SFBD Flow: A Continuous-Optimization Framework for Training Diffusion Models with Noisy Samples

UNIVERSITY OF WATERLOO

Haoye Lu[1,2] Darren Lo[1] Yaoliang Yu[1,2]

[1]University of Waterloo ▪ [2]Vector Institute

VECTOR INSTITUTE

Paper Project

Motivation

Diffusion models memorize training data, reproducing copyrighted or private content at inference time:

(Daras, Dimakis & Daskalakis, ICML'24)

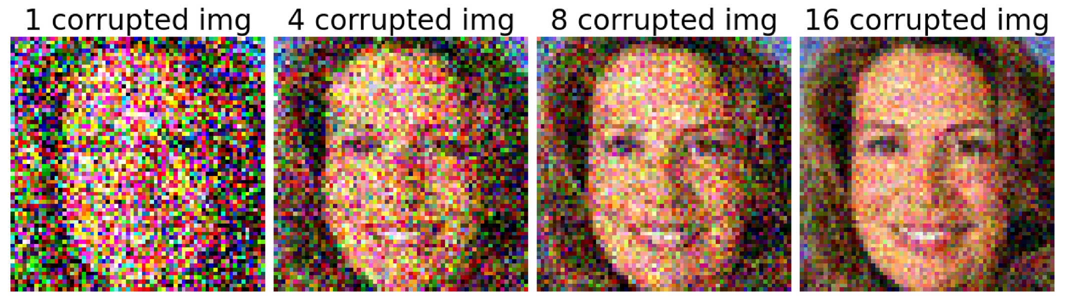
Large web-scraped datasets (LAION > 12M images) enable strong performance but introduce copyright and privacy risks.

One solution: never expose originals – train only on **noisy (corrupted) versions** of private data, combined with a small copyright-free clean set.

Problem Setting

Given:

- $\mathcal{E}_{\text{clean}}$: **tiny** copyright-free set ($\leq 1\%$ of full data)
- $\mathcal{E}_{\text{noisy}} = \{\mathbf{x}^{(i)} + \boldsymbol{\epsilon}^{(i)}\}$: **large** noisy private data, $\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0}, \sigma_\tau^2 \mathbf{I})$, known σ_τ
- Each sample is corrupted **once** – individual images are not recoverable



Goal: Learn p_{data} from $\mathcal{E}_{\text{noisy}}$ and $\mathcal{E}_{\text{clean}}$.

Key learnability insight: $p_{\text{noisy}} = p_{\text{data}} * \mathcal{N}(0, \sigma_\tau^2)$, so p_{data} is recoverable by deconvolution:

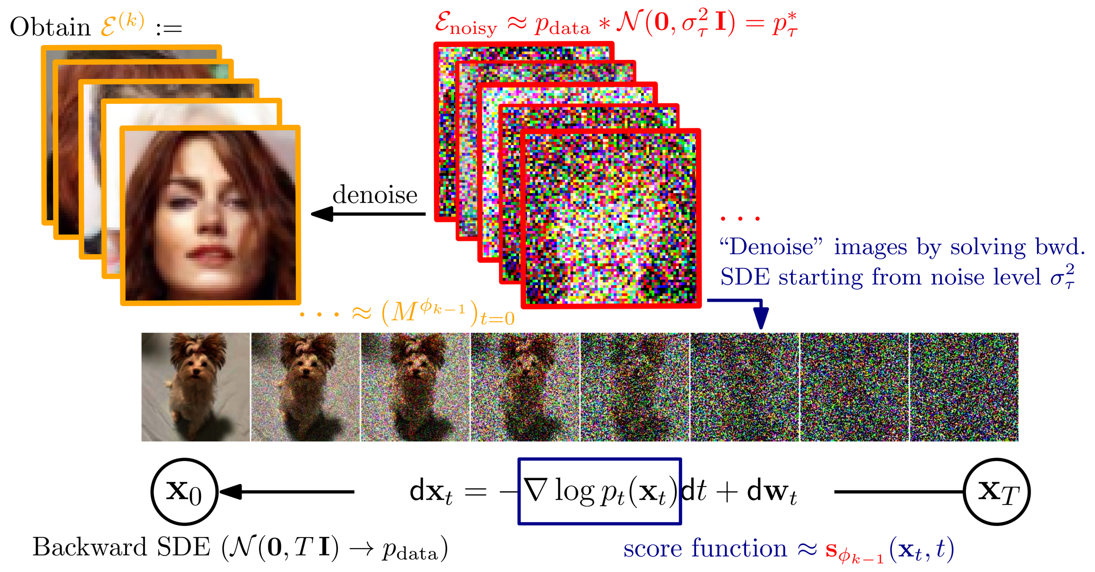
$$\begin{array}{ccl} p_{\text{data}} & \underset{\text{deconvolution}}{\overset{\text{convolution}}{\rightleftarrows}} & p_{\text{noisy}} = p_{\text{data}} * \mathcal{N}(\mathbf{0}, \sigma_\tau^2 \mathbf{I}) \\ \wr & & \wr \qquad (f * g)(\omega) := \int_{\mathbb{R}^d} f(\mathbf{x})\, g(\omega - \mathbf{x}) \mathrm{d}\mathbf{x} \\ \mathbf{x}^{(k)} & \underset{\text{irreversible}}{\overset{\text{add } \epsilon^{(k)} \sim \mathcal{N}(\mathbf{0}, \sigma_\tau^2 \mathbf{I})}{\rightleftarrows}} & \mathbf{x}^{(k)} + \epsilon^{(k)} \end{array}$$

Key Contributions

1. **Alternating-projection framework:** SFBD reinterpreted as projecting between path-measure sets $\mathcal{M}$ (backward-SDE) and $\mathcal{D}$ (forward-SDE). Each hard D-Proj jump forces a full optimizer restart.
2. **γ-SFBD = functional gradient descent:** Smooth blend with $\gamma \in (0, 1]$ is *exactly* discretized gradient descent on the drift-matching loss with **step size** γ, monotonically decreasing $D_{\mathrm{KL}}(p_{\text{data}} \,\|\, p_0^{k,\gamma})$. SFBD Flow ($\gamma \to 0$) is the continuous ODE limit. Enables principled **adaptive** γ (analogous to RMSProp).
3. **Online SFBD:** Single network, EMA target cache, adaptive γ based on estimation quality – **no restarts, 37% faster**. Achieves FID 3.12 on CIFAR-10, **surpassing DDPM trained on clean data (FID 4.04)**.

Prior Work: SFBD (Lu, Wu & Yu, ICML'25)

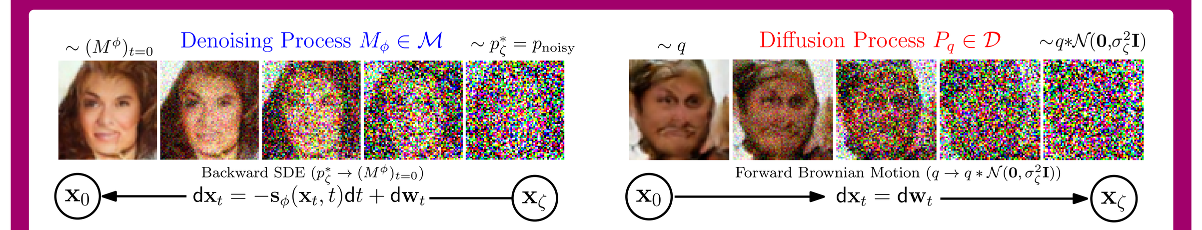
Solve deconvolution by alternating: for $k = 0, 1, \ldots, K$



1. **Train (M-Proj):** Pretrain/finetune diffusion model $\mathbf{s}_{\phi_k}$ on latest denoised samples
2. **Denoise (D-Proj):** Denoise noisy samples using $\mathbf{s}_{\phi_k}$ via backward SDE
3. Update denoised set and repeat

Limitation: K separate network trainings with manual optimizer resets. CIFAR-10 cost: **96 hours**.

SFBD as Alternating Projections



$\mathcal{M}$ = backward-SDE path measures; $\mathcal{D}$ = forward-SDE path measures. The **unique target** $P^* \in \mathcal{M} \cap \mathcal{D}$ has $p_0^* = p_{\text{data}}$.

$$\text{(M-Proj)}\ M^k = \operatorname*{argmin}_{M \in \mathcal{M}} D_{\mathrm{KL}}(P^k \,\|\, M)$$

$$\text{(D-Proj)}\ P^{k+1} = \operatorname*{argmin}_{P \in \mathcal{D}} D_{\mathrm{KL}}(M^k \,\|\, P)$$

M-Proj = score matching (training);
D-Proj = denoising via backward SDE.

Issue: D-Proj fully replaces the target distribution, forcing optimizer reset each round.

γ-SFBD & SFBD Flow: Algorithm & Intuition

γ-D-Proj: Blend instead of replace –

$$p_0^{k+1,\gamma} = (1 - \gamma)\, p_0^{k,\gamma} + \gamma\, m_0^k, \quad \gamma \in (0, 1]$$

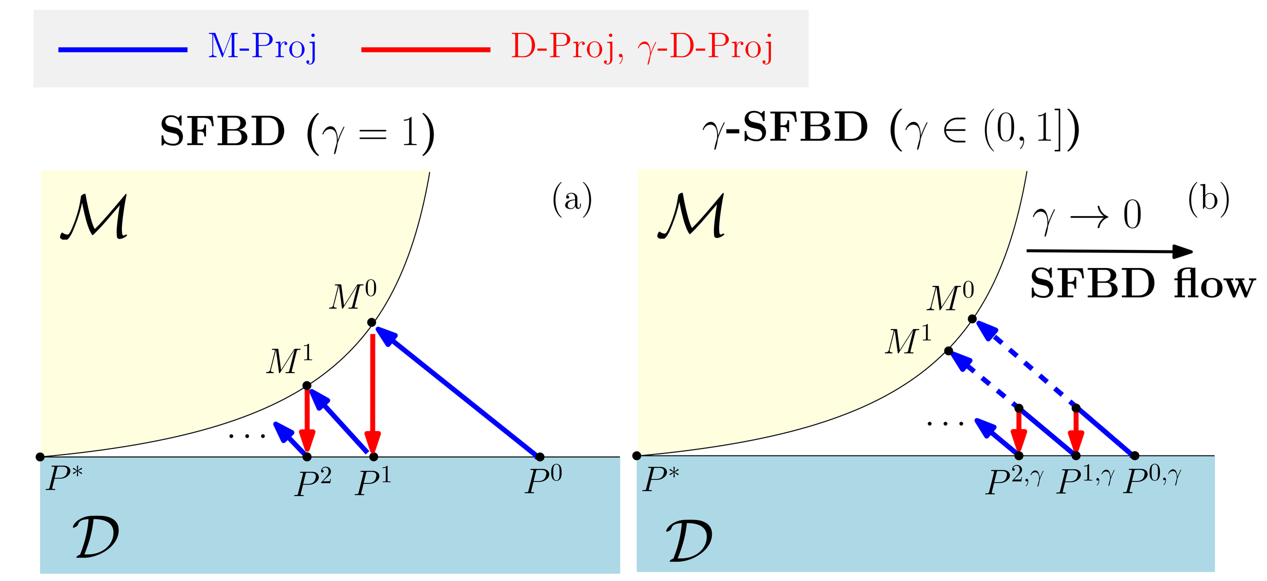
$p_0^{k,\gamma}$: current clean-data estimate; m_0^k: denoised output (D-Proj). $\gamma < 1$: **partial update** – only fraction γ of the estimate refreshed per step.

- $\gamma = 1$: full replacement $\Rightarrow$ recovers SFBD
- $\gamma \to 0$: **SFBD Flow** – continuous ODE:

$$\frac{\mathrm{d}}{\mathrm{d}\kappa} p_0^\kappa = m_0(\mathbf{s}^\kappa) - p_0^\kappa.$$

Intuition: Hard D-Proj ($\gamma = 1$) fully replaces the training target each round – a drastic shift that **invalidates Adam's momentum/variance** estimates, forcing a full optimizer reset. With small γ, the target drifts gradually; Adam stays valid and a **single network** tracks it continuously – **no restarts**.

γ-SFBD & SFBD Flow: Formal Theory



Formal view: γ-SFBD is discretized **functional gradient descent** on

$$\mathcal{L}(\mathbf{s}, M_0(\mathbf{s})) := \frac{1}{2} \int_0^\tau \mathbb{E}_{D(m_0)} \left\| \frac{\mathbf{x}_0 - \mathbf{x}_t}{t} - \mathbf{s}_t(\mathbf{x}_t) \right\|^2 \mathrm{d}t$$

(m_0 = denoised samples under $\mathbf{s}$) with step size γ. For each step:

$$D_{\mathrm{KL}}(p_{\text{data}} \,\|\, p_0^{k+1,\gamma}) \leq D_{\mathrm{KL}}(p_{\text{data}} \,\|\, p_0^{k,\gamma})$$

As $\gamma \to 0$: **SFBD Flow** – gradient flow (ODE shown in Col. 2).

Convergence bound (all $\gamma \in (0, 1]$, K steps):

$$\min_{k \leq K} \left| \Phi_{p_{\text{data}}}(\mathbf{u}) - \Phi_{p_0^{k,\gamma}}(\mathbf{u}) \right| \leq \exp\!\left(\frac{\tau}{2} \|\mathbf{u}\|^2\right) \left(\frac{2M}{\gamma K}\right)^{\frac{1}{2}}$$

$\Phi_p(\mathbf{u}) = \mathbb{E}_p[e^{i\mathbf{u}^\top \mathbf{x}}]$, $M = D_{\mathrm{KL}}(p_{\text{data}} \,\|\, p_{\mathcal{E}_{\text{clean}}})$.

Online SFBD: Practical Algorithm

Key insight: Small $\gamma \Rightarrow p_0^{k,\gamma}$ evolves slowly $\Rightarrow$ a **single network** $\mathbf{s}_\phi$ can track it continuously with a few gradient steps per update – **no fine-tuning loop**.

Each iteration:

1. **γ-D-Proj:** Denoise γ-fraction of cache $\mathcal{E}$ using EMA model (can be run **asynchronously on a separate GPU** while training continues)
2. **M-Proj:** Take m gradient steps minimizing score-matching loss on updated $p_0^{k,\gamma}$

Adaptive step size γ_k: The convergence bound shows larger γ is better but only when the network tracks the target. We use KID($\mathcal{E}_{\text{clean}}, \mathcal{E}_{\text{gen}}$) as quality signal δ_k with **RMSProp-style adaptation** and **provable efficiency guarantees**:

$$\gamma_k = \min\!\left(\frac{\gamma_0}{\sqrt{v_k}}, \Gamma\right), \quad v_k = \rho v_{k-1} + (1 - \rho)\delta_k^2$$

Large δ_k (network lags) $\Rightarrow \gamma_k \downarrow$; small δ_k (tracking well) $\Rightarrow \gamma_k \uparrow$.
In our implementation, δ_k is measured by KID (Kernel Inception Distance) – an **unbiased** alternative to FID that requires far fewer samples.

Pretraining Strategy

Ambient Score Matching (ASM). Score estimation uses two regimes split at corruption level τ:

- $t < \tau$: Train on small clean set $\mathcal{E}_{\text{clean}}$ directly.
- $t > \tau$: Noisy data $\tilde{\mathbf{x}} \sim p_{\text{noisy}}$ is already at noise level σ_τ; inject extra noise to reach $\sigma_t > \sigma_\tau$; train score on these doubly-noised samples.

OSFBD-AMB = Online SFBD + ASM pretraining.
OSFBD-VAN = Online SFBD + pretrained *only* on $\mathcal{E}_{\text{clean}}$.

Performance Comparison (FID ↓)

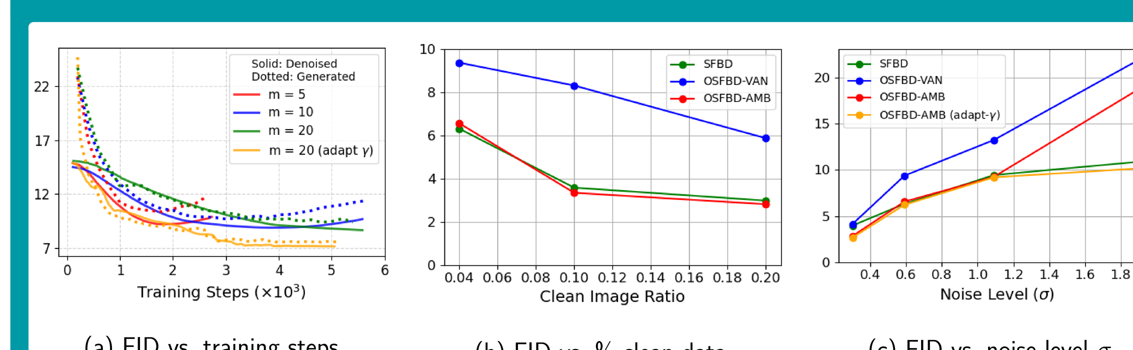
Method	Pretrain	CIFAR-10	CelebA
Reference: trained on clean data only			
DDPM	—	4.04	**3.26**
DDIM	—	4.16	6.53
EDM	—	**1.97**	—
Noisy data ($\sigma = 0.2$) + 50 clean imgs pretrain			
TweedieDiff+	✓	8.05	6.81
SFBD	✓	13.53	6.49
OSFBD-AMB (fixed γ)	✓	3.22	3.23
OSFBD-AMB (adapt. γ)	✓	**3.12**	**3.19**

Harder settings (higher noise / fewer clean samples)

Method	CIFAR-10		CelebA	
	$\sigma = 0.59$, 4%	$\sigma = 0.2$, 10%	$\sigma = 1.38$, 1500	$\sigma = 1.38$, 50
TweedieDiff+	6.75	2.81	6.81	35.65
SFBD	6.31	2.58	5.91	23.63
OSFBD-AMB (fixed γ)	6.56	2.73	5.72	27.09
OSFBD-AMB (adapt. γ)	**6.21**	**2.49**	**5.40**	**20.21**

Training time: OSFBD requires no manual resets – **60h vs 96h** (CIFAR-10) and **108h vs 156h** (CelebA), with **37%** and **31% speedup** over SFBD.

Ablation Studies



(a) FID vs. training steps (b) FID vs. % clean data (c) FID vs. noise level σ

(a) Solid = generated FID; dotted = denoised FID (target quality). Adaptive γ avoids divergence at small fixed m and converges faster. **(b–c)** OSFBD-AMB (ambient pretrain) consistently outperforms OSFBD-VAN (clean-only pretrain) across all settings.

Generated Samples (OSFBD adaptive γ)