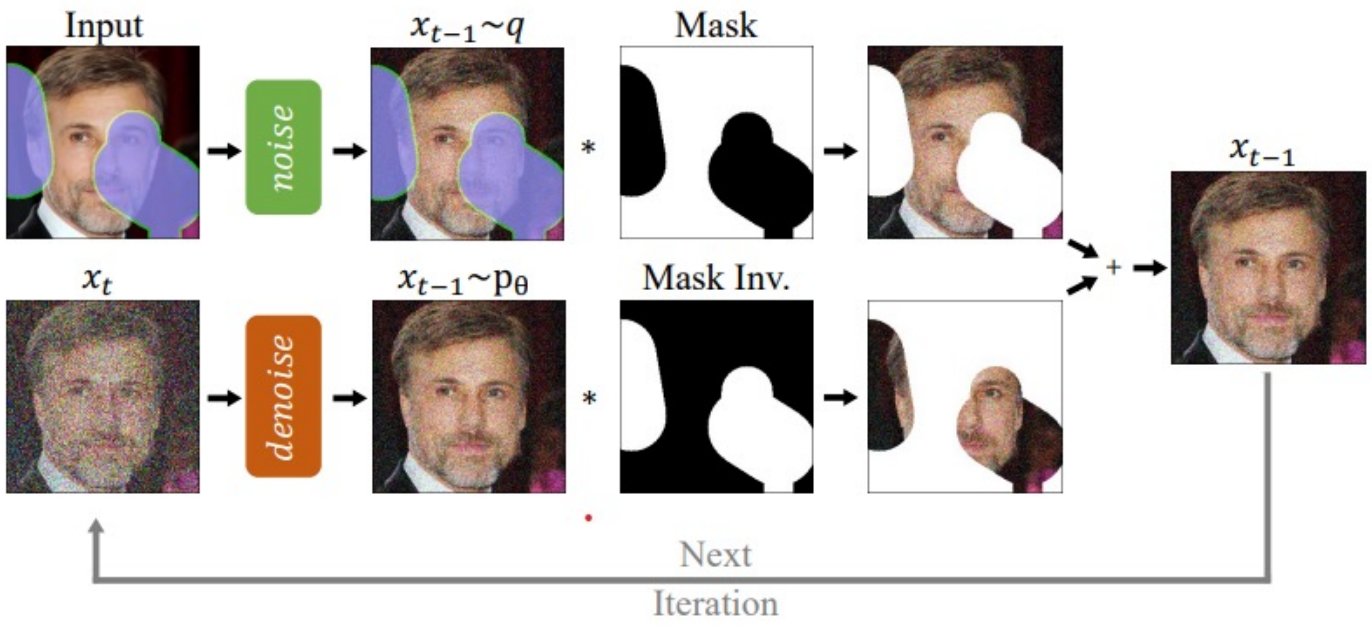
- LPIPS: a learned distance metric based on the deep feature space.
- FID/sFID/KID: 2-Wasserstein distance.
- Precision and Recall: Classification results of K-NN.
- Inception score: Inception-V3 classification statistics.
- CA: classification accuracy of a pretrained model.
- VOTES: human

Unsupervised Inpainting and Outpainting

Repainting

[2201.09865] RePaint: Inpainting using Denoising Diffusion Probabilistic Models (arxiv.org)

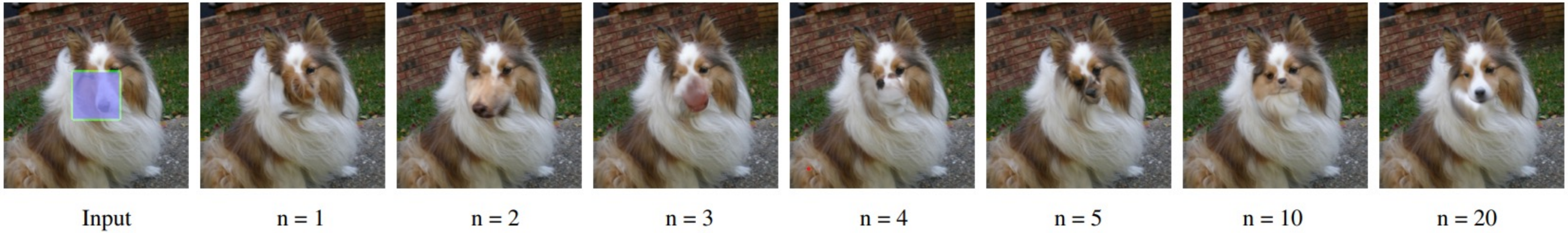
- Training a diffusion model could takes weeks and require multiple powerful GPUs, e.g., A100.
- By modifying the inference process, we can utilize a pretrained unconditional image generation model (e.g., DDPM) for inpainting and outpainting without further training.



Repainting

[2201.09865] RePaint: Inpainting using Denoising Diffusion Probabilistic Models (arxiv.org)

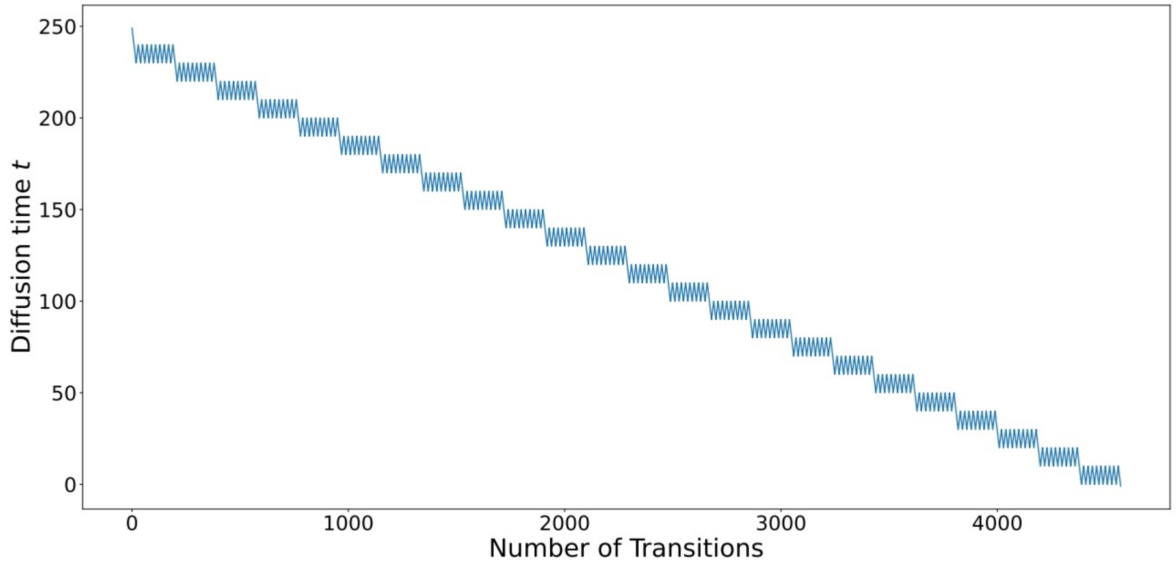
- The generated results will be correct considering the texture but wrong considering the content with single reverse pass.
- By travel back to the previous time step n times during sampling, the content will be also correct.



Repainting

[2201.09865] RePaint: Inpainting using Denoising Diffusion Probabilistic Models (arxiv.org)

- Resampling (traveling forward and backward) is time consuming.
- It takes over 4000 steps for one 256*256 image.



Iterations with resampling

Repainting Results

Original | Input | Output

inpainting

Original/Input | Output

outpainting

Classifier Guidance sampling

[2105.05233.pdf (arxiv.org)](2105.05233.pdf)

- Perturb the output towards including more information of the target class.
- p_ϕ is trained on the same noising distribution as the corresponding diffusion model.
- The classifier guidance can also be replaced by the gradient of other functions, e.g., total variation (TV).

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \underbrace{\nabla_{x_t} \log p_\phi(y|x_t)}_{\text{Classifier guidance}}, \Sigma)$
end for
return x_0 Guidance strength

Variants of Repaint

Improving Diffusion Models for Inverse Problems

[2206.00941] Improving Diffusion Models for Inverse Problems using Manifold Constraints (arxiv.org)

- We can avoid the resampling process by the gradient guidance of MCG.
- However, computing the gradient is expensive and time consuming.

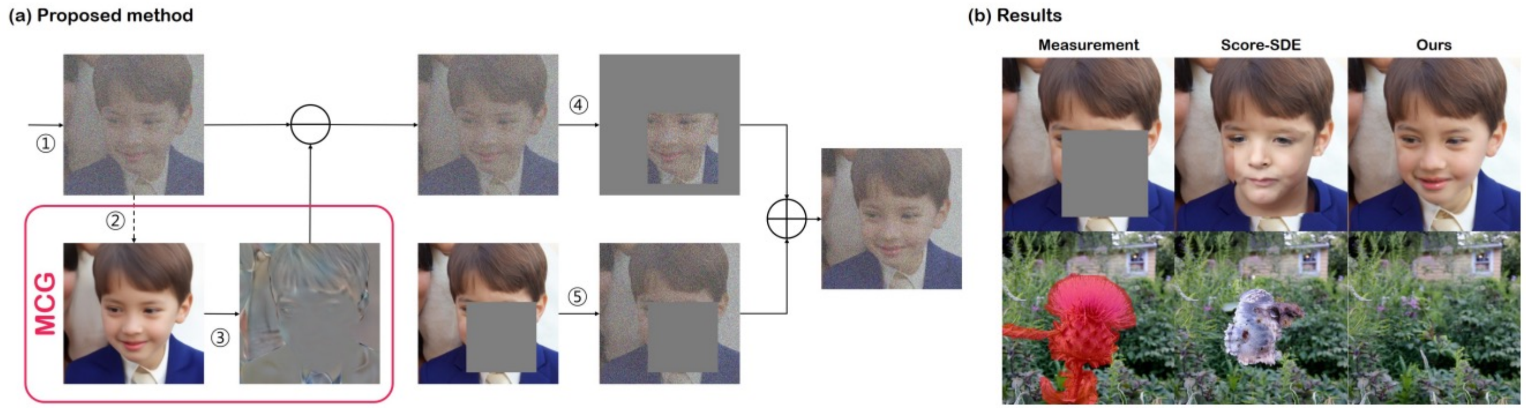


Figure 1: Visual schematic of the MCG correction step. (a) ① Unconditional reverse diffusion generates $\boldsymbol{x}_i$; ② Q_i maps the noisy $\boldsymbol{x}_i$ to generate $\hat{\boldsymbol{x}}_0$; ③ Manifold Constrained Gradient (MCG) $\frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_0)\|_2^2$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

Pseudoinverse-Guided Diffusion

pdf (openreview.net)

The problem-specific score can be decomposed via Bayes' rule:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t),$$

where the first term can be approximated with the score network $S_\theta(\mathbf{x}_t; \sigma_t)$ (Vincent, 2011), and the second term is a *guidance* term which is the score of $p_t(\mathbf{y}|\mathbf{x}_t)$.

$y = Hx_0 + n, n \sim N(0, \sigma_y)$

$p_t(\mathbf{x}_0|\mathbf{x}_t) \approx \mathcal{N}(\hat{\mathbf{x}}_t, r_t^2 \boldsymbol{I})$ $\hat{x}_t$ the estimated x_0 (DDIM) r_t depends on σ_t

Pseudoinverse-Guided Diffusion

pdf (openreview.net)

Our next step is to approximate the score of $p_t(\mathbf{y}|\mathbf{x}_t)$. Since the measurement model obtains $\mathbf{y}$ by performing a linear transform on $\mathbf{x}_0$ and adding independent Gaussian noise (Eq. 2), and $p_t(\mathbf{x}_0|\mathbf{x}_t)$ is Gaussian under our approximation (Eq. 4), the distribution of $\mathbf{y}$ conditioned on $\mathbf{x}_t$ is also Gaussian under our approximation, as follows:

$$p_t(\mathbf{y}|\mathbf{x}_t) \approx \mathcal{N}(\boldsymbol{H}\hat{\mathbf{x}}_t, r_t^2 \boldsymbol{H}\boldsymbol{H}^\top + \sigma_{\mathbf{y}}^2 \boldsymbol{I}). \tag{6}$$

Thus, we have the following approximation to the score[2]:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx \Bigg(\underbrace{(\mathbf{y} - \boldsymbol{H}\hat{\mathbf{x}}_t)^\top \left(r_t^2 \boldsymbol{H}\boldsymbol{H}^\top + \sigma_{\mathbf{y}}^2 \boldsymbol{I}\right)^{-1} \boldsymbol{H}}_{\text{vector}} \underbrace{\frac{\partial \hat{\mathbf{x}}_t}{\partial \mathbf{x}_t}}_{\text{Jacobian}} \Bigg)^\top. \tag{7}$$

This is a vector-Jacobian product and can be computed with backpropagation.

Pseudoinverse-Guided Diffusion

pdf (openreview.net)

In many cases, we have that $\sigma_{\mathbf{y}} = 0$, and thus, Eq. 7 can be simplified to:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx r_t^{-2} \Big((\boldsymbol{H}^{\dagger}\mathbf{y} - \boldsymbol{H}^{\dagger}\boldsymbol{H}\hat{\mathbf{x}}_t)^{\top} \frac{\partial \hat{\mathbf{x}}_t}{\partial \mathbf{x}_t} \Big)^{\top}; \tag{8}$$

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx r_t^{-2} \Big((h^{\dagger}(\mathbf{y}) - h^{\dagger}(h(\hat{\mathbf{x}}_t)))^{\top} \frac{\partial \hat{\mathbf{x}}_t}{\partial \mathbf{x}_t} \Big)^{\top}, \tag{9}$$

which generalizes the linear case (Eq. 8) when $h(\mathbf{x}) = \boldsymbol{H}\mathbf{x}$ and $h^{\dagger}(\mathbf{x}) = \boldsymbol{H}^{\dagger}\mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^n$.

Table 1: Comparison of different guidance methods.

Guidance	Expression	$\mathbf{x}_t \rightarrow \mathbf{y}$ differentiable	Train on $(\mathbf{x}_t, \mathbf{y})$	Noisy $\mathbf{y}$
Classifier	$\nabla_{\mathbf{x}_t} \log q(\mathbf{y}\|\mathbf{x}_t)$	Required	Yes	-
Reconstruction	$\nabla_{\mathbf{x}_t} \|\mathbf{y} - \boldsymbol{H}\hat{\mathbf{x}}_t\|_2^2$	Required	No	No
Pseudoinverse	Eqs. 7 to 9	Not required	No	Yes

Diffusion Null-space Model

[Unlimited-Size Diffusion Restoration (thecvf.com)]

- Assuming $y = Ax + n, n = 0$ or $n \sim N(0, \sigma_y)$, A is a known transformation and y is the observation, then the reverse process solves an inversion task.
- DDNM = Repaint + correction

Algorithm 1 Sampling of DDNM

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, ..., 1$ **do**
  $\mathbf{x}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\sqrt{1-\bar{\alpha}_t}\right)$
  $\hat{\mathbf{x}}_{0|t} = \mathbf{A}^{\dagger}\mathbf{y} + (\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A})\mathbf{x}_{0|t}$
  $\mathbf{x}_{t-1} \sim p(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_{0|t})$
return $\mathbf{x}_0$

Algorithm 2 Sampling of DDNM^{+}

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, ..., 1$ **do**
  $L = \min\{T - t, l\}$
  $\mathbf{x}_{t+L} \sim q(\mathbf{x}_{t+L}|\mathbf{x}_t)$
  for $j = L, ..., 0$ **do**
    $\mathbf{x}_{0|t+j} = \frac{1}{\sqrt{\bar{\alpha}_{t+j}}}\left(\mathbf{x}_{t+j} - \mathcal{Z}_{\boldsymbol{\theta}}(\mathbf{x}_{t+j}, t+j)\sqrt{1-\bar{\alpha}_{t+j}}\right)$
    $\hat{\mathbf{x}}_{0|t+j} = \mathbf{x}_{0|t+j} - \boldsymbol{\Sigma}_{t+j}\mathbf{A}^{\dagger}(\mathbf{A}\mathbf{x}_{0|t+j} - \mathbf{y})$
    $\mathbf{x}_{t+j-1} \sim \hat{p}(\mathbf{x}_{t+j-1}|\mathbf{x}_{t+j}, \hat{\mathbf{x}}_{0|t+j})$
return $\mathbf{x}_0$

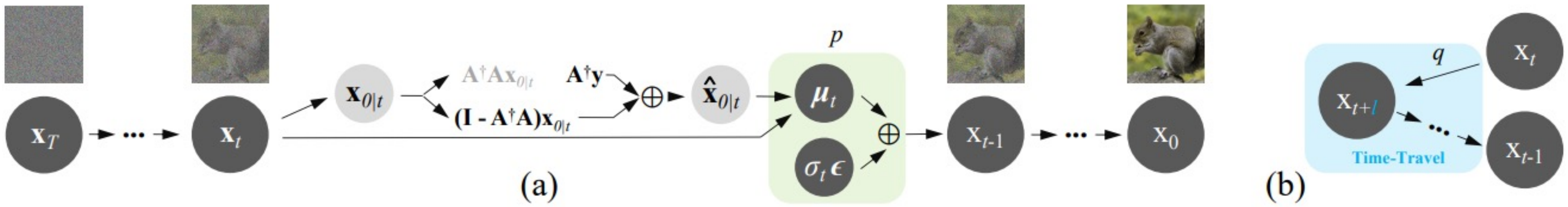


Figure 2: Illustration of (a) DDNM and (b) the time-travel trick.

Diffusion Null-space Model

[Unlimited-Size Diffusion Restoration (thecvf.com)]

The pseudo-inverse should satisfy $\mathrm{AA}^{\dagger} = \mathrm{I}$, it could be solved by SVD, or

- Inpainting: $\mathrm{AAA} = \mathrm{A}$, so A can be the mask.
- Colorization: $\mathrm{A} = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right], \mathrm{A}^{\dagger} = [1,1,1]^{\mathrm{T}}$
- Super resolution: $\mathrm{A} = \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right], \mathrm{A}^{\dagger} = [1,1,1,1]^{\mathrm{T}}$

Plug-and-Play Image Restoration

[Denoising Diffusion Models for Plug-and-Play Image Restoration (thecvf.com)](https://openaccess.thecvf.com)

- We want to solve this optimization problem:

$$\underbrace{\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma_n^2}\|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \lambda\mathcal{P}(\mathbf{x})}_{\text{x appears both terms, unstable during optimization}} \qquad \mathbf{y} = \mathcal{H}(\mathbf{x}_0) + \mathbf{n}$$

HQS algorithm

$$\begin{cases} \mathbf{z}_k = \arg\min_{\mathbf{z}} \frac{1}{2(\sqrt{\lambda/\mu})^2}\|\mathbf{z} - \mathbf{x}_k\|^2 + \mathcal{P}(\mathbf{z}) & \text{(10a)} \\ \mathbf{x}_{k-1} = \underbrace{\arg\min_{\mathbf{x}} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \mu\sigma_n^2\|\mathbf{x} - \mathbf{z}_k\|^2}_{\text{x has closed form here}}, & \text{(10b)} \end{cases}$$

$z_k = \mathrm{x}_k + n, n \sim N(0, \frac{\lambda}{\mu})$

z_k is a denoised version of x_k

x_k is a noised version of x_0

Let z_k be $\widehat{\mathrm{x}_0}$

Plug-and-Play Image Restoration

[Denoising Diffusion Models for Plug-and-Play Image Restoration (thecvf.com)]

Algorithm 1 DiffPIR

Require: $\mathbf{s}_\theta, T, \mathbf{y}, \sigma_n, \{\bar{\sigma}_t\}_{t=1}^{T}, \zeta, \lambda$

Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, pre-calculate $\rho_t \triangleq \lambda\sigma_n^2/\bar{\sigma}_t^2$.

for $t = T$ **to** 1 **do**

Solve optimization at t

$\mathbf{x}_0^{(t)} = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t + (1 - \bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t))$ // *Predict $\hat{\mathbf{z}}_0$ with score model as denoiser*

$\hat{\mathbf{x}}_0^{(t)} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \rho_t\|\mathbf{x} - \mathbf{x}_0^{(t)}\|^2$ // *Solving data proximal subproblem*

x_t depends on both ϵ and x_0,
ϵ should be updated when $\widehat{x_0}$ changes

$\hat{\epsilon} = \frac{1}{\sqrt{1-\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0^{(t)})$ // *Calculate effective $\hat{\epsilon}(\mathbf{x}_t, \mathbf{y})$*

Sampling based on DDIM

$\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0^{(t)} + \sqrt{1 - \bar{\alpha}_{t-1}}(\sqrt{1-\zeta}\hat{\epsilon} + \sqrt{\zeta}\epsilon_t)$ // *Finish one step reverse diffusion sampling*

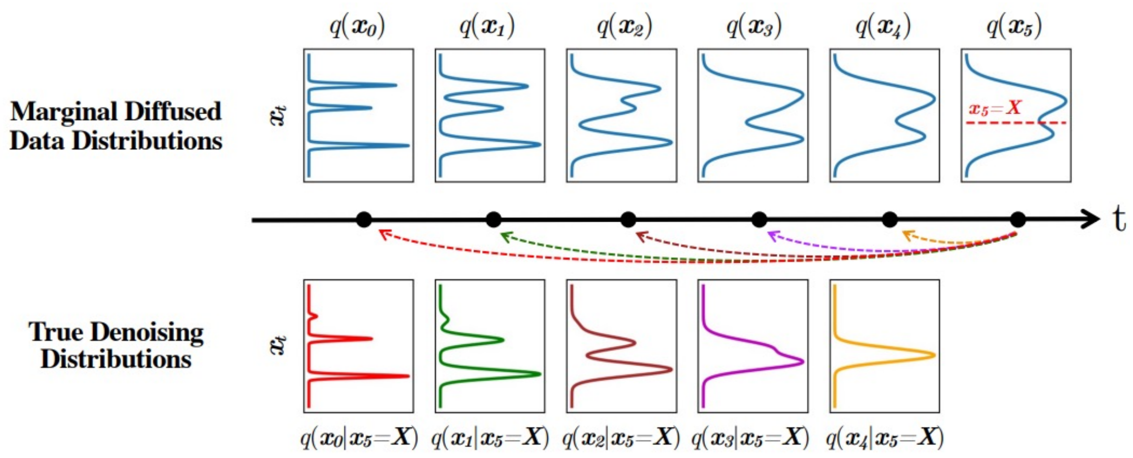
end for

return $\mathbf{x}_0$

Diffusion-GAN and Wavelet Diffusion

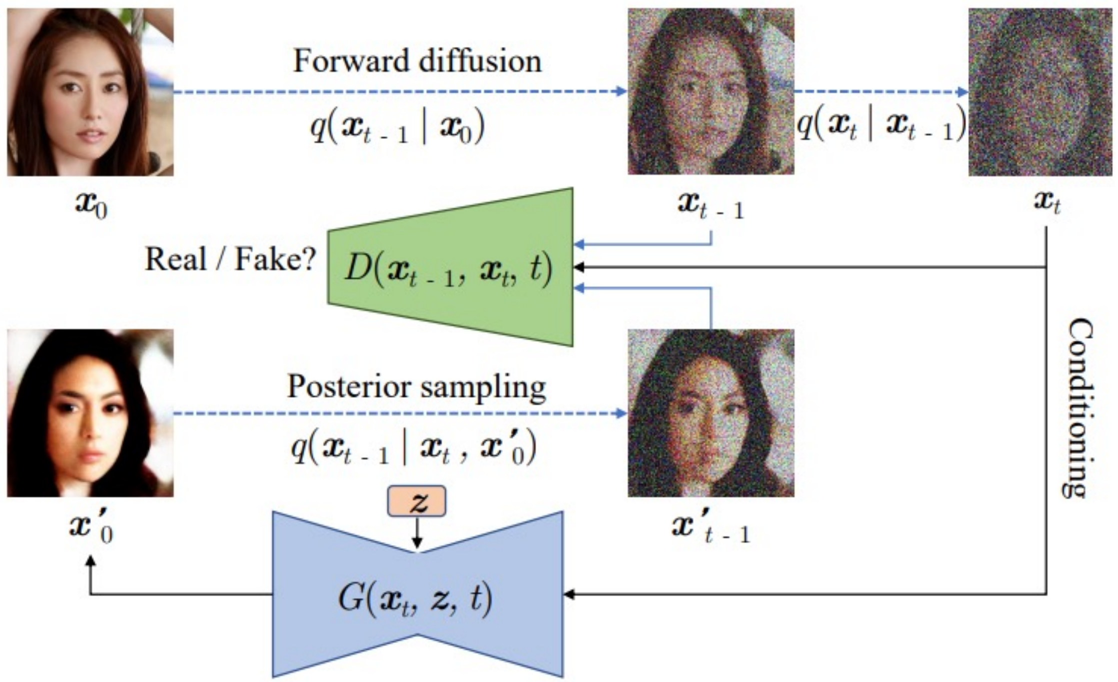
Diffusion GAN

[2112.07804.pdf (arxiv.org)](https://arxiv.org/pdf/2112.07804.pdf)



Advantages:

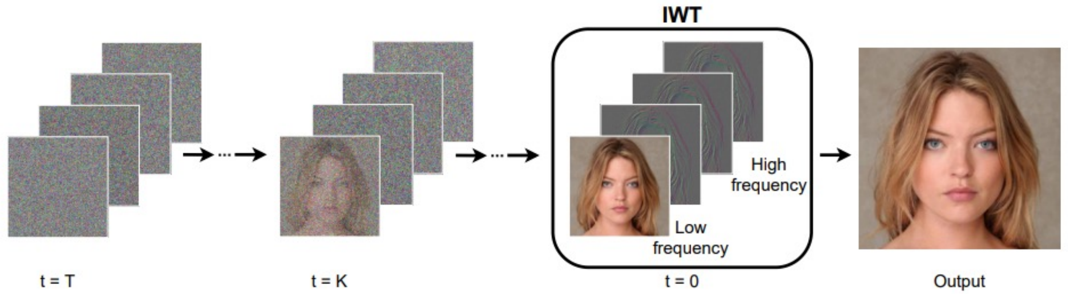
- Consistent with DDPM
- Using a single network to predict x_{t-1} directly at different t may be difficult.



Wavelet Diffusion Model

[Wavelet Diffusion Models Are Fast and Scalable Image Generators (thecvf.com)](https://thecvf.com)

The wavelet diffusion model is faster and smaller than the Diffusion GAN.



Adversarial objective Following [49], we optimize the generator and the discriminator through the adversarial loss:

$$\mathcal{L}_{adv}^{D} = -\log(D(y_{t-1}, y_t, t)) + \log(D(y'_{t-1}, y_t, t)),$$
$$\mathcal{L}_{adv}^{G} = -\log(D(y'_{t-1}, y_t, t)). \quad (4)$$

Reconstruction term In addition to the adversarial objective in Eq. (4), we add a reconstruction term to not only impede the loss of frequency information but also preserve the consistency of wavelet subbands. It is formulated as an L1 loss between a generated image and its ground-truth:

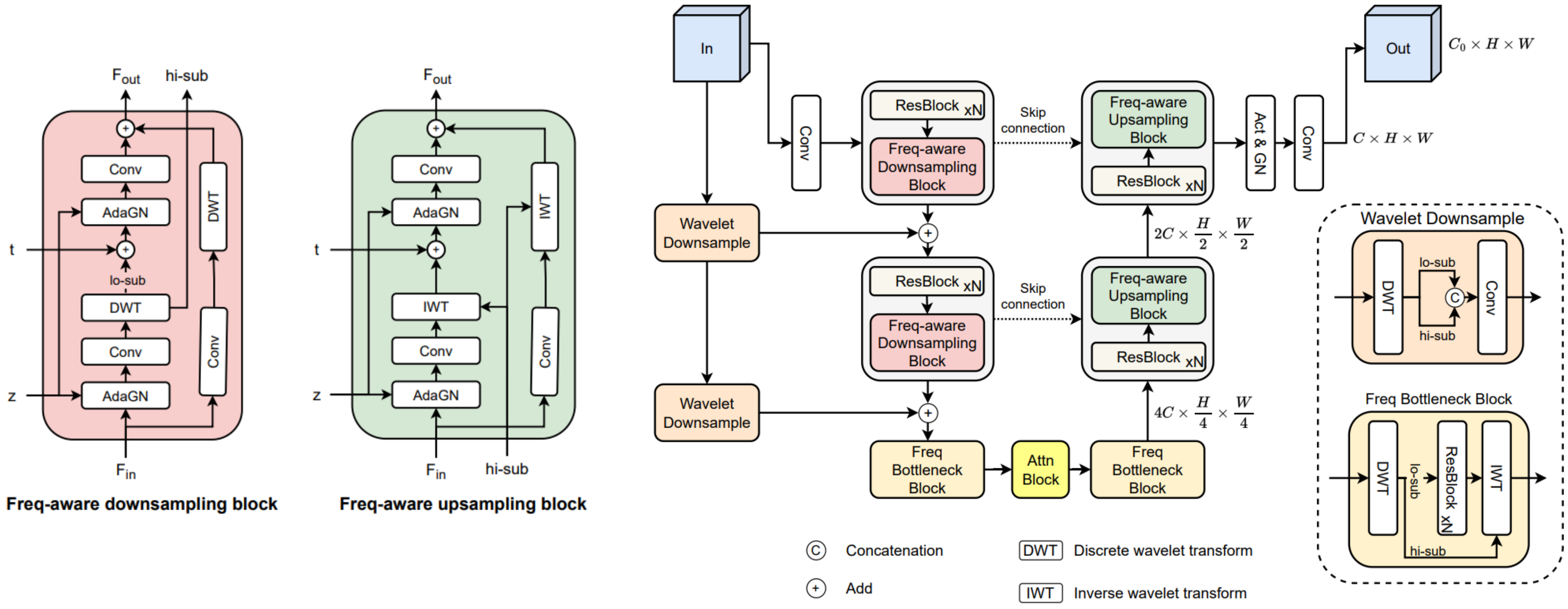
$$\mathcal{L}_{rec} = \|y'_0 - y_0\|. \quad (5)$$

The overall objective of the generator is a linear combination of adversarial loss and reconstruction loss:

$$\mathcal{L}^{G} = \mathcal{L}_{adv}^{G} + \lambda \mathcal{L}_{rec}, \quad (6)$$

Wavelet Diffusion Model

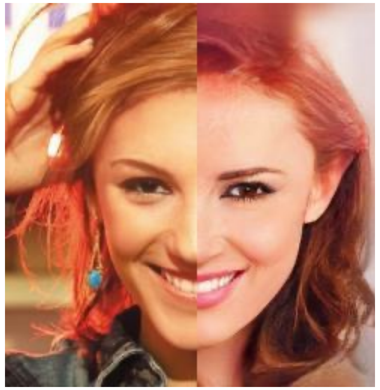
[Wavelet Diffusion Models Are Fast and Scalable Image Generators (thecvf.com)]



Blending

[Gradient Domain Fusion Using Poisson Blending (brown.edu)]

- Applying repainting on the wavelet diffusion model will generate results with the distribution shift.
- The Poisson Blending algorithm is introduced as a postprocessing step to solve the issue.



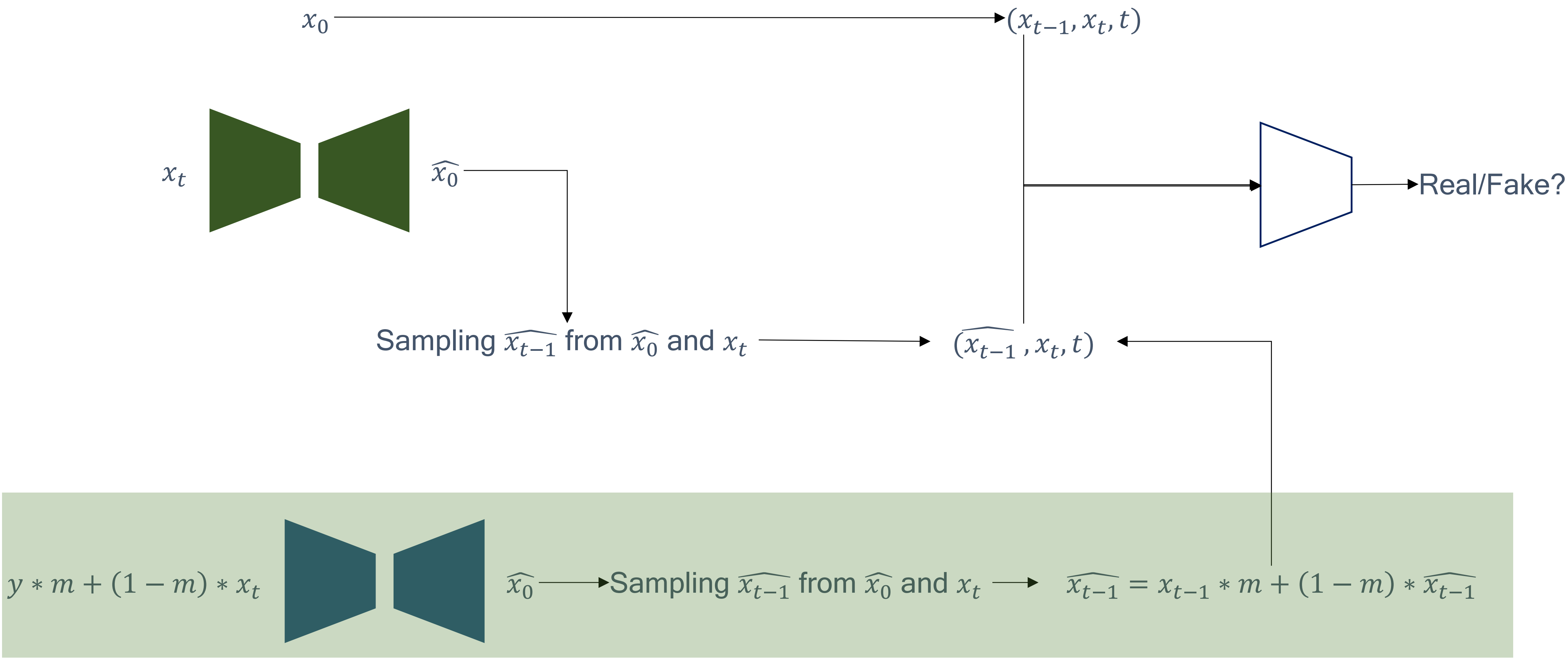
Repaint

Poisson blend

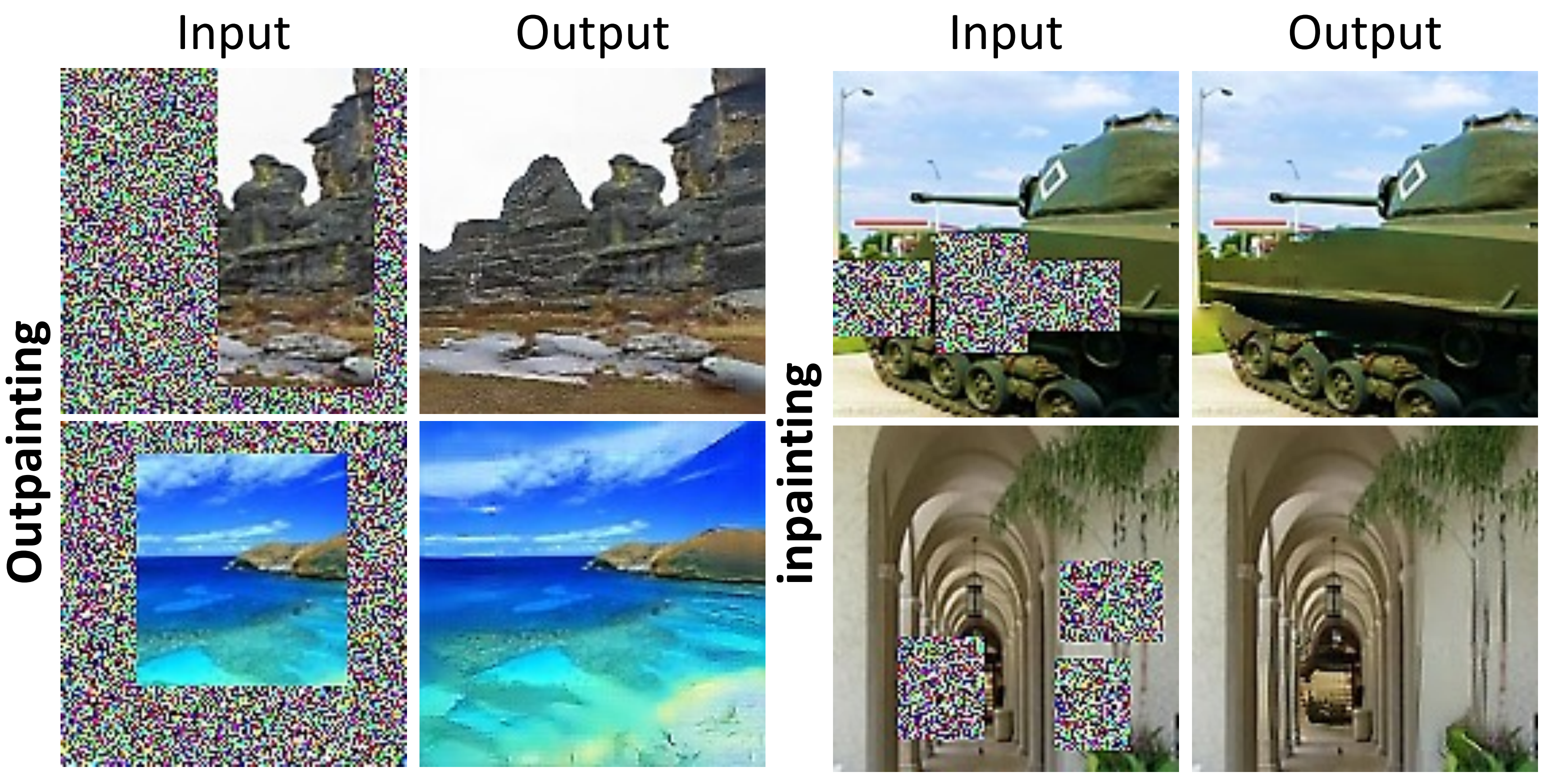
Outpainting the right half of the face.

Supervised Inpainting and Outpainting

Supervised Inpainting and Outpainting



Results



Takeaways

- The mask should be given to both discriminator and the generator.
- The inference results depend more on the design of the discriminators.
- Without the Haar wavelet transformation, the model fails to generate high frequency details.
- If the input of the discriminator is the Haar wavelet features, the results include more high-frequency details, but also include checker-box artifacts.
- The wavelet generator architecture does not provide high-frequency details if the input of the discriminator is the image but not the Haar wavelet feature.
- The results include high frequency details w/o the artifacts if using two discriminators, one works in the image domain and the other works in the wavelet domain.
- Reconstruction loss is important to make the training faster and stable.
- More discriminators could be incorporated, e.g., the patch-wised discriminator.
- More reconstruction loss could be considered, e.g., perceptual loss.

Free-Size Inpainting and Outpainting

Free-Size Inpainting and Outpainting

[Unlimited-Size Diffusion Restoration (thecvf.com)]

To reuse a low resolution inpainting/outpainting model, we can first inpaint the low-resolution image patch by patch and then inpaint the high-resolution image condition on the low-resolution results.

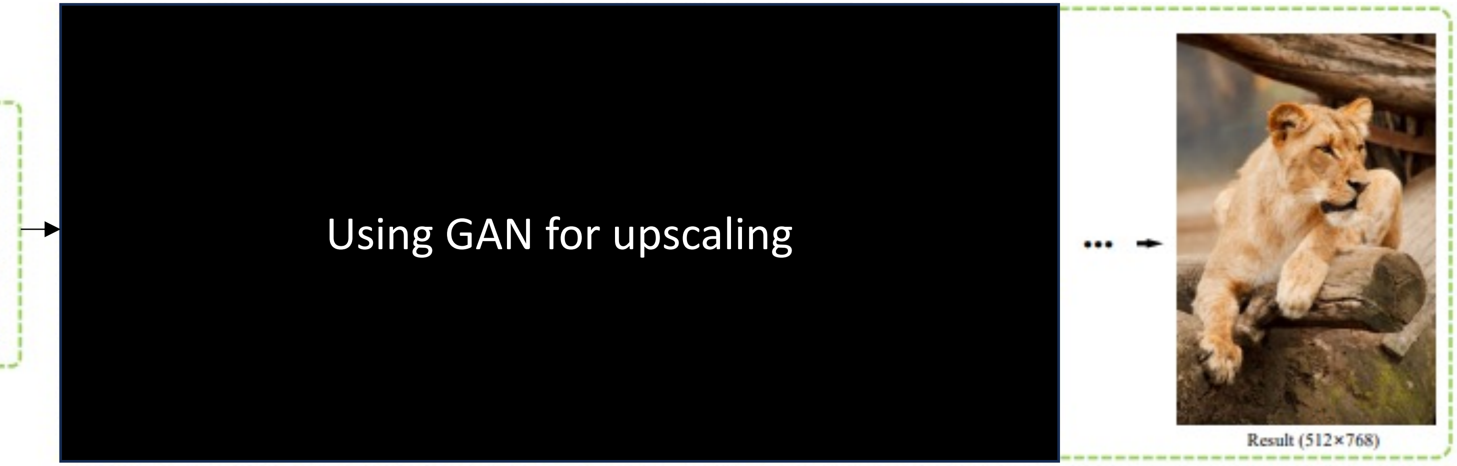
(a) Phase 1 of HiR

(b) Phase 2 of HiR

Future work: Diffusion + GAN

- Using diffusion models for patch-wised inpainting/outpainting is time-consuming.
- The other option is first using the diffusion model to inpaint the low-resolution image and then upscaling the image using a GAN to the target resolution.

(a) Phase 1 of HiR



(b) Phase 2 of HiR

Data Sources

- Places365 MIT Places Database for Scene Recognition
- CelebA HQ CelebA-HQ resized (256x256) | Kaggle

Conclusion

- The diffusion model is a proven to generating realistic results for inpainting and outpainting.
- The results of unsupervised methods are realistic, but they are time consuming and are biased to the pretrained dataset.
- The supervised method is 1000 times faster than the unsupervised methods, but it requires a large dataset and is time consuming during training.
- The supervised method is potential to be applied on mobile devices.