Coupled Data and Measurement Space Dynamics for Enhanced Diffusion Posterior Sampling

Shayan Mohajer Hamidi[1], Ben Liang[2], En-Hui Yang[3]

[1]Stanford University, [2]University of Toronto,

[3] University of Waterloo

Linear Inverse Problems (LIP)

- We are given measurement $\boldsymbol{y} \in \mathbb{R}^m$
- The goal is to reconstruct unknown signal $\boldsymbol{x}_0 \in \mathbb{R}^d$

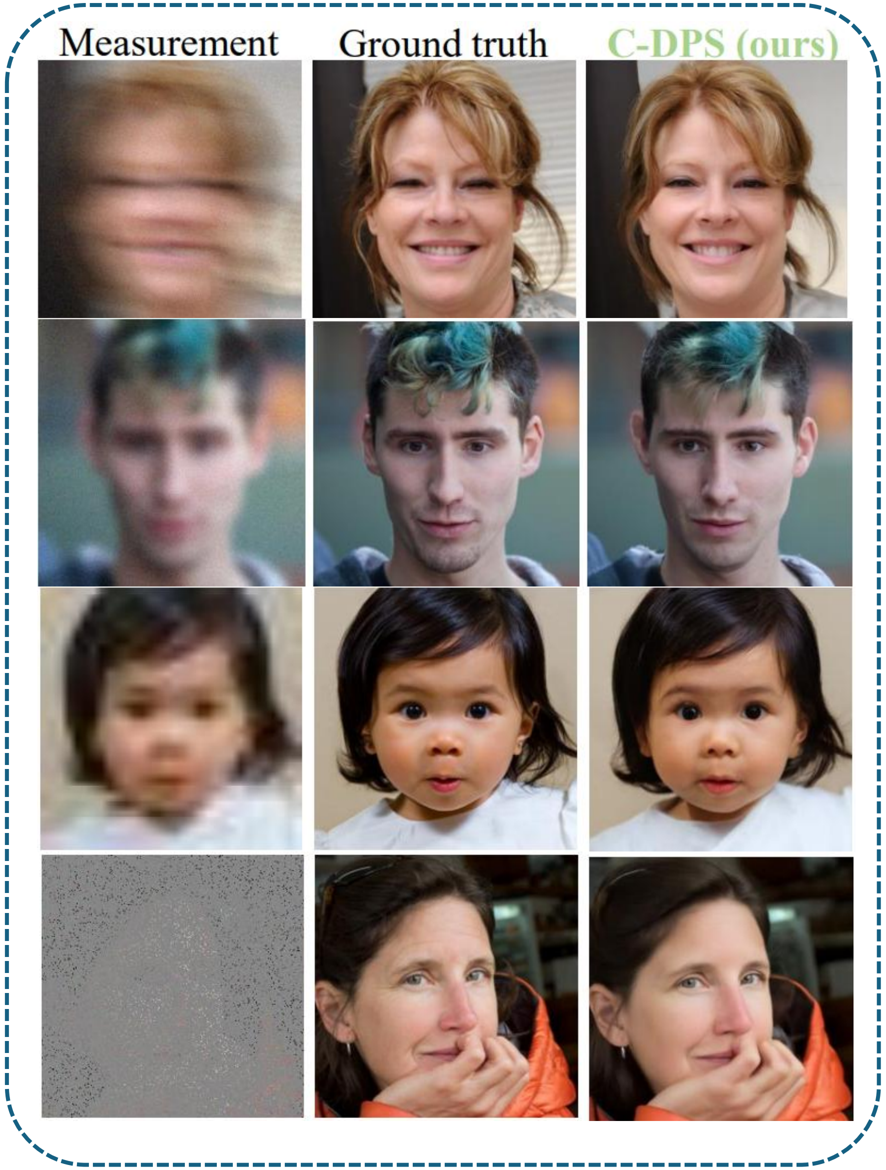
, where $\mathbf{A} \in \mathbb{R}^{m \times d}$ is a known measurement.

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x}_0 + \mathbf{n}$$

, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{n}})$

- This leads to a Gaussian likelihood

$$p(\boldsymbol{y}|\boldsymbol{x}_0) = \mathcal{N}(\mathbf{A}\boldsymbol{x}_0, \boldsymbol{\Sigma}_{\boldsymbol{n}})$$



Background (Diffusion Models)

Forward process is described by Ito stochastic differential equation (SDE):

$$d\boldsymbol{x} = -\frac{\beta(t)}{2}\boldsymbol{x}\,dt + \sqrt{\beta(t)}\,d\boldsymbol{w},$$

noise schedule

Wiener process

$\boldsymbol{x}(0) \sim p_{\text{data}}$ ⇒ $\boldsymbol{x}(T) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Backward process is the reverse of above SDE:

$$d\boldsymbol{x} = \left[-\frac{\beta(t)}{2}\boldsymbol{x} - \beta(t)\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t)\right]dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}},$$

This term is called score function and is learned by a diffusion
$\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x}_t) \simeq \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t)$

Reverse Wiener process

Conventional Methods for Solving LIP Using Diffusion Models

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x}_0 + \mathbf{n}$$

Noisy measurement

Known measurement $\boldsymbol{A} \in \mathbb{R}^{m \times d}$

Unknown signal

$\sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$

This is an ill-posed problem with many solutions for $\boldsymbol{x}_0$. Thus, we need some kind of prior to obtain meaningful $\boldsymbol{x}_0$.

In Bayesian framework:

$$p(\boldsymbol{x}_0|\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x}_0)p(\boldsymbol{x}_0)/p(\boldsymbol{y})$$

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

Unconditional Reverse Path

$$d\boldsymbol{x} = \left[-\frac{\beta(t)}{2}\boldsymbol{x} - \beta(t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right] dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}},$$

Conditional Reverse Path

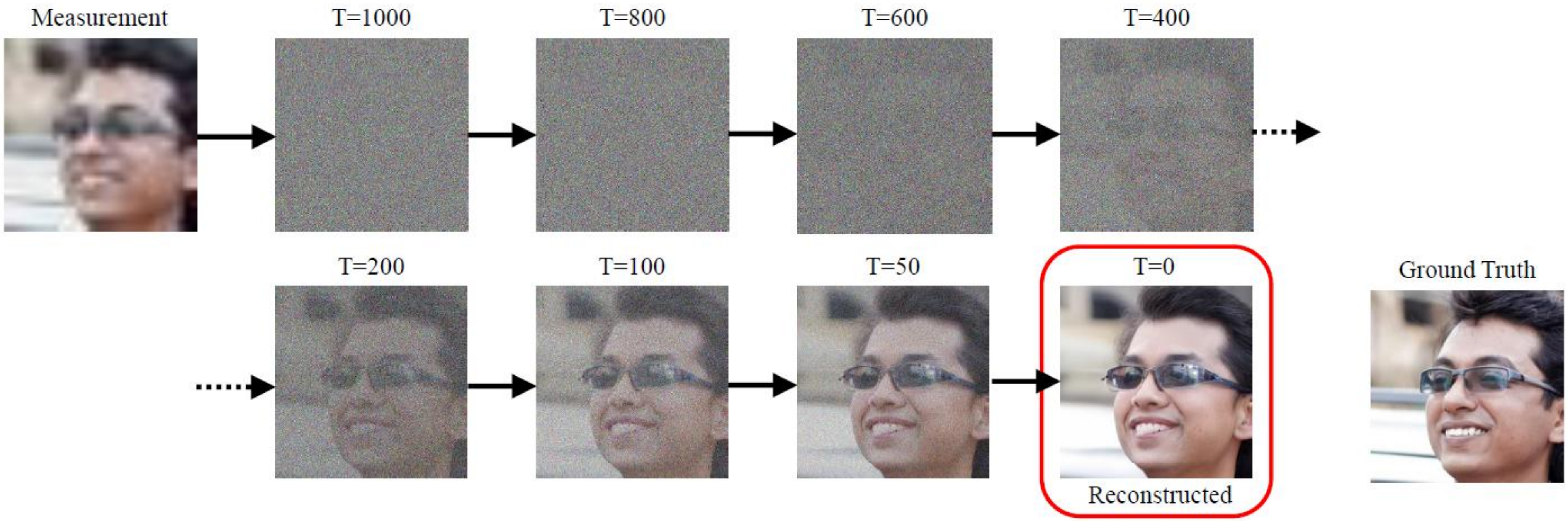
$$d\boldsymbol{x} = \left[-\frac{\beta(t)}{2}\boldsymbol{x} - \beta(t)\left(\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)\right)\right] dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}}$$

Replaced with: $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{y}) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$

Conventional Methods for Solving LIP Using Diffusion Models

Conditional Reverse Path

$$d\boldsymbol{x} = \left[-\frac{\beta(t)}{2}\boldsymbol{x} - \beta(t)(\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)) \right] dt + \sqrt{\beta(t)} d\bar{\boldsymbol{w}}$$



Conventional Methods for Solving LIP Using Diffusion Models

$$d\boldsymbol{x} = \left[-\frac{\beta(t)}{2}\boldsymbol{x} - \beta(t)\left(\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)\right)\right] dt + \sqrt{\beta(t)}d\bar{\boldsymbol{w}}$$

This term is estimated by a pre-trained diffusion model

This term is intractable and should be **estimated**.

$$p(\boldsymbol{y}|\boldsymbol{x}_t) = \int p(\boldsymbol{y}|\boldsymbol{x}_0, \boldsymbol{x}_t)p(\boldsymbol{x}_0|\boldsymbol{x}_t)d\boldsymbol{x}_0 = \int p(\boldsymbol{y}|\boldsymbol{x}_0)p(\boldsymbol{x}_0|\boldsymbol{x}_t)d\boldsymbol{x}_0$$

This term is known $p(\boldsymbol{y}|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{y}|\mathbf{A}\boldsymbol{x}_0, \sigma^2\mathbf{I})$

This term should be estimated

Prior art assumed that $p_t(\boldsymbol{x}_0|\boldsymbol{x}_t) \sim \mathcal{N}(\tilde{\boldsymbol{x}}_0 = E[x_0|x_t], r_t^2\boldsymbol{I})$.

$$\tilde{x}_0 = \frac{1}{\sqrt{\bar{\alpha}(t)}}(\boldsymbol{x}_t + (1-\bar{\alpha}(t))\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t))$$

Heuristically selected the covariance r_t^2

Our Approach (C-DPS)

How to eliminate the need for this approximation $p_t(\boldsymbol{x_0}|\boldsymbol{x}_t) \sim \mathcal{N}(\tilde{\boldsymbol{x}}_0 = E[x_0|x_t], \; r_t^2 \boldsymbol{I})$?

- Can we directly find a closed form formula for $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}_{t-1})$
- ***Our solution***: We define forward stochastic process in the measurement space $\{\boldsymbol{y}_t\}_{t=0}^T$
- By coupling the data-space process $\{\boldsymbol{x}_t\}_{t=0}^T$ and $\{\boldsymbol{y}_t\}_{t=0}^T$ we find a closed form formula for $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}_{t-1})$

Markov Chain in the Measurement Space

NEURAL INFORMATION PROCESSING SYSTEMS

$$\boldsymbol{y}_0 = \mathbf{A}\boldsymbol{x}_0 + \mathbf{n},$$
$$\boldsymbol{y}_t = \sqrt{1-\beta_t}\,\boldsymbol{y}_{t-1} + \sqrt{\beta_t}\,\boldsymbol{z}_t$$

$$\boldsymbol{y}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{y}_0 + \sqrt{1-\bar{\alpha}_t}\,\zeta$$

$\bar{\alpha}_t = \prod_{j=1}^{t} \alpha_j$

$\sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$

It is easy to check that the distribution of $\boldsymbol{y_t}$ is a Gaussian whose mean and covariance at step t are given by:

$$\mu_{\boldsymbol{y},t} = \sqrt{\bar{\alpha}_t}\,\mathbf{A}\,\boldsymbol{x}_0,$$
$$\boldsymbol{\Sigma}_{\boldsymbol{y},t} = \bar{\alpha}_t\,\boldsymbol{\Sigma}_{\mathbf{n}} + \left(1-\bar{\alpha}_t\right)\mathbf{I}$$

Generating $\{\boldsymbol{x}_t\}$ Consistent with $\{\boldsymbol{y}_t\}$

- Initialization: $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- Backward Recursion: $p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}_{t-1}) \propto p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})\, p(\boldsymbol{y}_{t-1} \mid \boldsymbol{x}_{t-1})$

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\mathbf{I})$$

$$\boldsymbol{y}_0 = \mathbf{A}\,\mathrm{x}_0 + \mathbf{n}$$

$$\boldsymbol{y}_t = \sqrt{\bar{\alpha}_t}\;\boldsymbol{y}_0 + \sqrt{1-\bar{\alpha}_t}\;\boldsymbol{\zeta}$$

$$\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t) = \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$$

$$\begin{aligned}\boldsymbol{y}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\left(\mathbf{A}\left[\frac{\boldsymbol{x}_{t-1} + (1-\bar{\alpha}_{t-1})\boldsymbol{s}_\theta(\boldsymbol{x}_{t-1}, t-1)}{\sqrt{\bar{\alpha}_{t-1}}}\right] + \mathbf{n}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\,\boldsymbol{\zeta} \\ &= \mathbf{A}\boldsymbol{x}_{t-1} + \mathbf{A}\big((1-\bar{\alpha}_{t-1})\boldsymbol{s}_\theta(\boldsymbol{x}_{t-1}, t-1)\big) + \sqrt{\bar{\alpha}_{t-1}}\mathbf{n} + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\zeta}.\end{aligned}$$

$$p(\boldsymbol{y}_{t-1}|\boldsymbol{x}_{t-1}) \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{y}|\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}\right),$$

where $\boldsymbol{\mu}_{\boldsymbol{y}|\boldsymbol{x}} = \mathbf{A}\boldsymbol{x}_{t-1} + (1-\bar{\alpha}_{t-1})\mathbf{A}\boldsymbol{s}_\theta(\boldsymbol{x}_{t-1}, t-1),$

and $\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}} = \bar{\alpha}_{t-1}\boldsymbol{\Sigma}_{\mathbf{n}} + (1-\bar{\alpha}_{t-1})\mathbf{I}.$

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

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}_{t-1}) \propto \exp\Big[-\frac{1}{2\beta_t}\big\|\boldsymbol{x}_t - \sqrt{1-\beta_t}\,\boldsymbol{x}_{t-1}\big\|^2 - \frac{1}{2}\big(\boldsymbol{y}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{y}|\boldsymbol{x}}\big)^\top \boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}^{-1}\big(\boldsymbol{y}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{y}|\boldsymbol{x}}\big)\Big]$$

The first term is quadratic in $\boldsymbol{x}_{t-1}$. However, the second term involves the score network $s_\theta(\boldsymbol{x}_{t-1}, t-1)$ making the conditional mean $\mu_{\boldsymbol{y}|\boldsymbol{x}}$ a nonlinear function of $\boldsymbol{x}_{t-1}$.

$$\boldsymbol{s}_\theta(\boldsymbol{x}_{t-1}, t-1) \quad \longrightarrow \quad \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$$

Using this approximation, $\mu_{\boldsymbol{y}|\boldsymbol{x}}$ becomes affine in $\boldsymbol{x}_{t-1}$:

$$\boldsymbol{\mu}_{\boldsymbol{y}|\boldsymbol{x}} = \mathbf{A}\boldsymbol{x}_{t-1} + \underbrace{(1-\bar{\alpha}_{t-1})\mathbf{A}\,\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)}_{\triangleq\, \boldsymbol{b}_{t-1}}$$

In this case $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}_{t-1})$ ***becomes Gaussian*** with:

$$\boldsymbol{\Sigma}_{\text{post}}^{-1} = \frac{1-\beta_t}{\beta_t}\mathbf{I} + \mathbf{A}^\top \boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}^{-1}\mathbf{A},$$

$$\boldsymbol{\mu}_{\text{post}} = \boldsymbol{\Sigma}_{\text{post}}\Big[\frac{\sqrt{1-\beta_t}}{\beta_t}\boldsymbol{x}_t + \mathbf{A}^\top \boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}^{-1}(\boldsymbol{y}_{t-1} - \boldsymbol{b}_{t-1})\Big]$$

Efficient Sampling from $p(\boldsymbol{x}_{t-1} | \boldsymbol{x}_t, \boldsymbol{y}_{t-1})$

NEURAL INFORMATION PROCESSING SYSTEMS

- Define the posterior precision operator

$$\Lambda_t = \boldsymbol{\Sigma}_{\text{post}}^{-1} = c_t \boldsymbol{I} + \boldsymbol{A}^\top \boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}^{-1} \boldsymbol{A}, \qquad c_t = \frac{1-\beta_t}{\beta_t}$$

- Direct Cholesky on dense $\Sigma_{\mathbf{y}|\boldsymbol{x}}$ is $O(d^3)$ and impractical.
- We use two matrix-free conjugate gradient (CG) solves per reverse step:
- **Step 1:** mean solve

$$\Lambda_t \boldsymbol{\mu}_{\text{post}} = c_t \boldsymbol{x}_t + \boldsymbol{A}^\top \boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}^{-1} (\boldsymbol{y}_{t-1} - \boldsymbol{b}_{t-1}), \qquad \boldsymbol{b}_{t-1} = (1 - \bar{\alpha}_{t-1}) \boldsymbol{A} \hat{\boldsymbol{s}}$$

- **Step 2:** noise draw (PW-CG).

Algorithm 3 PW-CG (draw $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\text{post}})$ without Cholesky)

Input: precision operator Λ_t, matrix $\boldsymbol{A}$, action of $\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}^{-1}$ (or its square root), scalar c_t

1: Draw $\boldsymbol{\varepsilon}_1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$, $\boldsymbol{\varepsilon}_2 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_m)$, independent

2: (Prewhiten) define a whitening operator $\boldsymbol{W}$ with $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}^{-1}$, and set $\tilde{\boldsymbol{A}} = \boldsymbol{W}\boldsymbol{A}$

3: **Form** $\boldsymbol{z} \leftarrow \sqrt{c_t}\, \boldsymbol{\varepsilon}_1 + \tilde{\boldsymbol{A}}^\top \boldsymbol{\varepsilon}_2$

4: **Solve** $\boldsymbol{v} \leftarrow \text{CG-solve}(\Lambda_t, \boldsymbol{z})$ (that is, solve $\Lambda_t \boldsymbol{v} = \boldsymbol{z}$)

Output: $\boldsymbol{v}$ (then $\text{cov}(\boldsymbol{v}) = \Lambda_t^{-1} = \boldsymbol{\Sigma}_{\text{post}}$)

- **Step 3:** update $\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_{\text{post}} + \boldsymbol{v}$

Experiments

Table 1: Quantitative results on the 1k validation sets of FFHQ 256×256 and ImageNet 256×256. **Bold** and underline indicate the best and second-best results, respectively. Green and red denote performance improvements and degradations relative to the best baseline.

Pixel-Domain Methods																
Dataset	**Method**	**Inpaint (Random)**			**Inpaint (Box)**			**Deblur (Gaussian)**			**Deblur (Motion)**			**SR** (4×)		
		FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑	FID ↓	LPIPS ↓	SSIM ↑
FFHQ	DPS	21.19	0.212	0.851	33.12	0.168	0.873	44.05	0.257	0.811	39.92	0.242	0.859	39.35	0.214	0.852
	ΠGDM	21.27	0.221	0.840	34.79	0.179	0.860	40.21	0.242	0.825	33.24	0.221	0.887	34.98	0.202	0.854
	DDRM	69.71	0.587	0.319	42.93	0.204	0.869	74.92	0.332	0.767	–	–	–	62.15	0.294	0.835
	MCG	29.26	0.286	0.751	40.11	0.309	0.703	101.2	0.340	0.051	–	–	–	87.64	0.520	0.559
	ILVR	25.74	0.231	0.672	37.24	0.175	0.854	52.93	0.297	0.784	–	–	–	47.59	0.253	0.844
	ReSample	21.25	0.202	0.847	33.51	0.160	0.866	37.05	0.251	0.822	31.19	0.220	0.892	30.48	0.204	0.851
	PnP-ADMM	123.6	0.692	0.325	151.9	0.406	0.642	90.42	0.441	0.812	–	–	–	66.52	0.353	0.855
	Score-SDE	76.54	0.612	0.437	60.06	0.331	0.678	109.0	0.403	0.109	–	–	–	96.72	0.563	0.617
	ADMM-TV	181.5	0.463	0.784	68.94	0.322	0.814	186.7	0.507	0.801	–	–	–	110.6	0.428	0.803
	PnP-DM	21.15	0.208	0.858	32.21	0.155	0.877	41.92	0.251	0.816	37.21	0.233	0.871	36.21	0.210	0.859
	DAPS	20.77	0.201	0.869	29.44	0.144	0.882	35.84	0.242	0.830	30.26	0.215	0.911	30.15	0.202	0.854
	DMPlug	**20.12**	0.197	0.877	27.12	0.140	**0.888**	32.44	**0.230**	0.830	27.55	**0.210**	**0.925**	28.55	0.199	**0.862**
	C-DPS	20.14	**0.195**	**0.881**	**26.33**	**0.132**	0.871	**32.24**	0.238	**0.832**	**27.29**	0.217	0.921	**28.41**	**0.196**	0.855
ImageNet	DPS	35.87	0.303	0.739	38.82	0.262	0.794	62.72	0.444	0.706	56.08	0.389	0.634	50.66	0.337	0.781
	ΠGDM	41.82	0.356	0.705	42.26	0.284	0.752	59.79	0.425	**0.717**	54.18	0.373	**0.675**	54.26	0.352	0.765
	DDRM	114.9	0.665	0.403	45.95	0.245	**0.814**	63.02	0.427	0.705	–	–	–	59.57	0.339	0.790
	MCG	39.19	0.414	0.546	39.74	0.330	0.633	95.04	0.550	0.441	–	–	–	144.5	0.637	0.227
	ILVR	38.27	0.372	0.656	39.51	0.278	0.726	71.24	0.421	0.662	–	–	–	95.3	0.532	0.498
	ReSample	33.47	0.289	0.730	39.54	0.259	0.799	61.24	0.439	0.708	55.76	0.370	0.637	49.19	0.339	0.777
	PnP-ADMM	114.7	0.677	0.300	78.24	0.367	0.657	100.6	0.519	0.669	–	–	–	97.27	0.433	0.761
	Score-SDE	127.1	0.659	0.517	54.07	0.354	0.612	120.3	0.667	0.436	–	–	–	170.7	0.701	0.256
	ADMM-TV	189.3	0.510	0.676	87.69	0.319	0.785	155.7	0.588	0.634	–	–	–	130.9	0.523	0.679
	PnP-DM	34.92	0.296	0.736	37.67	0.258	0.797	61.06	0.433	0.707	55.33	0.372	0.636	50.10	0.336	0.786
	DAPS	33.94	0.282	0.741	35.46	0.248	0.801	60.12	0.419	0.709	54.82	0.365	0.639	49.62	0.333	0.789
	DMPlug	32.85	0.226	0.748	34.28	0.247	0.804	57.42	0.407	0.714	53.13	0.366	0.642	48.96	0.324	0.793
	C-DPS	**32.37**	**0.214**	**0.755**	**33.24**	**0.236**	0.807	**56.36**	**0.391**	0.712	**52.06**	**0.352**	0.644	**47.30**	**0.316**	**0.795**

Experiments (Qualitative Results)