I2SB: Image-to-Image Schrödinger Bridge

Presenter: Yifan Wang

Introduction

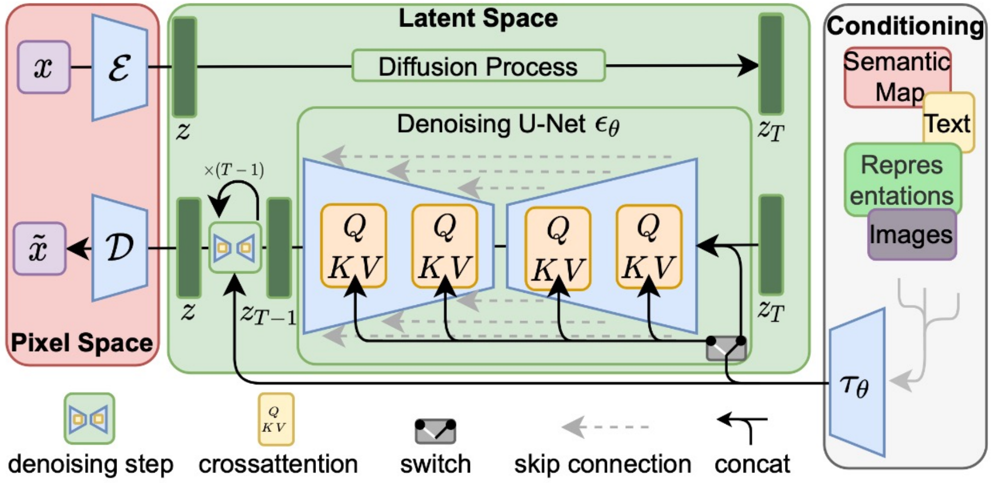
Image restoration (Conditional generation)

- ❖ Super-resolution.
- ❖ Inpainting.
- ❖ Deblurring.

Introduction

Previous Approaches

❖ Conditional Diffusion model. $s_\theta(x_t, t) \rightarrow s_\theta(x_t, t, y)$

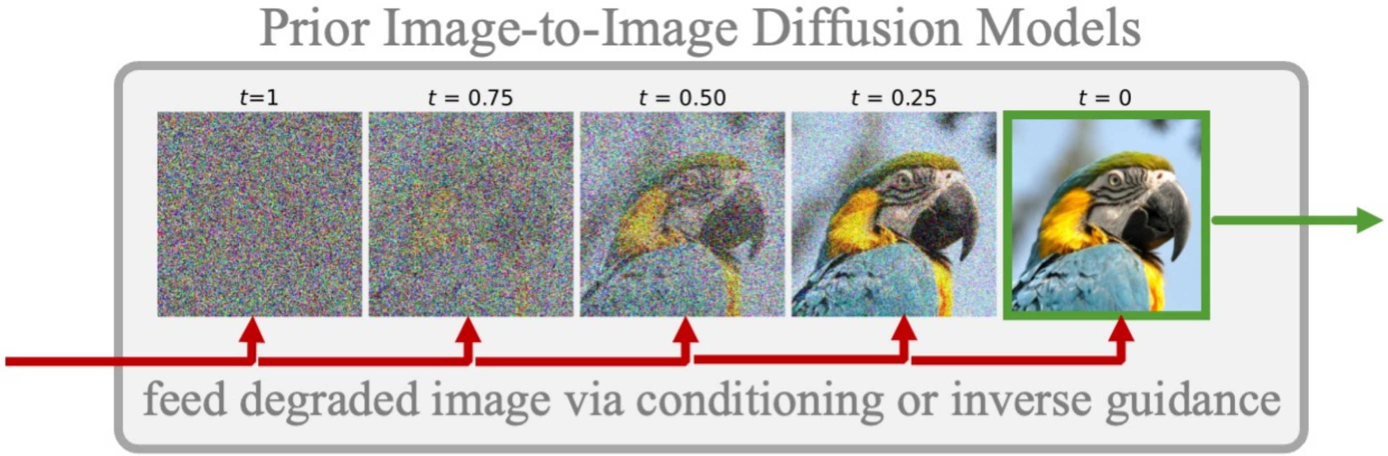


❖ Inverse problem model. $y = Ax + n$

Introduction

Drawbacks

❖ Begin with Gaussian Noise



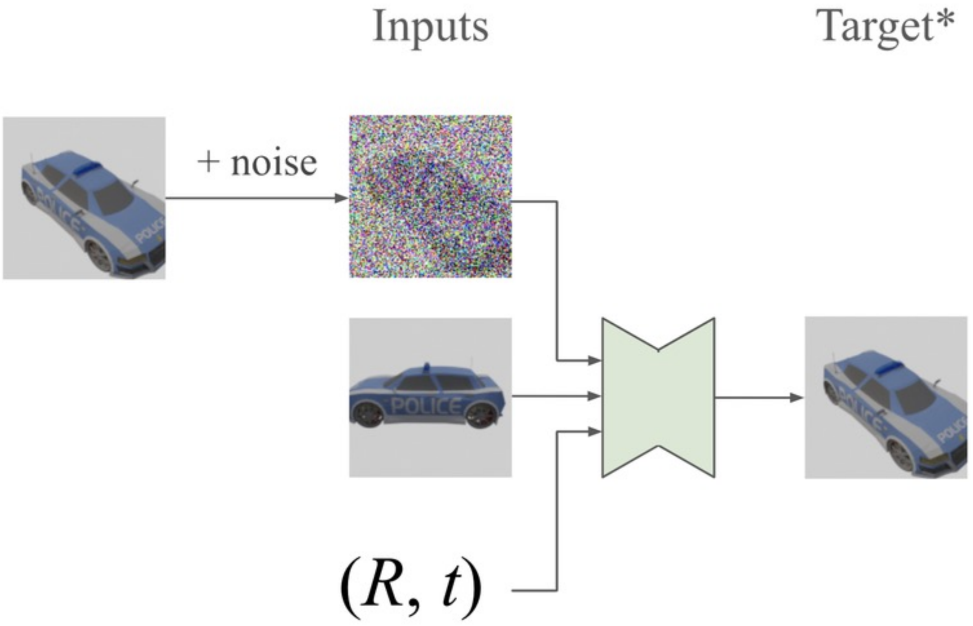
$X_1 \sim \mathcal{N}(0,1)$

$X_0 \sim p_A$

Introduction

Drawbacks and idea

- Degraded images are much more structurally informative compared to random noise.



- Start the generative processes from degraded images ?

Introduction

Schrodinger bridge (SB) Model

- A generalized **nonlinear** score-based model which defines optimal transport **between two arbitrary distributions**.

- Computationally **unfavorable.**

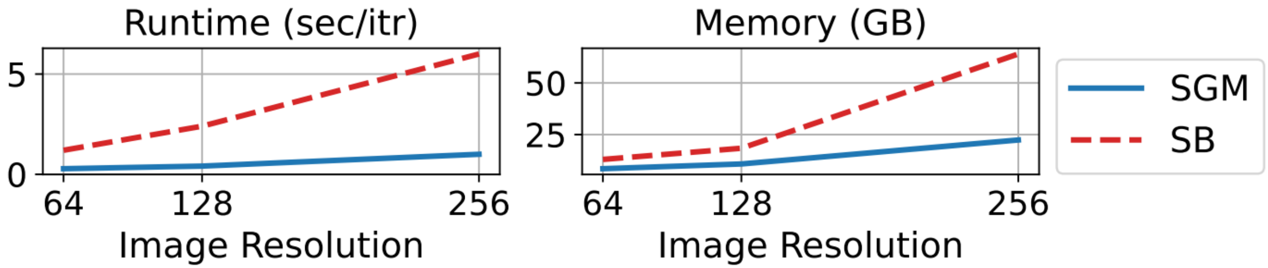


Figure 2. Complexity of SGM and SB (Chen et al., 2021a) On 256×256 resolution, SB is 6× slower and consumes 3× memory.

Main Contribution

Main contribution of I2SB

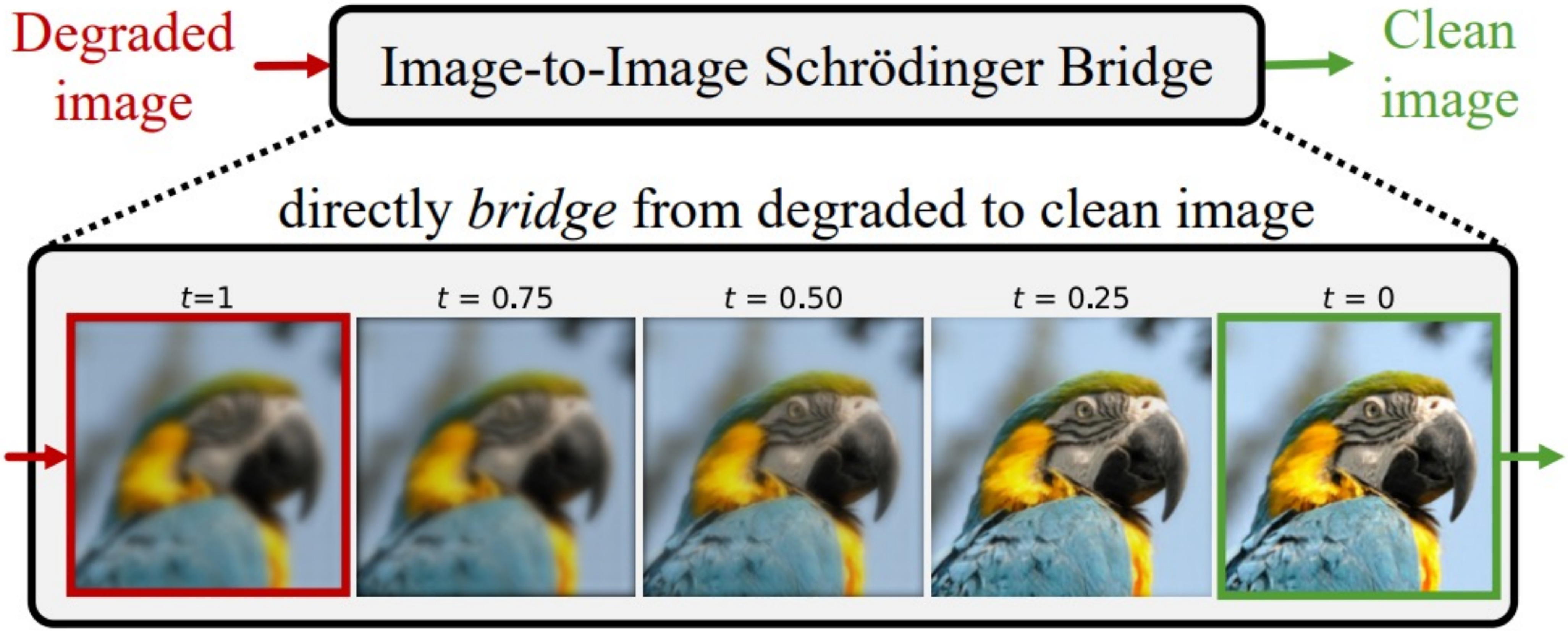
- ❖ Computational framework used in standard score-based models. ✓

+

- ❖ SB has the ability to transport between two arbitrary distributions. ✓

I2SB is a sub-class of SB with nonlinear diffusion models that **share** the same computational framework used in standard score-based models.

Main Contribution



Preliminaries

Score-based Generative Model (SGM)

$$dX_t = f_t(X_t)dt + \sqrt{\beta_t}dW_t \qquad X_1 \sim \mathcal{N}(0,1) \qquad X_0 \sim p_A$$

- ❑ β_t is a properly choosing the diffusion coefficient.
- ❑ $f_t(X_t)$ is called the base drift, usually **linear** in X_t

$$f_t(X_t) = -\frac{\beta_t}{2}X_t$$

$$dX_t = [f_t - \beta_t \nabla log p(X_t, t)]dt + \sqrt{\beta_t}d\widehat{W_t}$$

Preliminaries

Schrodinger Bridge (SB)

- SB an entropy regularized optimal transport model:

$$\mathrm{d}X_t = [f_t + \beta_t \nabla \log \Psi(X_t, t)]\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}W_t$$

$$\mathrm{d}X_t = [f_t - \beta_t \nabla \log \widehat{\Psi}(X_t, t)]\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\widehat{W}_t$$

$$X_0 \sim p_A \qquad X_1 \sim p_B$$

$$\begin{cases} \frac{\partial \Psi(x,t)}{\partial t} = -\nabla \Psi^\top f - \frac{1}{2}\beta \Delta \Psi \\ \frac{\partial \widehat{\Psi}(x,t)}{\partial t} = -\nabla \cdot (\widehat{\Psi} f) + \frac{1}{2}\beta \Delta \widehat{\Psi} \end{cases}$$

$$\text{s.t. } \Psi(x,0)\widehat{\Psi}(x,0) = p_{\mathcal{A}}(x),\ \Psi(x,1)\widehat{\Psi}(x,1) = p_{\mathcal{B}}(x)$$

Preliminaries

SGM is a special case of SB | SB generalizes SGM to nonlinear structure

SGM

$$dX_t = f_t(X_t)dt + \sqrt{\beta_t}dW_t$$

$$\mathrm{d}X_t = [f_t - \beta_t \nabla \log p(X_t, t)]\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\widehat{W}_t$$

$$\mathrm{X}_0 \sim \mathrm{p}_\mathrm{A} \qquad \mathrm{X}_1 \sim \mathcal{N}(0,1)$$

SB

Usually, nonlinear

$$\mathrm{d}X_t = [f_t + \beta_t \nabla \log \Psi(X_t, t)]\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}W_t$$

$$\mathrm{d}X_t = [f_t - \beta_t \nabla \log \widehat{\Psi}(X_t, t)]\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\widehat{W}_t$$

$$\mathrm{X}_0 \sim \mathrm{p}_\mathrm{A} \qquad \mathrm{X}_1 \sim \mathrm{p}_\mathrm{B}$$

- $\beta_t \nabla \log \widehat{\Psi}(X_t, t)$ is no longer the score function

Preliminaries

SGM is a special case of SB

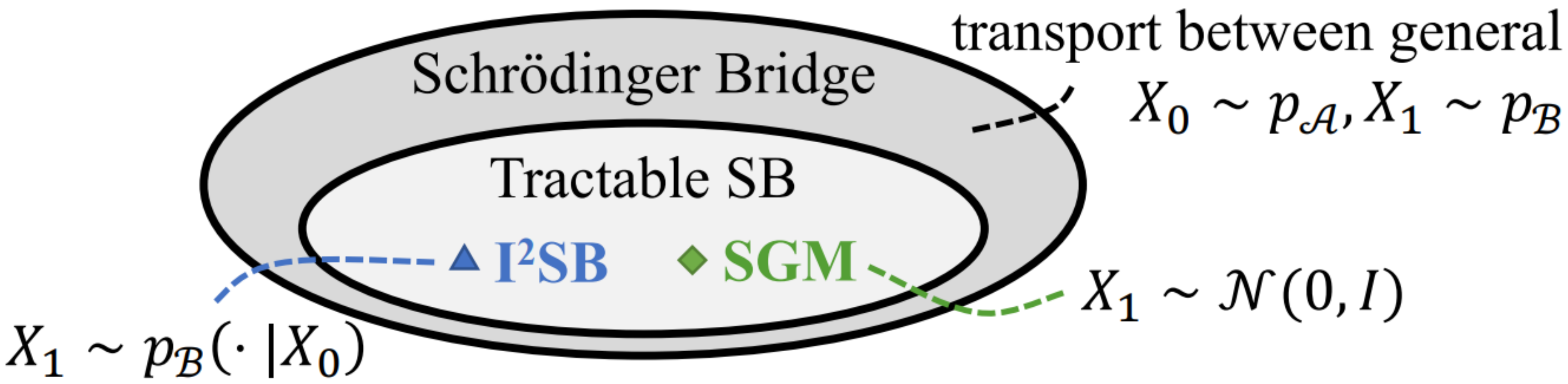


Figure 3. **I²SB** belongs to a tractable class of SB that shares the same computational framework of SGM and rebases the terminal distribution beyond simple Gaussian priors.

Solving SB using SGM Framework

Notice that

- The nonlinear drifts in SB ($\nabla \log \widehat{\Psi}(X_t, t)$) resemble the score function if we view $\Psi(X_t, t)$ and $\widehat{\Psi}(X_t, t)$ as the densities.
- **Equation below** gives the solution to the Fokker-Plank equation that characterizes the marginal density induced by the linear SDE in SGM.

$$dX_t = f_t(X_t)dt + \sqrt{\beta_t}dW_t$$

$$\begin{cases} \frac{\partial \Psi(x,t)}{\partial t} = -\nabla \Psi^\top f - \frac{1}{2}\beta \Delta \Psi \\ \frac{\partial \widehat{\Psi}(x,t)}{\partial t} = -\nabla \cdot (\widehat{\Psi} f) + \frac{1}{2}\beta \Delta \widehat{\Psi} \end{cases}$$

$$\text{s.t. } \Psi(x,0)\widehat{\Psi}(x,0) = p_{\mathcal{A}}(x), \Psi(x,1)\widehat{\Psi}(x,1) = p_{\mathcal{B}}(x)$$

Solving SB using SGM Framework

Theorem 3.1

When the Schrodinger systems hold, $\nabla \log \widehat{\Psi}(X_t, t)$ and $\nabla \log \Psi(X_t, t)$ are the score functions of the following **linear** SDEs, respectively:

$$dX_t = f_t(X_t)dt + \sqrt{\beta_t} dW_t \qquad \boldsymbol{X_0 \sim \widehat{\Psi}(\cdot, 0)}$$

$$dX_t = f_t(X_t)dt + \sqrt{\beta_t} d\widehat{W}_t \qquad X_1 \sim \Psi(\cdot, 1)$$

Solving SB using SGM Framework

Ideas from Theorem 3.1

- The backward drift $\nabla \log \widehat{\Psi}(X_t, t)$ can be used to reverse the following SDE:

$$dX_t = f_t(X_t)dt + \sqrt{\beta_t}dW_t \qquad\qquad X_0 \sim \widehat{\Psi}(\cdot, 0)$$

$$\mathrm{d}X_t = \left[f_t - \beta_t \nabla \log \widehat{\Psi}(X_t, t)\right]\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\widehat{W_t}$$

- Essentially, the **nonlinearity** of $\nabla \log \widehat{\Psi}(X_t, t)$, as the combination of the nonlinear forward drift and its score function, is **absorbed into the initial condition**. $\mathrm{X}_0 \sim \widehat{\Psi}(\cdot, 0)$

Solving SB using SGM Framework

Ideas from Theorem 3.1

- Hence, **if we can draw samples** from $X_0 \sim \widehat{\Psi}(\cdot, 0)$, then we can parameterize $\nabla \log \widehat{\Psi}(X_t, t)$ with the score network and apply practical techniques from SGM.

- However, more work is needed to make this **"if" is true.**

$$\begin{cases} \frac{\partial \Psi(x,t)}{\partial t} = -\nabla \Psi^\top f - \frac{1}{2}\beta \Delta \Psi \\ \frac{\partial \widehat{\Psi}(x,t)}{\partial t} = -\nabla \cdot (\widehat{\Psi} f) + \frac{1}{2}\beta \Delta \widehat{\Psi} \end{cases}$$

$$\text{s.t. } \Psi(x,0)\widehat{\Psi}(x,0) = p_{\mathcal{A}}(x),\ \Psi(x,1)\widehat{\Psi}(x,1) = p_{\mathcal{B}}(x)$$

Solving SB using SGM Framework

Corollary 3.2

Let $p_A(\cdot) = \delta_a(\cdot)$ be the Dirac delta distribution centered at a. Then:

$$\hat{\Psi}(\cdot, 0) = \delta_a(\cdot) \qquad\qquad \Psi(\cdot, 1) = \frac{\mathrm{p_B}}{\hat{\Psi}(\cdot, 1)}$$

- This breaks the dependency on $\Psi(\cdot,\cdot)$ for solving $\hat{\Psi}(\cdot, 0)$.
- Adopts the same boundary $\delta_a(\cdot)$ on one side and generalizes the other side from Gaussian to arbitrary $\mathrm{p_B}$.

Solving SB using SGM Framework

A summary

- The above theories suggest an efficient pipeline for training

$$\nabla \log \widehat{\Psi}(X_t, t)$$

  - This will be our new “score function” in I2SB.
  - Based on Theorem 3.1, we do not need to dealing with the intractability of reversing the nonlinear forward drift.
  - In this way, we formulated a tractable SB compatible with the SGM framework.

Algorithmic Design

Training

- Training scalable diffusion models requires efficient computation of X_t.
- Assume pair information is available during training $(\mathrm{X}_0, \mathrm{X}_1)$:

$$\mathrm{p}_A(\mathrm{X}_0) \qquad \mathrm{p}_B(\mathrm{X}_1|\mathrm{X}_0)$$

Algorithmic Design

Proposition 3.3

$$dX_t = [f_t + \beta_t \nabla \log \Psi(X_t, t)]dt + \sqrt{\beta_t} dW_t$$

The posterior of (5) given some boundary pair (X_0, X_1), admits an analytic form:

$$q(X_t | X_0, X_1) = \mathcal{N}(X_t; \mu_t(X_0, X_1), \Sigma_t)$$

$$\mu_t = \frac{\widehat{\sigma_t}^2}{\widehat{\sigma_t}^2 + \sigma_t^2} X_0 + \frac{\sigma_t^2}{\widehat{\sigma_t}^2 + \sigma_t^2} X_1 \qquad \Sigma_t = \frac{\sigma_t^2 \widehat{\sigma_t}^2}{\widehat{\sigma_t}^2 + \sigma_t^2}$$

$$\sigma_t^2 := \int_0^t \beta_\tau d\tau \qquad \widehat{\sigma_t}^2 := \int_t^1 \beta_\tau d\tau$$

Algorithmic Design

Proposition 3.3

Further, this posterior marginalizes the recursive posterior sampling in DDPM

$$q(X_n | X_0, X_N) = \int \prod_{k=n}^{N-1} p(X_k | X_0, X_{k+1}) dX_{k+1}$$

Algorithmic Design

Proposition 3.3

- During training when (X_0, X_1) are available, we can sample X_t **directly** from Proposition 3.3 **without** solving any nonlinear diffusion as in prior SB models

- During generation when **only** $\mathbf{X_1} \sim \mathbf{p_B}$ is given, running standard DDPM starting from X_1 induces the same marginal density of SB paths.

Algorithmic Design

Parameterization & Objective

- I2SB adopt the same network parameterization $\varepsilon(X_t, t: \theta)$ from SGM as the score function $\nabla \log \widehat{\Psi}(X_t, t)$, with previous defined X_t:

$$||\varepsilon(X_t, t; \theta) - \frac{X_t - X_1}{\sigma_t}||$$

Algorithmic Design

Algorithm 1 Training

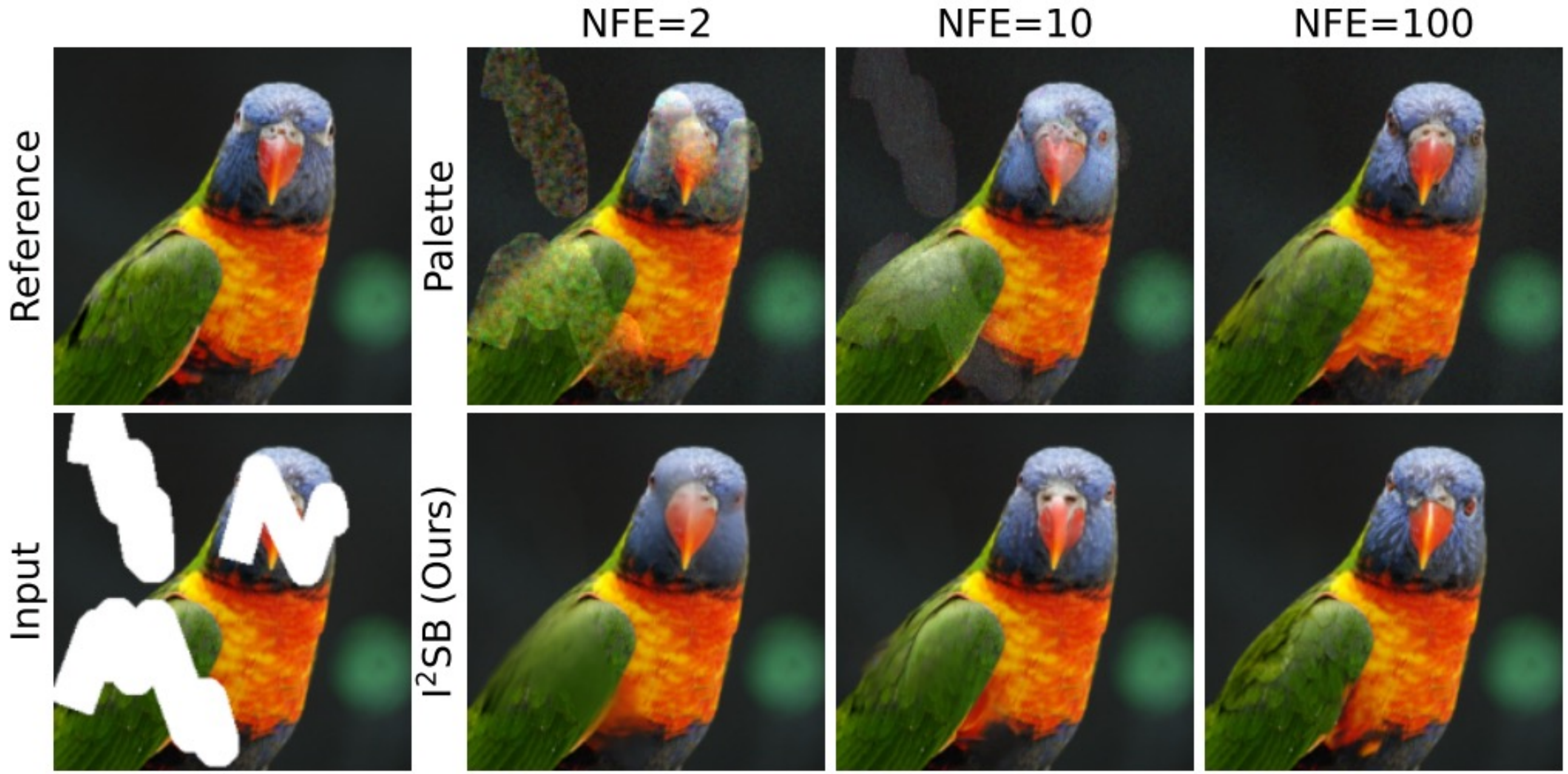
Input: clean $p_{\mathcal{A}}(\cdot)$ and degraded $p_{\mathcal{B}}(\cdot|X_0)$ datasets
repeat
  $t \sim \mathcal{U}([0,1])$, $X_0 \sim p_{\mathcal{A}}(X_0)$, $X_1 \sim p_{\mathcal{B}}(X_1|X_0)$
  $X_t \sim q(X_t|X_0, X_1)$ according to (11)
  Take gradient descent step on $\epsilon(X_t, t; \theta)$ using (12)
until converges

Algorithm 2 Generation

Input: $X_N \sim p_{\mathcal{B}}(X_N)$, trained $\epsilon(\cdot, \cdot; \theta)$
for $n = N$ **to** 1 **do**
  Predict X_0^{ϵ} using $\epsilon(X_n, t_n; \theta)$
  $X_{n-1} \sim p(X_{n-1}|X_0^{\epsilon}, X_n)$ according to DDPM (4)
end for
return X_0

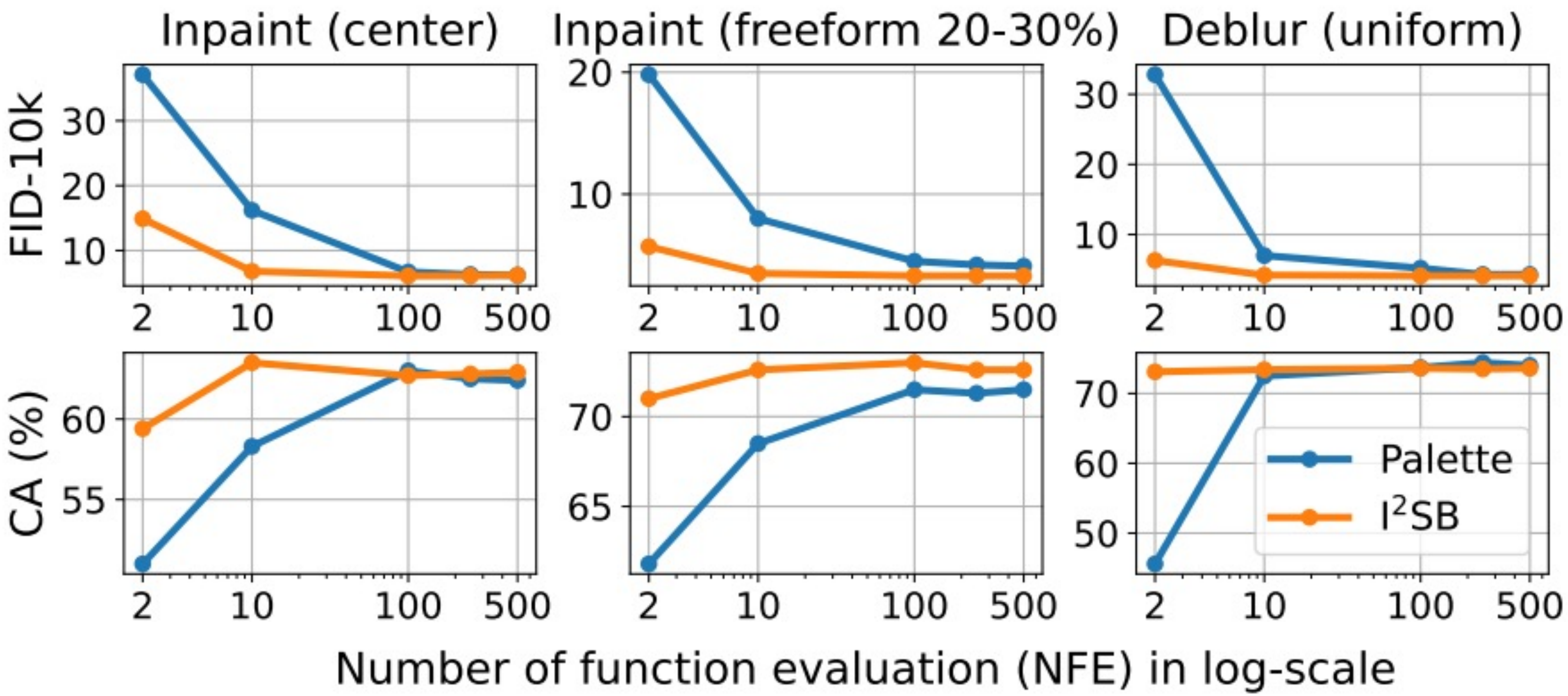
Experiment / Result

Number of function evaluation (NFE)



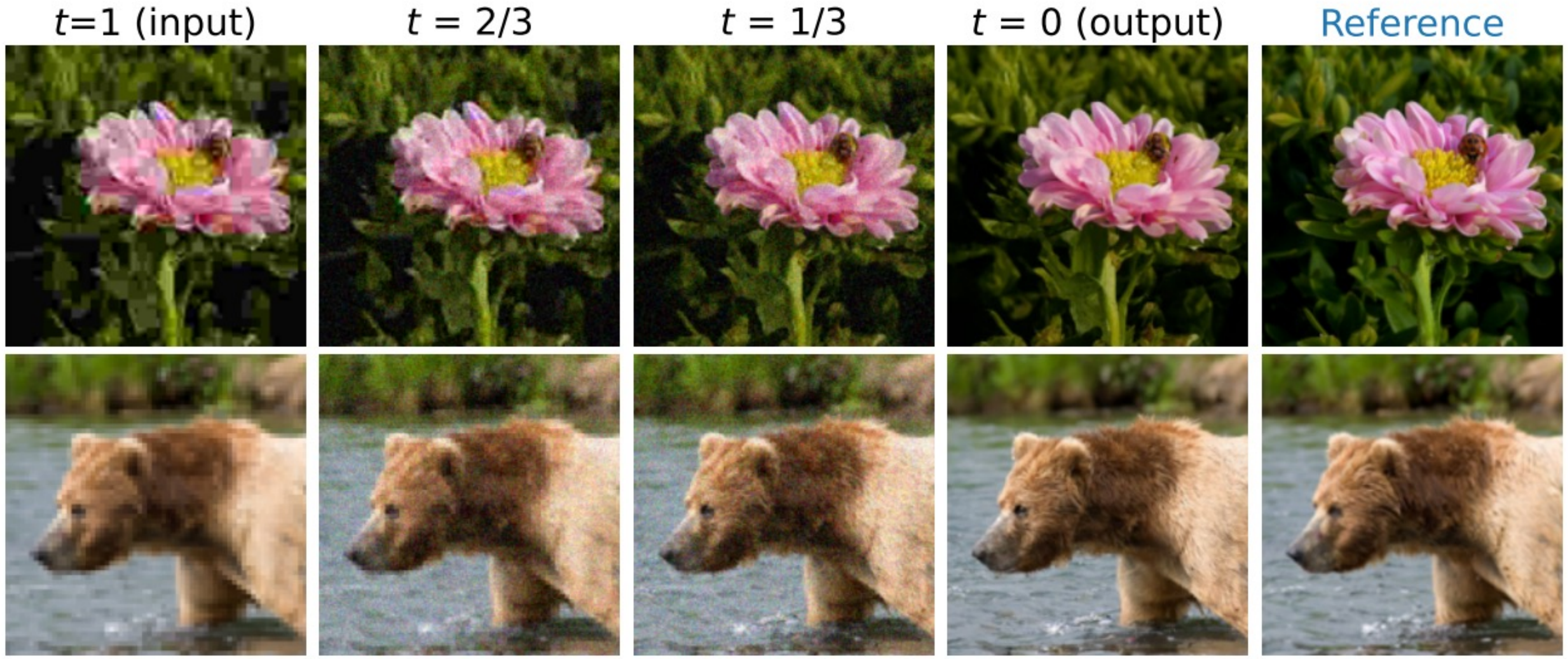
Experiment / Result

Number of function evaluation (NFE)



Experiment / Result

More natural and interpretable generative diffusion processes from degraded to clean images



Experiment / Result

General image-to-image translation

Quiz

? Will there still have diversity among the generated images when using I2SB with fixed condition image as input?

Quiz

? Will there still have diversity among the generated images when using I2SB with fixed condition image as input?

Quiz

? Is it necessary to have access to paired data (such as clean and degraded image pairs) during the training process of I2SB?

Quiz

? Is it necessary to have access to paired data (such as clean and degraded image pairs) during the training process of I2SB?

Yes, the tractability of I2SB requires knowing paired data during training.

$$q(X_t|X_0, X_1) = \mathcal{N}(X_t; \mu_t(X_0, X_1), \Sigma_t)$$

This is a limitation of I2SB method.