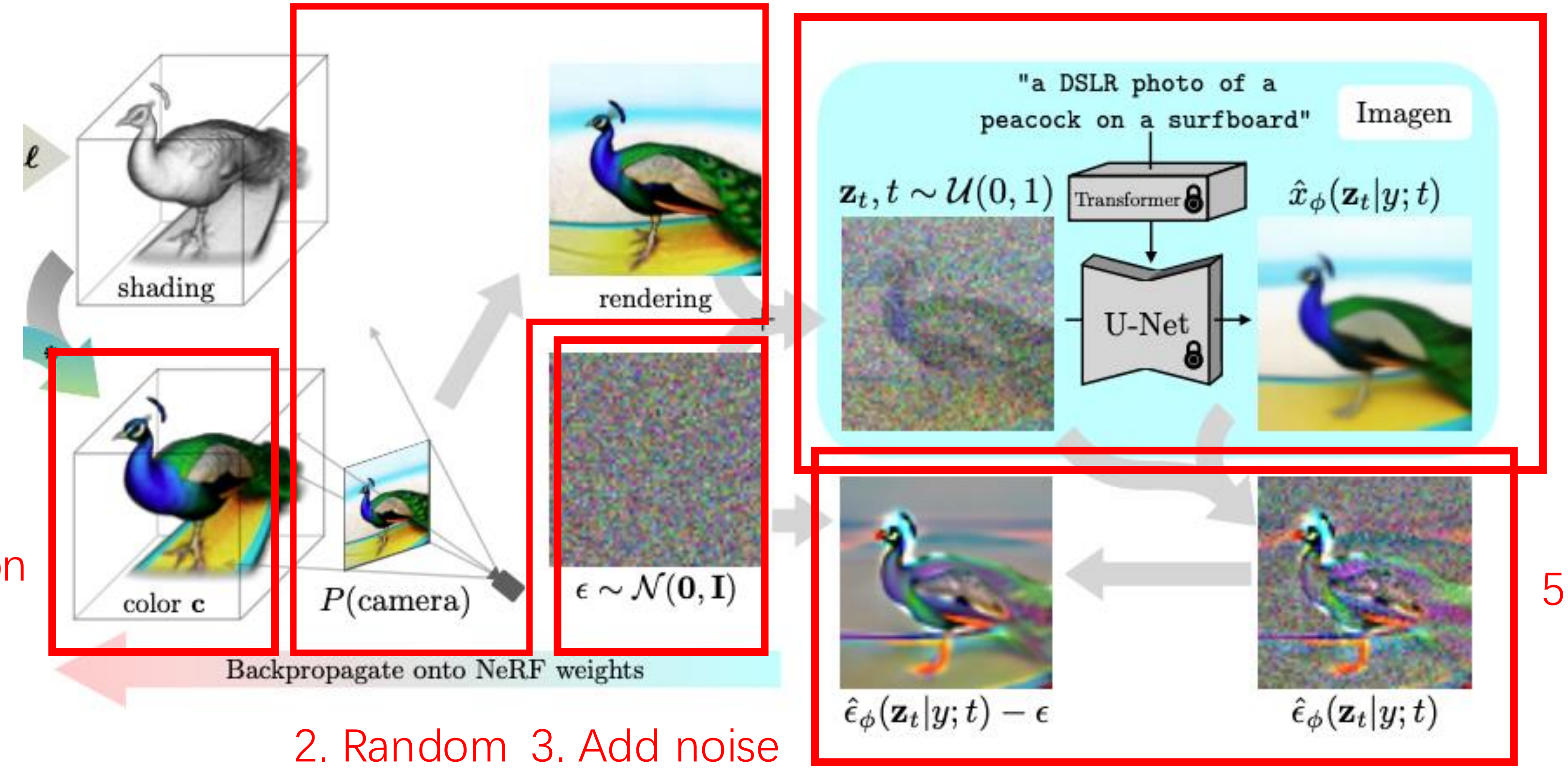
Walking the Schrödinger Bridge: A Direct Trajectory for Text-to-3D Generation

Ziying Li, Xuequan Lu, Xinkui Zhao*,
Guanjie Cheng, Shuiguang Deng, Jianwei Yin

NeurIPS 2025 Poster

Preliminaries: SDS (Score Distillation Sampling)

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x} = g(\theta)) \triangleq \mathbb{E}_{t,\epsilon}\left[w(t)\left(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon\right)\frac{\partial \mathbf{x}}{\partial \theta}\right] \quad (3)$$



1. 3D Representation

2. Random Render

3. Add noise

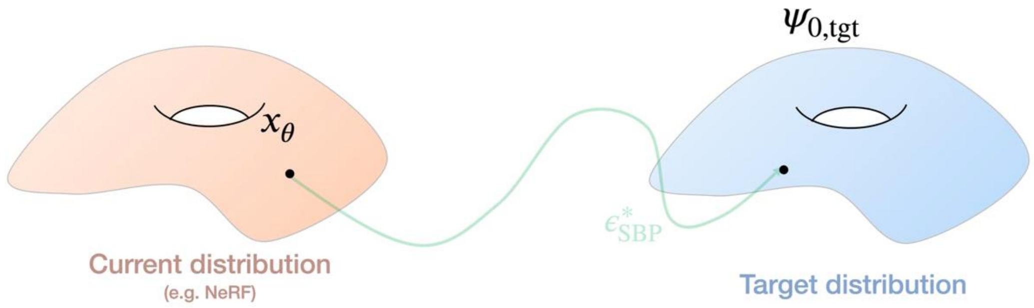
4. Noise Predicted by Pretrained T2I Model

5. Calculate Loss

From “DREAMFUSION: TEXT-TO-3D USING 2D DIFFUSION”

A Better Understanding of SDS:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x} = g(\theta)) \triangleq \mathbb{E}_{t,\epsilon}\left[w(t)\left(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon\right)\frac{\partial \mathbf{x}}{\partial \theta}\right] \quad (3)$$



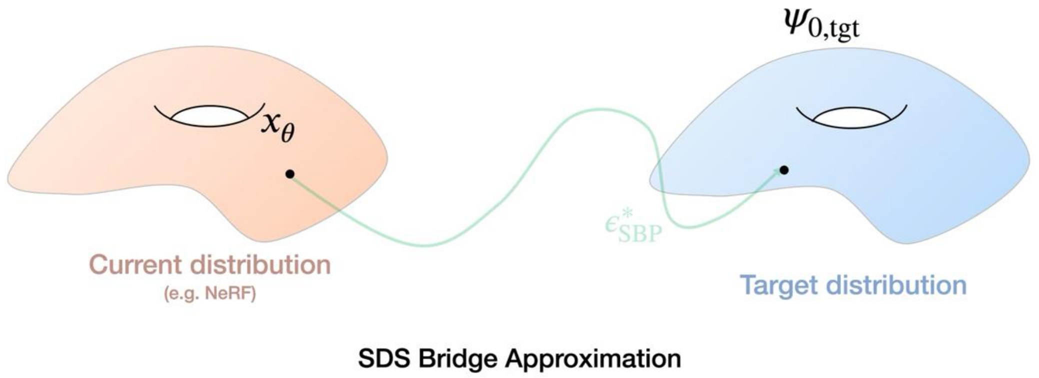
SDS Bridge Approximation

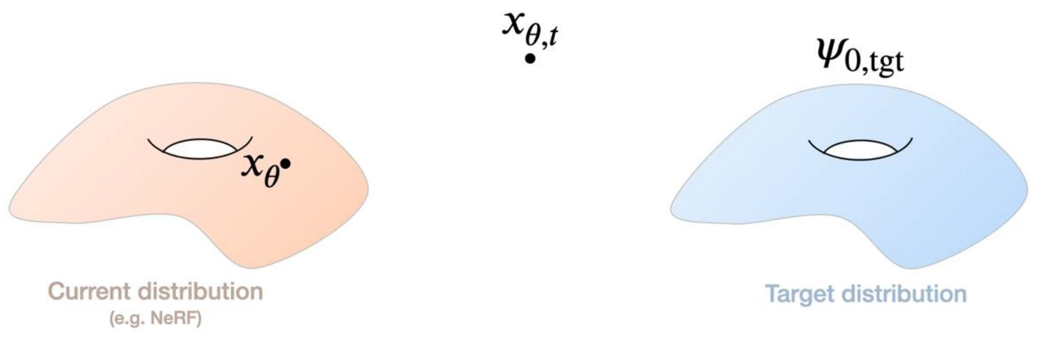
Ideally……

GIF from https://sds-bridge.github.io/

A Better Understanding of SDS:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x} = g(\theta)) \triangleq \mathbb{E}_{t,\epsilon}\left[w(t)\left(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon\right)\frac{\partial \mathbf{x}}{\partial \theta}\right] \quad (3)$$





Ideally……

In reality……

Predicted source distribution != source/current distribution!

GIF from https://sds-bridge.github.io/

Preliminaries: What Diffusion Model/SGM Does

Diffusion models/SGM:

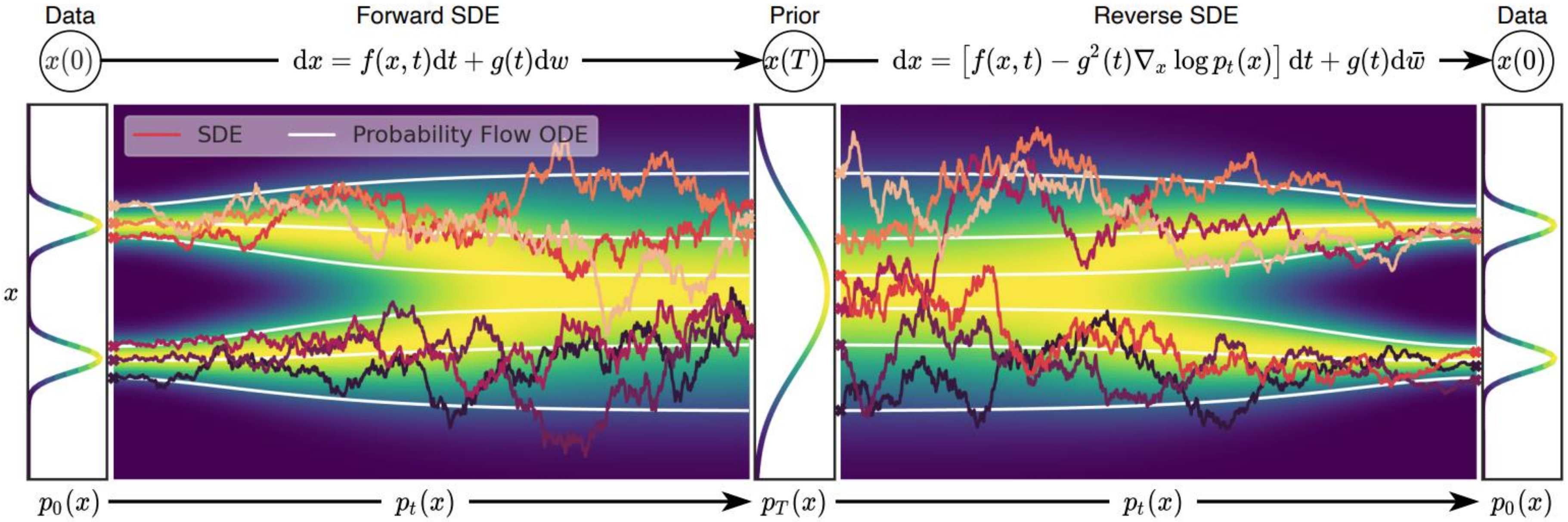
Diffusion Process — Reverse Process

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \longrightarrow p(\boldsymbol{x}_t|\boldsymbol{x}_0) \longrightarrow p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \longrightarrow p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$$

$$dX_t = f_t(X_t)dt + g_t dW_t \text{ (forward)}$$

$$dX_t = \left[f_t(X_t) - g_t^2 \nabla_{X_t} \log p(X_t, t)\right] dt + g_t d\bar{W}_t \text{ (backward)}$$

$$s_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x}(t)|\mathbf{x}(0))$$



Preliminaries: What Diffusion Model/SGM Does

Diffusion models/SGM:

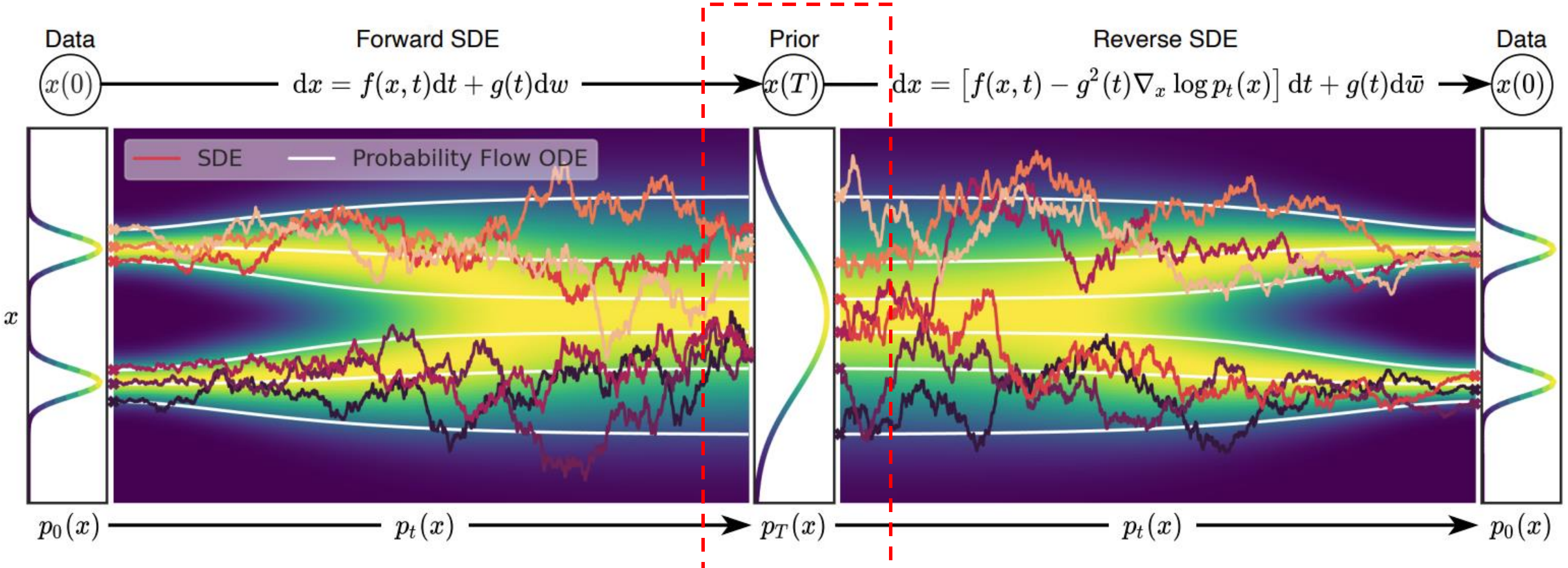
Diffusion Process | Reverse Process

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \longrightarrow p(\boldsymbol{x}_t|\boldsymbol{x}_0) \longrightarrow p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \longrightarrow p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$$

$$dX_t = f_t(X_t)dt + g_t dW_t \text{ (forward)}$$

$$dX_t = \left[f_t(X_t) - g_t^2 \nabla_{X_t} \log p(X_t, t)\right] dt + g_t d\bar{W}_t \text{ (backward)}$$

$$s_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x}(t)|\mathbf{x}(0))$$



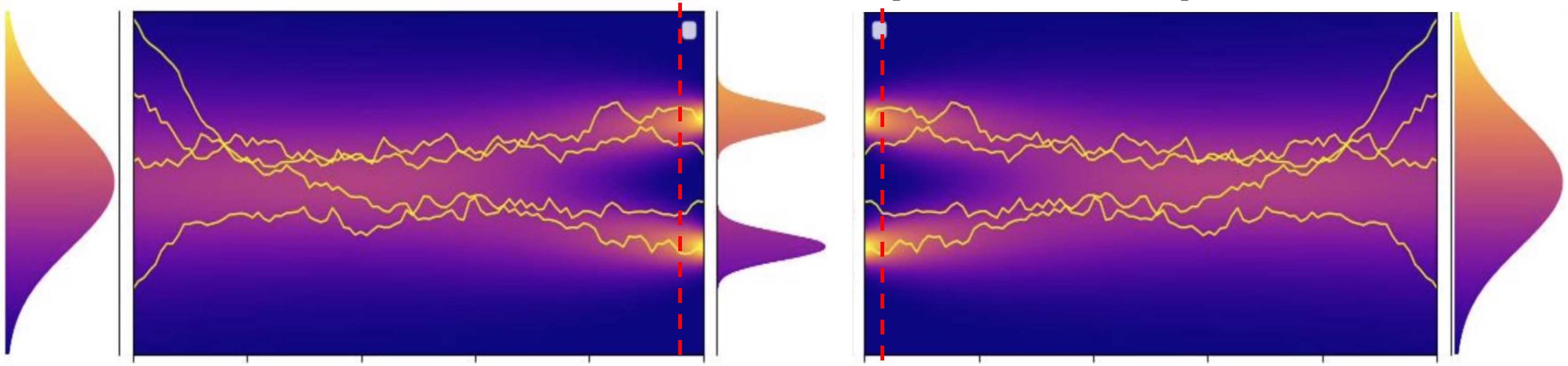
Preliminaries: Schrödinger Bridge (SB)

Non-linear SB System (what we care about)

$\nabla \hat{\Psi}(X_t, t)$

Reverse drift term

$$dX_t = \left[f_t(X_t) + \beta_t \nabla \Psi(X_t, t)\right] dt + \sqrt{\beta_t} dW_t \text{ (forward)} \quad dX_t = \left[f_t(X_t) - \beta_t \nabla \hat{\Psi}(X_t, t)\right] dt + \sqrt{\beta_t} d\bar{W}_t \text{ (backward)}$$



Distribution A Distribution B Distribution A

Preliminaries: Schrödinger Bridge (SB)

Non-linear SB System (what we care about)

$\nabla\hat{\Psi}(X_t, t)$

Reverse drift term

→ Nelson's duality $q(\cdot, t) = \Psi(\cdot, t)\hat{\Psi}(\cdot, t)$ →

$q(X_t|X_0, X_1) = \Psi(X_t, t|X_0)\widehat{\Psi}(X_t, t|X_1).$

Real distribution of SB

→ Fokker-Plank equations →

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

↓ Sampling

X_t

Training Data

← Fokker-Plank equations ←

$$\frac{\partial p(x,t)}{\partial t} = -\nabla \cdot (pf) + \frac{1}{2}\beta\Delta p$$

$$\frac{\partial \hat{\Psi}(x,t)}{\partial t} = -\nabla \cdot (\hat{\Psi}f) + \frac{1}{2}\beta\Delta\hat{\Psi}$$

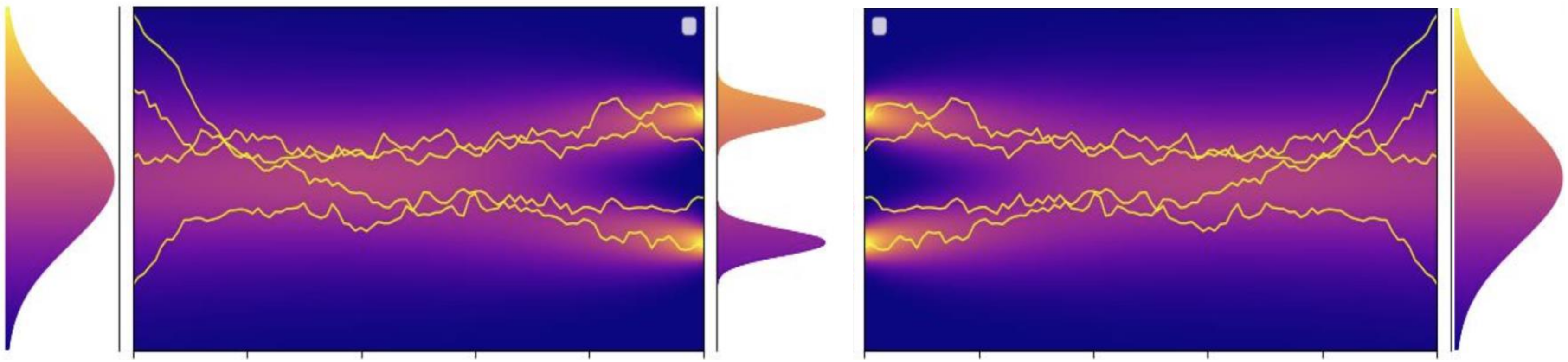
$\nabla \log \widehat{\Psi}(X_t, t|X_0) \equiv \nabla \log p(X_t, t|X_0).$

←

$\nabla \log p(X_t, t|X_0)$

Linear SDE System (used for training)

$dX_t = [f_t(X_t) + \beta_t \nabla\Psi(X_t, t)]\, dt + \sqrt{\beta_t} dW_t$ (forward) $\quad dX_t = \left[f_t(X_t) - \beta_t \nabla\hat{\Psi}(X_t, t)\right] dt + \sqrt{\beta_t} d\bar{W}_t$ (backward)



Distribution A Distribution B Distribution A

Preliminaries: Schrödinger Bridge (SB)

Non-linear SB System (what we care about)

$\nabla\hat{\Psi}(X_t, t)$ — Reverse drift term

→ Nelson's duality $q(\cdot, t) = \Psi(\cdot, t)\hat{\Psi}(\cdot, t)$ →

$q(X_t|X_0, X_1) = \Psi(X_t, t|X_0)\widehat{\Psi}(X_t, t|X_1).$ — Real distribution of SB

→ Fokker-Plank equations →

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

↓ Sampling

X_t — Training Data

Linear SDE System (used for training)

← Fokker-Plank equations ←

$$\frac{\partial p(x,t)}{\partial t} = -\nabla \cdot (pf) + \frac{1}{2}\beta\Delta p$$

$$\frac{\partial \hat{\Psi}(x,t)}{\partial t} = -\nabla \cdot (\hat{\Psi} f) + \frac{1}{2}\beta\Delta \hat{\Psi}$$

$\nabla \log \widehat{\Psi}(X_t, t|X_0) \equiv \nabla \log p(X_t, t|X_0).$

← $\nabla \log p(X_t, t|X_0)$

Schrödinger Bridge (SB) can directly use pre-trained diffusion models, and can model the trajectory between two random distributions!

Now, we can get two points:

Predicted source distribution != source/current distribution!

Schrödinger Bridge (SB) can directly use pre-trained diffusion models, and can model the trajectory between two random distributions!

Now, we can get two points:

Predicted source distribution != source/current distribution!

↓

Schrödinger Bridge (SB) solve the above problem of SDS

↑

Schrödinger Bridge (SB) can directly use pre-trained diffusion models, and can model the trajectory between two random distributions!

Score Distillation Sampling as a Special Case of Schrödinger Bridge

$$-\nabla_{X_t} \log \hat{\Psi}(X_t, t)$$

→ Nelson's duality: $\hat{\Psi}(X_t, t) = q(X_t, t)/\Psi(X_t, t)$

$$-\nabla_{X_t} \log \left(\frac{q(X_t, t)}{\Psi(X_t, t)} \right)$$

→ SDS case: $\Psi(x, t) \approx 1$, $P_B \sim \mathcal{N}(0, I)$

$$-\nabla_{X_t} \log \left(\frac{q(X_t, t)}{1} \right)$$

↓

$$-\nabla_{X_t} \log q(X_t, t)$$

← If $\Psi(x, t) \approx 1$, then its gradient $\nabla \log \Psi(X_t, t) \approx 0$.

SB's forward: $dX_t = [f_t(X_t) + \beta_t \nabla \Psi(X_t, t)]\, dt + \sqrt{\beta_t} dW_t$ (forward)

has non-linear drift term vanishes.
i.e., SGM forward = SB's forward

$$-\nabla_{X_t} \log p(X_t, t)$$

←

$$s_\theta(X_t, t)$$

SDS operates using this learned score!

$$\nabla_\theta \mathcal{L}_{\text{SDS}} \approx \mathbb{E}_{t,\epsilon} \left[w(t) \left(s_\theta(X_t, t) + \frac{\epsilon}{\sqrt{1 - \bar{\alpha}_t}} \right) \frac{\partial X_t}{\partial x_0} \frac{\partial x_0}{\partial \theta} \right]$$

TraCe: Overview

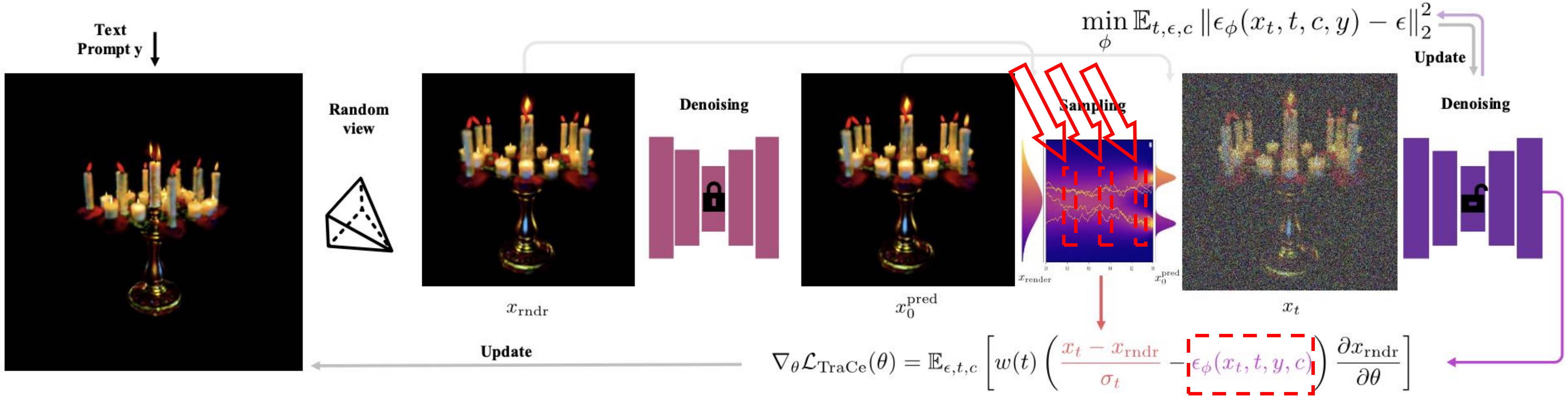


Figure 3: **Overview of Trajectory-Centric Distillation (TraCe).** Our approach optimizes 3D parameters θ by computing a distillation gradient with a LoRA-adapted 2D diffusion model, ϵ_ϕ. Given a text prompt y and camera parameters c, (1) The current 3D model is rendered in a random view to produce x_{rndr}. (2) An ideal target view x_0^{pred} is estimated from x_{rndr} using a pre-trained diffusion model $\epsilon_{\text{pretrain}}$ via one-step denoising. (3) An intermediate latent x_t is sampled from the analytic bridge posterior $q(x_t \mid x_0^{\text{pred}}, x_{\text{rndr}})$ at time t. (4) The LoRA model ϵ_ϕ predicts the noise for x_t, and the difference between this prediction and the target noise is computed. (5) This difference directs the calculation of the TraCe gradient $\nabla_\theta \mathcal{L}_{\text{TraCe}}$, and drives the update of LoRA parameters ϕ.

Results:

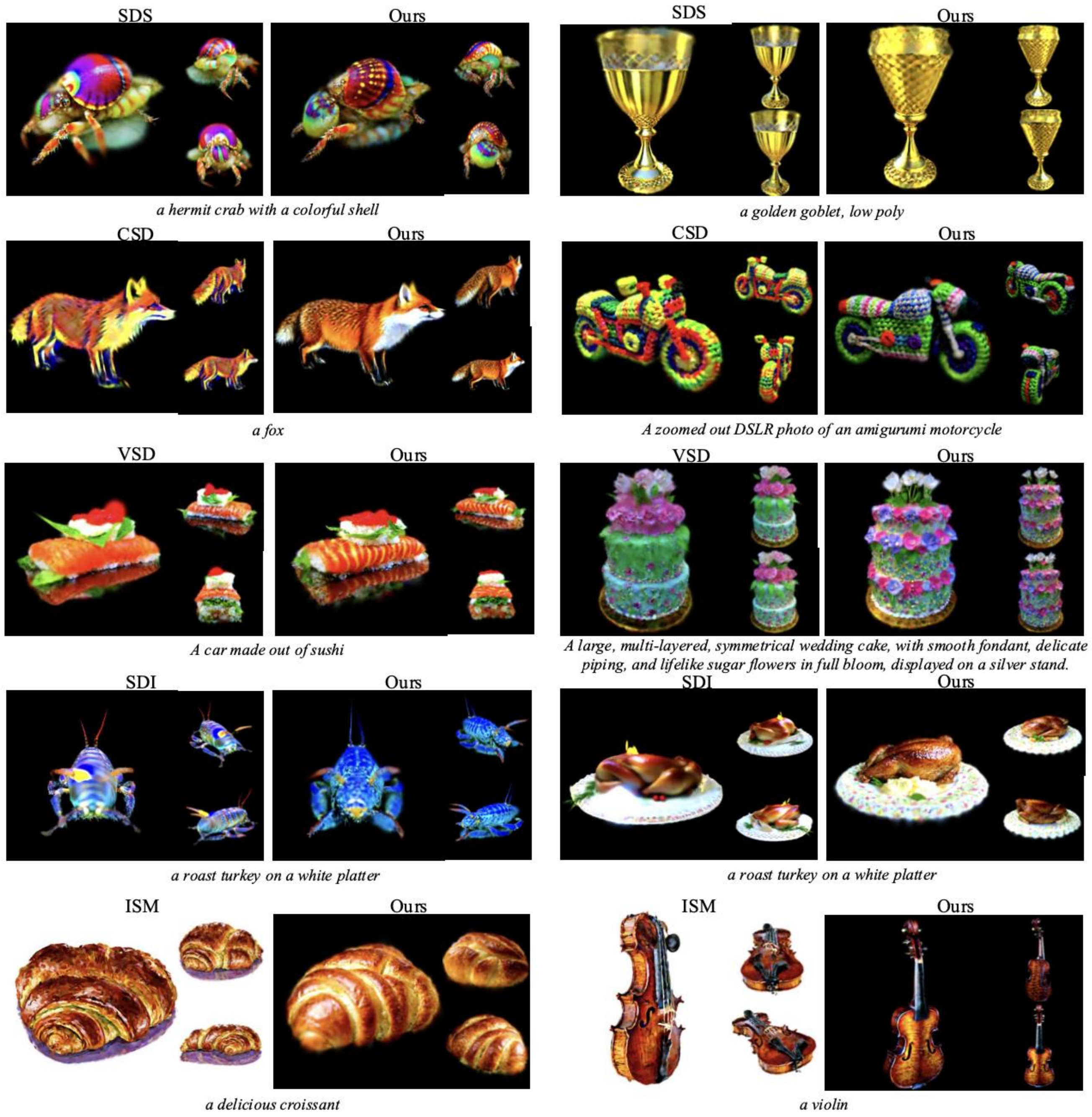


Figure 4: **Qualitative comparisons.** We present visual examples with the same text prompt.

Results:

Table 1: **Quantitative comparisons.** Comparison of different methods on CLIP Score, GPTEval3D Score, ImageReward Score, running time, and VRAM usage. We report mean and standard deviation across 83 prompts and 120 views.

Method	**CLIP Score (%) ↑**			**GPTEval3D (Overall)↑**	**ImageReward↑**	**Time**	**VRAM**
	ViT-L/14	ViT-B/16	ViT-B/32				
SDS [47]	68.6146±7.9134	27.7049±3.7004	27.5561±3.5893	1018.09	-0.4329±0.9125	10min	18147MiB
CSD [48]	68.0282±7.5093	27.0886±3.7342	26.5844±3.8703	983.04	-0.6715±0.7482	11min	19804MiB
VSD [45]	67.2697±8.5573	27.0749±3.9675	26.9722±3.9563	1007.49	-0.5330±0.8927	17min	26473MiB
ISM [23]	69.0093±10.2400	27.5460±3.6817	26.9822±3.5495	1012.37	-0.3904±0.9503	20min	10151MiB
SDI [29]	63.0409±11.7841	25.6487±5.2540	25.5421±5.0903	971.98	-0.8334±1.0391	10min	16011MiB
TraCe	**69.2609±7.8366**	**27.9334±3.7382**	**27.7049±3.8671**	**1028.03**	**-0.2855±0.8909**	14min	18741MiB

NEURAL INFORMATION PROCESSING SYSTEMS

浙江大学 ZHEJIANG UNIVERSITY 1897

SEEK WISDOM

THE UNIVERSITY OF WESTERN AUSTRALIA

Paper

Github