A Mixture-based framework for guiding Diffusion models

Alain Oliviero Durmus

Ecole polytechnique

Yazid Janati, Badr Moufad, Medhi Abou El Qassime, Eric Moulines and Jimmy Olsson

Bayesian inverse problems

$$Y = A(X) + Z\,, \quad X \sim p_0$$

Goal: Reconstruct X.

In many examples, A is a linear operator.

Bayesian inverse problems

$$Y = A(X) + Z\,, \quad X \sim p_0$$

Probabilistic modelling:

- if $Z \sim \mathcal{N}(0, \Sigma_y)$ then $g_0(y \,|\, x) = \mathrm{N}(y; A(x), \Sigma_y)$ and the joint model is

$$p_{(X,Y)}(x, y) = g_0(y \,|\, x) p_0(x)$$

- Z could be Poisson noise.

Bayesian inverse problems

The reconstructions are encoded in the posterior distribution

$$\pi_0^y(x) \propto g_0(y \,|\, x)\, p_0(x)$$

where $g_0(y \,|\, x) = \mathrm{N}(y; A(x), \Sigma_y)$.

Sampling plausible reconstructions

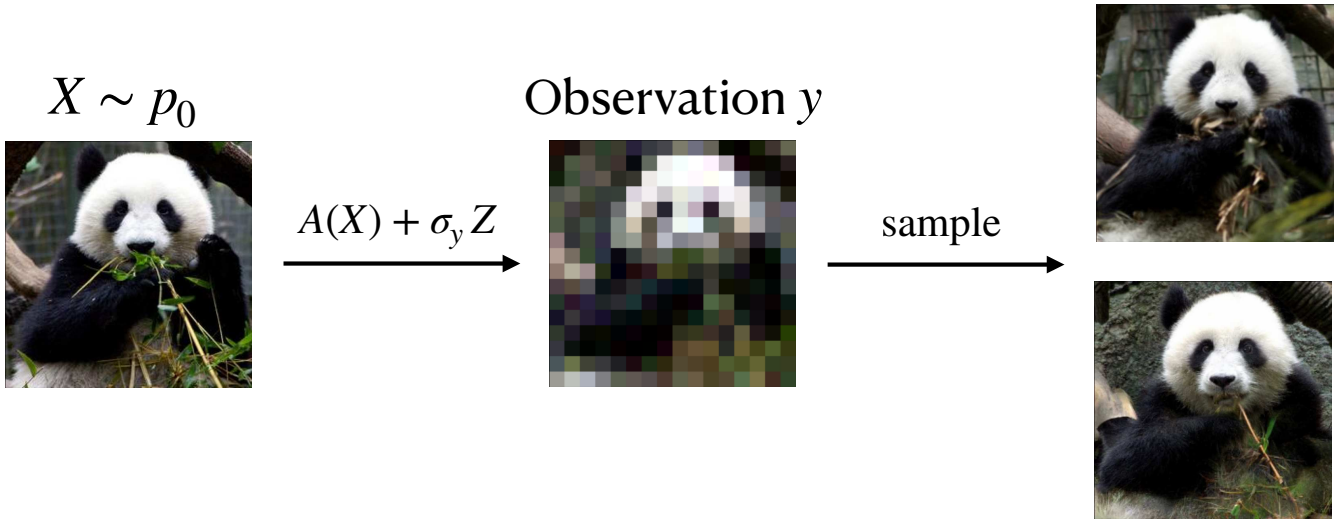
$\Longleftrightarrow$

Drawing samples from π_0^y

Bayesian inverse problems

$$Y = A(X) + Z\,, \quad X \sim p_0$$

Given a realisation $Y = y$, sample the most plausible reconstructions X



Conditional generative models

- We can train a generative model $\pi_0^\theta(\,\cdot\,|\,y) \approx \pi_0^y$ using a paired dataset $(X_i, Y_i)_{i=1}^N \overset{\text{i.i.d.}}{\sim} g_0(\mathrm{d}y\,|\,x)p_0(\mathrm{d}x)$

$$\text{minimize w.r.t. } \theta \quad \mathbb{E}\left[\mathsf{KL}\left(\pi_0^Y \| \pi_0^\theta(\,\cdot\,|\,Y)\right)\right]$$

- Given observation $Y = y$, reconstruct by sampling from $\pi_0^\theta(\,\cdot\,|\,y)$

Conditional generative models

- We can train a generative model $\pi_0^\theta(\,\cdot\,|\,y) \approx \pi_0^y$ using a paired dataset $(X_i, Y_i)_{i=1}^N \overset{\text{i.i.d.}}{\sim} g_0(\mathrm{d}y\,|\,x)p_0(\mathrm{d}x)$
- Drawbacks:
  1. Train from scratch if we change $g_0(\mathrm{d}y\,|\,x)$

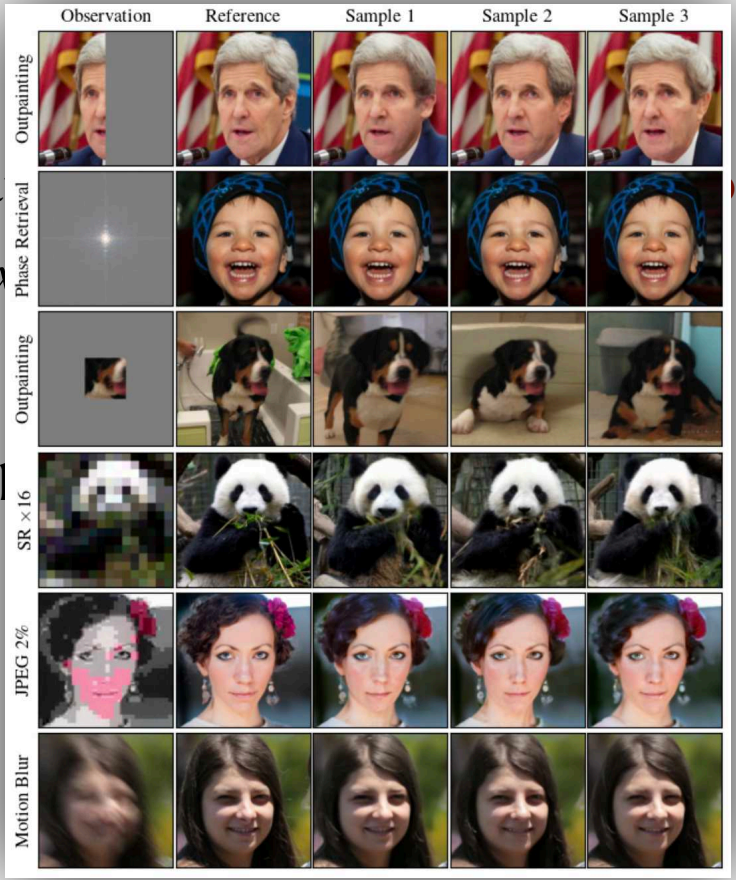
Conditional generative models

- We can train a genera aired dataset

  $(X_i, Y_i)_{i=1}^N \overset{\text{i.i.d.}}{\sim} g_0(\mathrm{d}y \mid x$

- Drawbacks:

  1. Train from scratc



Conditional generative models

- We can train a generative model $\pi_0^\theta(\,\cdot\,|\,y) \approx \pi_0^y$ using a paired dataset
  $(X_i, Y_i)_{i=1}^N \overset{\text{i.i.d.}}{\sim} g_0(\mathrm{d}y\,|\,x)p_0(\mathrm{d}x)$
- Drawbacks:
  1. Train from scratch if we change $g_0(\mathrm{d}y\,|\,x)$
  2. Paired dataset may not be available; what if we are interested in sampling

     $$\pi_0(x) \propto g_0(x)p_0(x)$$

     where g_0 is instead a reward function (RLHF, style...)

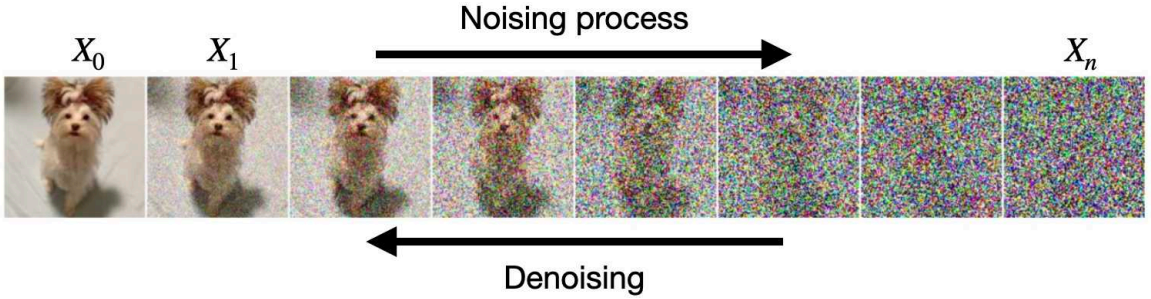
Setting

$$\pi_0^y(x) \propto g_0(y \mid x)\, p_0(x)$$

Given a **pre-trained** diffusion model $p_0^\theta \approx p_0$, develop an algorithm for sampling from π_0^y with no further model training

Posterior sampling with Denoising Diffusion models

Denoising Diffusion models



$$= \mathrm{N}(x_{k+1}; (\alpha_{k+1}/\alpha_k)x_k, (1 - \alpha_{k+1}^2/\alpha_k^2)\mathrm{I})$$

$$p_{0:n}(x_{0:n}) = p_0(x_0) \prod_{k=0}^{n-1} q_{k+1|k}(x_{k+1} \mid x_k) \qquad \text{where } \alpha_0 = 1,\ \alpha_n \approx 0$$

$$= p_n(x_n) \prod_{k=0}^{n-1} p_{k|k+1}(x_k \mid x_{k+1})$$

$$\approx \mathcal{N}(0,\mathrm{I}) \qquad \text{intractable}$$

[Ho, J., Jain, A. and Abbeel, P., 2020. Denoising diffusion probabilistic models.]

Training objective

$$p^{\theta}_{k|k+1}(x_k \,|\, x_{k+1}) = \mathrm{N}\big(x_k; \mu^{\theta}_{k|k+1}(x_{k+1}), \Sigma^{\theta}_{k|k+1}(x_{k+1})\big)$$

$$\begin{aligned}\mathsf{KL}(p_{\text{data}} \parallel p^{\theta}_0) &\leq \mathsf{KL}(p_{0:n} \parallel p^{\theta}_{0:n}) \\ &= \sum_{k=0}^{n-1} \mathbb{E}_{p_{k+1}}\Big[\mathsf{KL}\big(p_{k|k+1}(\,\cdot\,|X_{k+1}) \parallel p^{\theta}_{k|k+1}(\,\cdot\,|X_{k+1})\big)\Big] + C_1\end{aligned}$$

Gaussian moment matching

Optimal parameters:

$$\begin{cases} \mu^{\theta_*}_{k|k+1}(x_{k+1}) = \mathbb{E}[X_k \,|\, X_{k+1} = x_{k+1}] \\ \Sigma^{\theta_*}_{k|k+1}(x_{k+1}) = \mathbb{C}\mathrm{ov}[X_k \,|\, X_{k+1} = x_{k+1}] \end{cases}$$

Parameterization

$$p_{k|k+1}(x_k \,|\, x_{k+1}) = \int q_{k|0,k+1}(x_k \,|\, x_0, x_{k+1}) p_{0|k+1}(x_0 \,|\, x_{k+1}) \, \mathrm{d}x_0 \,, \quad k \geq 1$$

where $\quad q_{k|0,k+1}(x_k \,|\, x_0, x_{k+1}) = \mathrm{N}(x_k; \eta_k x_0 + \gamma_k x_{k+1}, \sigma^2_{k|0,k+1} I_d)$

It follows that

$$\begin{cases} \mathbb{E}[X_k \,|\, X_{k+1}] = \gamma_k X_{k+1} + \eta_k \, D_{k+1}(X_{k+1}) \\ \mathbb{C}\mathrm{ov}[X_k \,|\, X_{k+1}] = \sigma^2_{k|0,k+1} I_d + \eta_k^2 \mathbb{C}\mathrm{ov}[X_0 \,|\, X_{k+1}] \end{cases}$$

where $\quad D_{k+1}(X_{k+1}) = \mathbb{E}[X_0 \,|\, X_{k+1}]$

[Ho, J., Jain, A. and Abbeel, P., 2020. Denoising diffusion probabilistic models]
[Song, Jiaming, Chenlin Meng, and Stefano Ermon, 2020. Denoising diffusion implicit models.]

Parameterization

$$p_{k|k+1}(x_k \,|\, x_{k+1}) = \int q_{k|0,k+1}(x_k \,|\, x_0, x_{k+1}) p_{0|k+1}(x_0 \,|\, x_{k+1}) \,\mathrm{d}x_0 \,, \quad k \geq 1$$

where $\quad q_{k|0,k+1}(x_k \,|\, x_0, x_{k+1}) = \mathrm{N}(x_k; \eta_k x_0 + \gamma_k x_{k+1}, \sigma^2_{k|0,k+1} I_d)$

It follows that
$$\begin{cases} \mathbb{E}[X_k \,|\, X_{k+1}] = \gamma_k X_{k+1} + \eta_k \,\mathbb{E}[X_0 \,|\, X_{k+1}] \\ \mathbb{C}\mathrm{ov}[X_k \,|\, X_{k+1}] = \sigma^2_{k|0,k+1} I_d + \eta_k^2 \mathbb{C}\mathrm{ov}[X_0 \,|\, X_{k+1}] \end{cases}$$

The usual parameterization is
$$\begin{cases} \mu^\theta_{k|k+1}(x_{k+1}) = \gamma_k x_{k+1} + \eta_k \, D^\theta_{k+1}(x_{k+1}) \\ \Sigma^\theta_{k|k+1}(x_{k+1}) = \sigma^2_{k|0,k+1} I_d \end{cases}$$

Basically, $\quad p^\theta_{k|k+1}(x_k \,|\, x_{k+1}) = q_{k|0,k+1}(x_k \,|\, D^\theta_{k+1}(x_{k+1}), x_{k+1})$

[Ho, J., Jain, A. and Abbeel, P., 2020. Denoising diffusion probabilistic models]
[Song, Jiaming, Chenlin Meng, and Stefano Ermon, 2020. Denoising diffusion implicit models.]

Denoising objective

$$\mathsf{KL}(p_{0:n} \,\|\, p^{\theta}_{0:n}) = \sum_{k=0}^{n-1} \mathbb{E}_{p_{k+1}}\Big[\mathsf{KL}\big(p_{k|k+1}(\,\cdot\,|X_{k+1}) \,\|\, p^{\theta}_{k|k+1}(\,\cdot\,|X_{k+1})\big)\Big] + C_1$$

$$= \sum_{k=1}^{n} w_k \,\mathbb{E}\Big[\|\mathbb{E}[X_0\,|\,X_k] - D^{\theta}_k(X_k)\|^2\Big] + C_2$$

use previous parameterization

$$= \sum_{k=1}^{n} w_k \,\mathbb{E}\Big[\|X_0 - D^{\theta}_k(X_k)\|^2\Big] + C_3$$

Training a Denoising Diffusion model

$\Longleftrightarrow$

Training denoisers $(D^{\theta}_k)_{k=1}^{n}$

Denoising objective

Training a Denoising Diffusion model

$\Longleftrightarrow$

Training denoisers $(s_k^\theta)_{k=1}^n$ approximating $(\nabla \log p_k)_{k=1}^n$

$$\nabla \log p_k(x_k) = \frac{-x_k + \alpha_k D_k(x_k)}{1 - \alpha_k^2}$$

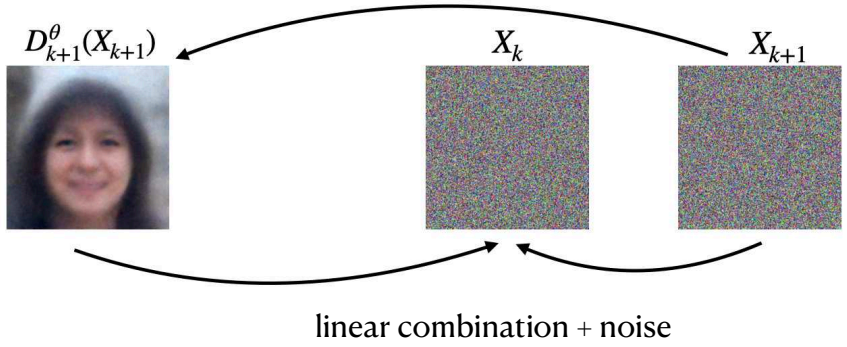
Backward transitions

$$p_{k|k+1}(x_k \mid x_{k+1}) = \int q_{k|0,k+1}(x_k \mid x_0, x_{k+1}) p_{0|k+1}(x_0 \mid x_{k+1}) \, \mathrm{d}x_0$$

$$\approx q_{k|0,k+1}(x_k \mid D^{\theta}_{k+1}(x_{k+1}), x_{k+1})$$

where $\quad D^{\theta}_{k+1}(x_{k+1}) \approx \mathbb{E}[X_0 \mid X_{k+1} = x_{k+1}]$

Implicitely assumes that $p_{0|k+1}(\,\cdot \mid x_{k+1}) \approx \delta_{D^{\theta}_{k+1}(x_{k+1})}$

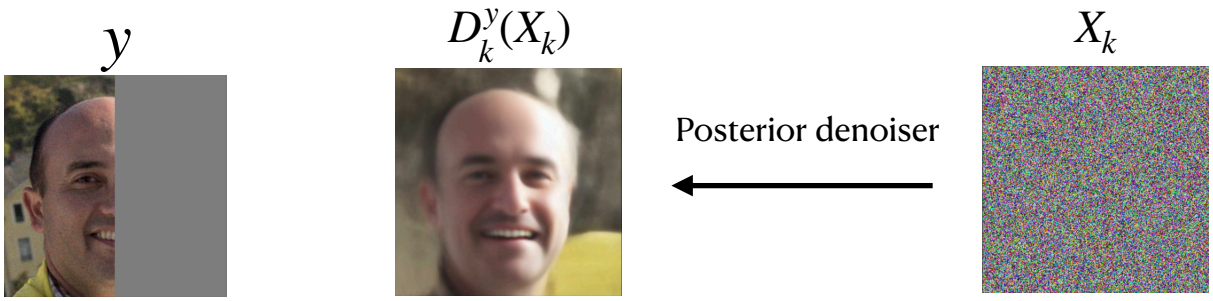


Posterior Diffusion model?

- We now assume that we have a pre-trained DDM for p_0.
- To have a DDM for π_0^y, we need to approximate the posterior denoiser

$$D_k^y(x_k) = \int x_0 \, \pi_{0|k}^y(x_0 \,|\, x_k) \, \mathrm{d}x_0$$

$$\text{where} \quad \pi_{0|k}^y(x_0 \,|\, x_k) \propto \pi_0^y(x_0) q_{k|0}(x_k \,|\, x_0)$$



Posterior Diffusion model?

- We now assume that we have a pre-trained DDM for p_0.
- To have a DDM for π_0^y, we need to approximate

$$\nabla \log \pi_k^y(x_k) \quad \text{where} \quad \pi_k^y(x_k) = \int q_{k|0}(x_k \,|\, x_0)\pi^y(x_0)\,\mathrm{d}x_0$$

Posterior denoiser

Define $\pi_k^y(x_k) = \int q_{k|0}(x_k \,|\, x_0)\, \pi_0^y(x_0) \mathrm{d}x_0.$

$$= p_{0|k}(x_0 \,|\, x_k) p_k(x_k)$$

$$\pi_k^y(x_k) \propto \int g_0(y \,|\, x_0) q_{k|0}(x_k \,|\, x_0) p_0(x_0)\, \mathrm{d}x_0$$

$$\propto \left[\int g_0(y \,|\, x_0) p_{0|k}(x_0 \,|\, x_k)\, \mathrm{d}x_0 \right] p_k(x_k)$$

$$:= g_k(y \,|\, x_k)$$

$$D_k^y(x_k) = \frac{x_k + (1-\alpha_k^2) \nabla \log \pi_k^y(x_k)}{\alpha_k} = \underset{\approx D_k^\theta(x_k)}{D_k(x_k)} + \alpha_k^{-1}(1-\alpha_k^2) \underset{??}{\nabla \log g_k(y \,|\, x_k)}$$

Posterior denoiser approximation

DPS approximation

$$g_k(y \mid x_k) = \int g_0(y \mid x_0) p_{0|k}(x_0 \mid x_k)\, \mathrm{d}x_0 \approx g_0(y \mid D_k(x_k))$$

implicitely assumes that $p_{0|k}(\,\cdot \mid x_k) \approx \delta_{D_k(x_k)}$

$$D_k^y(x_k) \approx D_k(x_k) + \alpha_k^{-1}(1-\alpha_k^2)\, \nabla_{x_k} \log g_0\big(y \mid D_k(x_k)\big)$$

- Samples diverge after a few iterations
- Instead, [Chung et al. 2023] plugs $\dfrac{C_k}{\|y - A(D_k(x_k))\|} \nabla_{x_k} \log g_0\big(y \mid D_k(x_k)\big)$

[Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M. and Fleet, D.J., 2022. Video diffusion models.]
[Chung, H., Kim, J., Mccann, M.T., Klasky, M.L. and Ye, J.C., 2022. Diffusion posterior sampling for general noisy inverse problems.]

Posterior denoiser approximation

DPS approximation

$$g_k(y \mid x_k) = \int g_0(y \mid x_0) p_{0|k}(x_0 \mid x_k) \, \mathrm{d}x_0 \approx g_0(y \mid D_k(x_k))$$

implicitely assumes that $p_{0|k}(\,\cdot \mid x_k) \approx \delta_{D_k(x_k)}$

- This Approximation is relatively good for k close to 0
- How can we use this insight?

[Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M. and Fleet, D.J., 2022. Video diffusion models.]
[Chung, H., Kim, J., Mccann, M.T., Klasky, M.L. and Ye, J.C., 2022. Diffusion posterior sampling for general noisy inverse problems.]

Likelihood approximation

For all $\ell \leq k$

$$g_k(y \,|\, x_k) = \int g_\ell(y \,|\, x_\ell) p_{\ell|k}(x_\ell \,|\, x_k) \, \mathrm{d}x_\ell \,.$$

Letting $\hat{g}_\ell(y \,|\, \cdot\,) = g_0(y \,|\, D_\ell(\,\cdot\,))$:

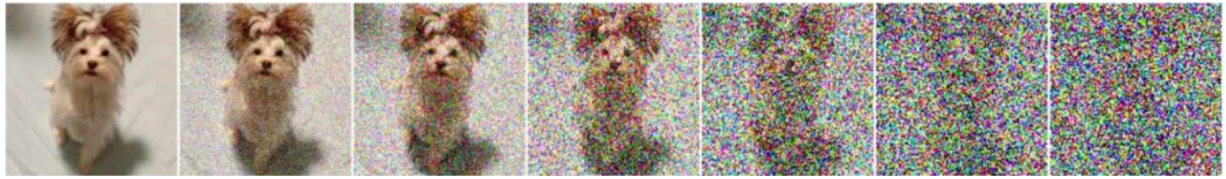
$$\hat{g}_k^\ell(y \,|\, x_k) = \int \hat{g}_\ell(y \,|\, x_\ell) p_{\ell|k}(x_\ell \,|\, x_k) \, \mathrm{d}x_\ell \,, \quad \ell \in [1 : k-1]$$

We get $k-1$ different approximations

with intractable scores!

Distribution path

$X_0 \sim p_0$ $\cdots$ $X_k \sim p_k$ $\cdots$ $X_n \sim p_n$



$$p_k(x_k) = \int p_0(x_0) q_{k|0}(x_k \mid x_0)\, \mathrm{d}x_0$$

Diffusion model for $\pi_0^y \iff$ follow path $(\pi_k^y)_{k=n}^0$ where

$$\pi_k^y(x_k) = \int \pi_0^y(x_0) q_{k|0}(x_k \mid x_0)\, \mathrm{d}x_0$$
$$\propto g_k(y \mid x_k) p_k(x_k)$$

Alternative distribution path

Recall that $\hat{g}_k^{\ell}(y \,|\, x_k) = \int \hat{g}_{\ell}(y \,|\, x_{\ell}) p_{\ell|k}(x_{\ell} \,|\, x_k) \, \mathrm{d}x_{\ell}$

$$\hat{\pi}_k^{\ell}(x_k) \propto \hat{g}_k^{\ell}(x_k) p_k(x_k) \;, \quad \ell \in [1, k-1]$$

We consider the following mixture approximation

$$\hat{\pi}_k^{y}(x_k) = \sum_{\ell=1}^{k-1} w_k^{\ell} \hat{\pi}_k^{\ell}(x_k) \;, \quad \text{where} \quad \sum_{\ell=1}^{k-1} w_k^{\ell} = 1 \quad k \geq 2$$

$$\hat{\pi}_1^{y}(x_1) \propto \hat{g}_1^{1}(x_1) p_1(x_1)$$

$\rightarrow$ Sequentially sample each marginal $(\hat{\pi}_k^{y})_{k=n}^{1}$: $(\hat{X}_k)_{k=n}^{1}$

Data augmentation

$$\hat{\pi}_k^y(x_k) = \sum_{\ell=1}^{k-1} w_k^\ell \hat{\pi}_k^\ell(x_k)$$

Given $\hat{X}_{k+1}$, we aim to sample $\hat{X}_k$:

First draw index $\ell_k \sim \text{Categorical}(\{w_k^i\}_{i=1}^{k-1})$ then sample

$$\begin{aligned}
\hat{\pi}_k^\ell(x_k) &\propto \hat{g}_k^\ell(x_k) p_k(x_k) \\
&\propto \int \hat{g}_\ell(x_\ell) p_{\ell|k}(x_\ell \mid x_k) p_k(x_k)\, \mathrm{d}x_\ell \\
&\propto \int p_{0|\ell}(x_0 \mid x_\ell) \hat{g}_\ell(x_\ell) p_{\ell|k}(x_\ell \mid x_k) p_k(x_k)\, \mathrm{d}x_0 \mathrm{d}x_\ell \\
&:= \hat{\pi}^y_{0,\ell,k}(x_0, x_\ell, x_k)
\end{aligned}$$

$\rightarrow$ Sample $(\hat{X}_0, \hat{X}_\ell, \hat{X}_k) \sim \hat{\pi}^y_{0,\ell,k}$ then keep $\hat{X}_k$

Gibbs sampling

$\hat{\pi}^y_{0,\ell,k}$ has rather nice full conditionals:

$$\hat{\pi}^y_{0|\ell,k}(x_0 \,|\, x_\ell, x_k) = p_{0|\ell}(x_0 \,|\, x_\ell) \qquad \text{denoising}$$

$$\hat{\pi}^y_{\ell|0,k}(x_\ell \,|\, x_0, x_k) \propto \hat{g}_\ell(y \,|\, x_\ell) q_{\ell|0,k}(x_\ell \,|\, x_0, x_\ell)$$

$$\hat{\pi}^y_{k|0,\ell}(x_k \,|\, x_0, x_\ell) = q_{k|\ell}(x_k \,|\, x_\ell) \qquad \text{noising}$$

Gibbs sampling

Deterministic scan Gibbs sampler:

Markov chain $(X_0^r, X_\ell^r, X_k^r)_{r\in\mathbb{N}}$ with stationary distribution $\hat{\pi}^y_{0,\ell,k}$ and transition

$$\hat{X}_\ell^{r+1} \sim \hat{\pi}^y_{\ell|0,k}(\,\cdot\,|\,\hat{X}_0^r, \hat{X}_k^r) \qquad \text{intractable step}$$

$$\hat{X}_0^{r+1} \sim p_{0|\ell}(\,\cdot\,|\,\hat{X}_\ell^{r+1})$$

$$\hat{X}_k^{r+1} \sim q_{k|\ell}(\,\cdot\,|\,\hat{X}_\ell^{r+1})$$

Approximate Gibbs sampler

Given (X_0^r, X_ℓ^r, X_k^r) :

1. Perform a few gradient steps on

$$\varphi \mapsto \mathsf{KL}\big(\mathcal{N}(\mu_\ell^\varphi, \Sigma_\ell^\varphi) \| \hat{\pi}^y_{\ell|0,k+1}(\,\cdot\,|\hat{X}_0^r, X_k^r)\big) \quad \text{wrt} \quad (\mu_\ell^\varphi, \Sigma_\ell^\varphi)$$

   and then draw $\hat{X}_\ell^{r+1} \sim \mathcal{N}(\mu_{\ell_k}^\varphi, \Sigma_{\ell_k}^\varphi)$.

2. Draw $\hat{X}_0^{r+1} \sim p_{0|\ell}(\,\cdot\,|\hat{X}_\ell^{r+1})$

3. Draw $\hat{X}_k^{r+1} \sim q_{k|\ell}(\,\cdot\,|\hat{X}_\ell^{r+1})$

How do we perform step 1?

KL minimization

Denote $\lambda^{\varphi}_{\ell|k+1} = \mathcal{N}(\mu^{\varphi}_{\ell}, \Sigma^{\varphi}_{\ell})$. In practice we set $\Sigma^{\varphi}_{\ell} = \mathrm{diag}(\mathrm{e}^{v_\ell})$ where $v_\ell \in \mathbb{R}^d$.

Minimizing $\varphi \mapsto \mathsf{KL}\big(\lambda^{\varphi}_{\ell|0,k} \| \hat{\pi}^{y}_{\ell|0,k}(\,\cdot\,|\,X^r_0, X^r_k)\big)$ is equivalent to minimizing

$$\mathscr{L}_k(\varphi) = -\,\mathbb{E}_{\lambda^{\varphi}_{\ell|0,k}}\big[\log g_0(y\,|\,D_\ell(\hat{X}^{\varphi}_{\ell})\big] + \mathsf{KL}\big(\lambda^{\varphi}_{\ell|0,k} \| q_{\ell|0,k}(\,\cdot\,|\,X^r_0, X^r_k)\big)$$

$$= -\,\mathbb{E}\big[\log g_0(y\,|\,D_\ell\big(\mu^{\varphi}_{\ell} + (\Sigma^{\varphi}_{\ell})^{1/2} Z\big)\big] + \mathsf{KL}\big(\lambda^{\varphi}_{\ell|0,k} \| q_{\ell|0,k}(\,\cdot\,|\,X^r_0, X^r_k)\big)$$

Monte Carlo estimate — closed form

Some questions

Some questions

- Why not sample the mixture index too?

Some questions

- Why not sample the mixture index too?
- Why not a simpler data augmentation?

Some questions

- Why not sample the mixture index too?
- Why not a simpler data augmentation?
- How do you choose the weights?

Some questions

- Why not sample the mixture index too?
- Why not a simpler data augmentation?
- How do you choose the weights?

Figure 4: Evolution of the running state $\hat{X}_0^*$ in Algorithm 2 for the two time-sampling distributions given in (21) and (22).

Experiments

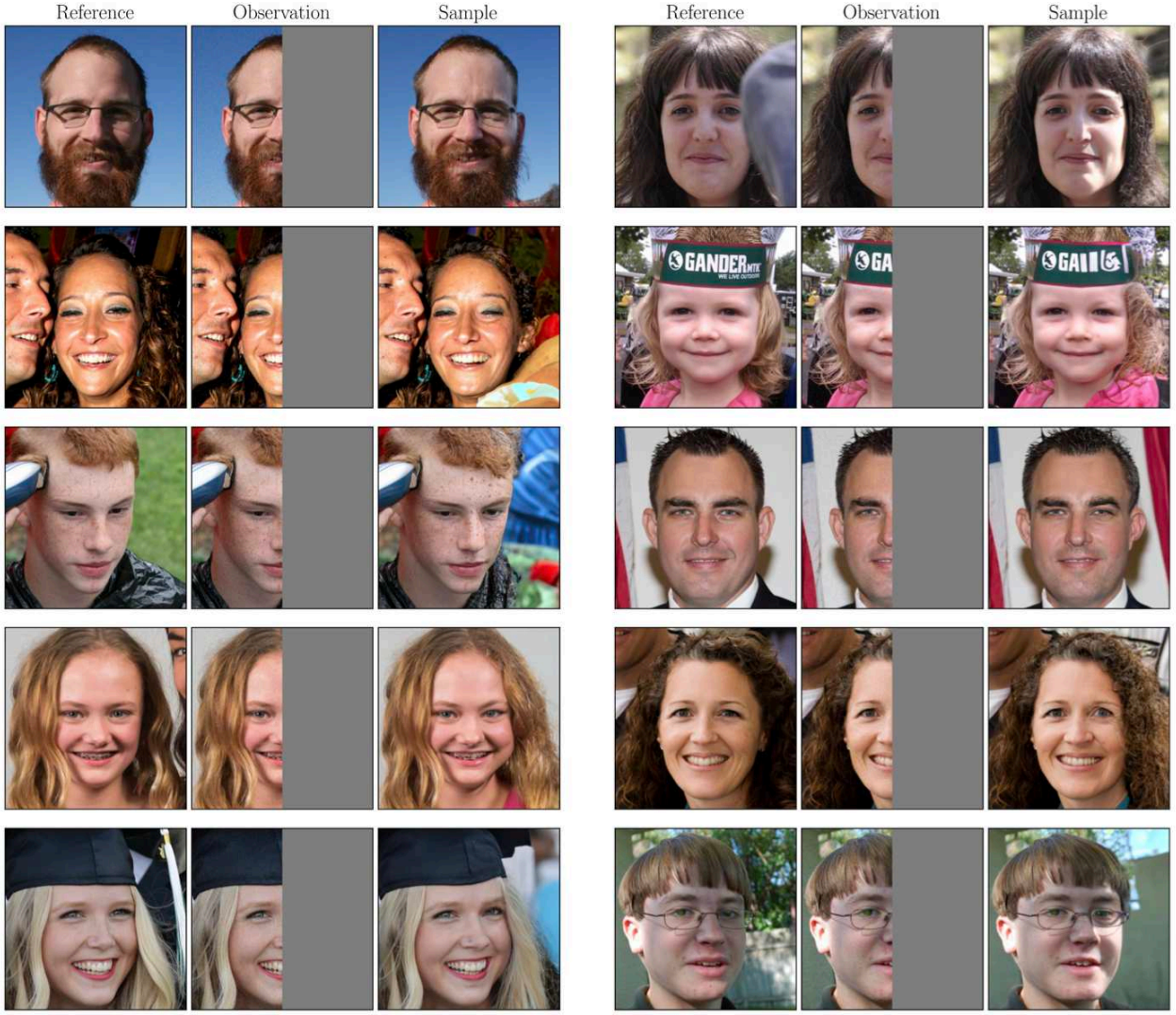
Image inverse problems

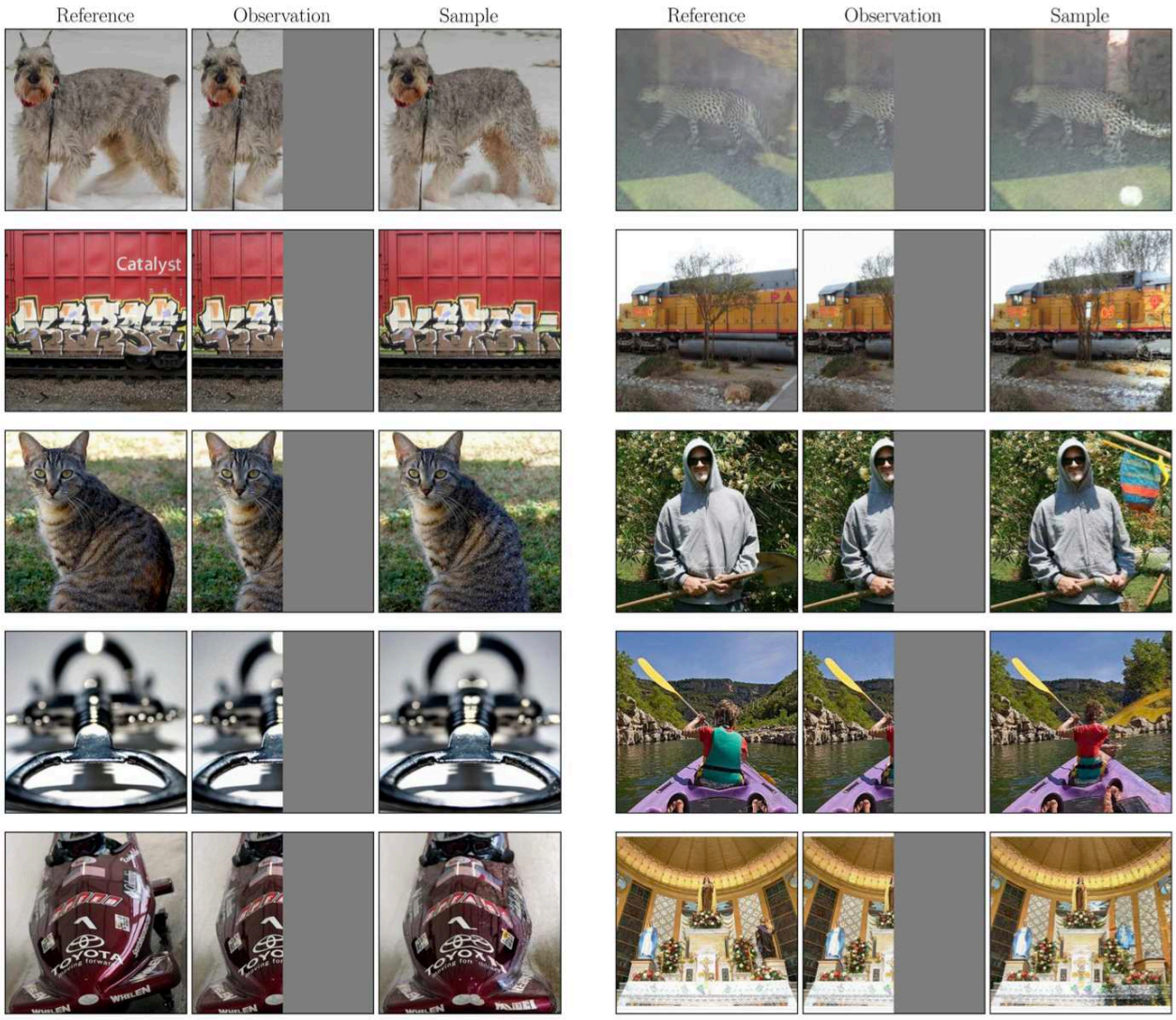
- Much harder evaluation; we do not have samples from the posterior
- We compare the features against the ground truth image

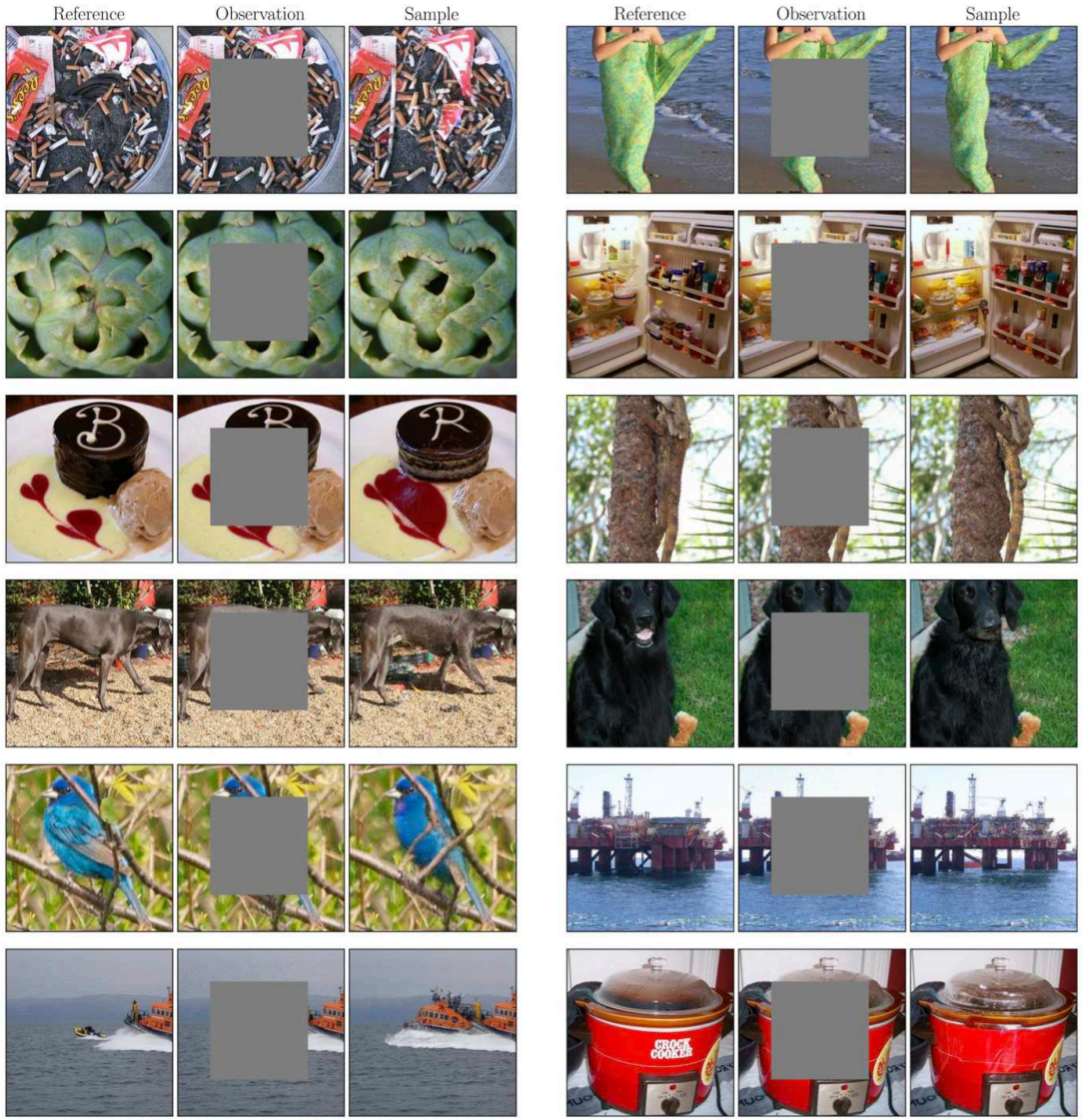
Table 1: Mean LPIPS for linear/nonlinear imaging tasks on the `FFHQ` and `ImageNet` datasets with $\sigma_{\mathbf{y}} = 0.05$. Lower metrics are better.

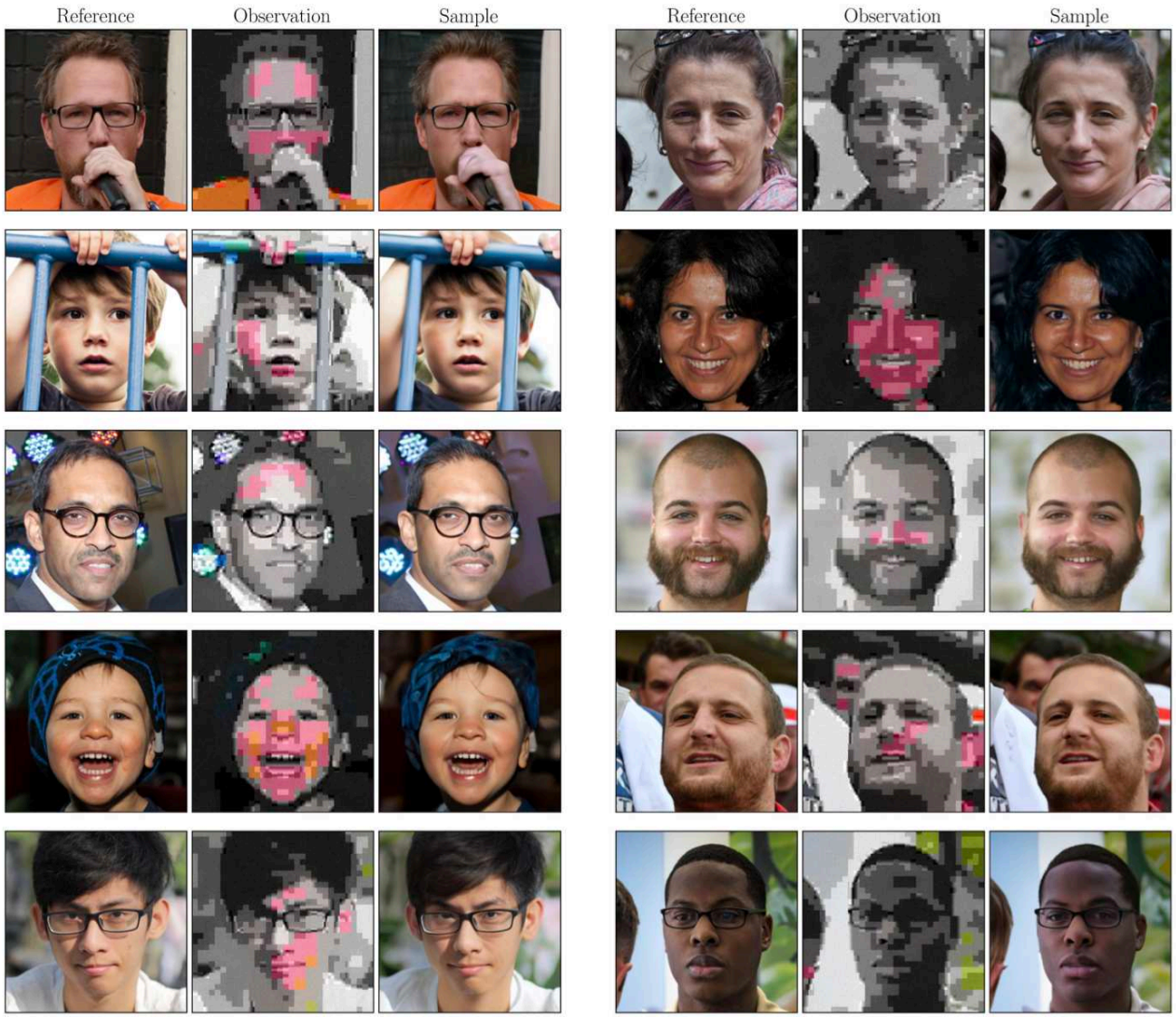
	FFHQ								ImageNet							
Task	MGDM	DPS	PGDM	DDNM	DiffPIR	ReDDiff	DAPS	PnP-DM	MGDM	DPS	PGDM	DDNM	DiffPIR	ReDDiff	DAPS	PnP-DM
SR (×4)	0.09	0.09	0.30	0.15	0.10	0.39	0.16	0.10	0.26	0.25	0.56	0.34	0.31	0.57	0.37	0.66
SR (×16)	0.24	0.23	0.42	0.33	0.23	0.55	0.40	0.29	0.55	0.44	0.62	0.71	0.50	0.85	0.75	1.03
Box inpainting	0.10	0.17	0.17	0.12	0.14	0.19	0.13	0.18	0.23	0.35	0.29	0.28	0.30	0.36	0.30	0.42
Half mask	0.20	0.24	0.24	0.23	0.25	0.28	0.23	0.32	0.31	0.40	0.34	0.38	0.40	0.46	0.40	0.54
Gaussian Deblur	0.12	0.17	0.87	0.20	0.12	0.24	0.24	0.14	0.30	0.37	1.00	0.45	0.30	0.53	0.59	0.76
Motion Deblur	0.09	0.17	–	–	–	0.22	0.19	0.21	0.22	0.40	–	–	–	0.39	0.42	0.52
JPEG (QF = 2)	0.14	0.34	1.12	–	–	0.32	0.22	0.29	0.38	0.60	1.32	–	–	0.49	0.45	0.56
Phase retrieval	0.11	0.40	–	–	–	0.26	0.14	0.34	0.55	0.62	–	–	–	0.61	0.50	0.66
Nonlinear deblur	0.27	0.51	–	–	–	0.68	0.28	0.31	0.41	0.82	–	–	–	0.66	0.41	0.49
HDR	0.12	0.40	–	–	–	0.20	0.10	0.19	0.21	0.84	–	–	–	0.19	0.14	0.31

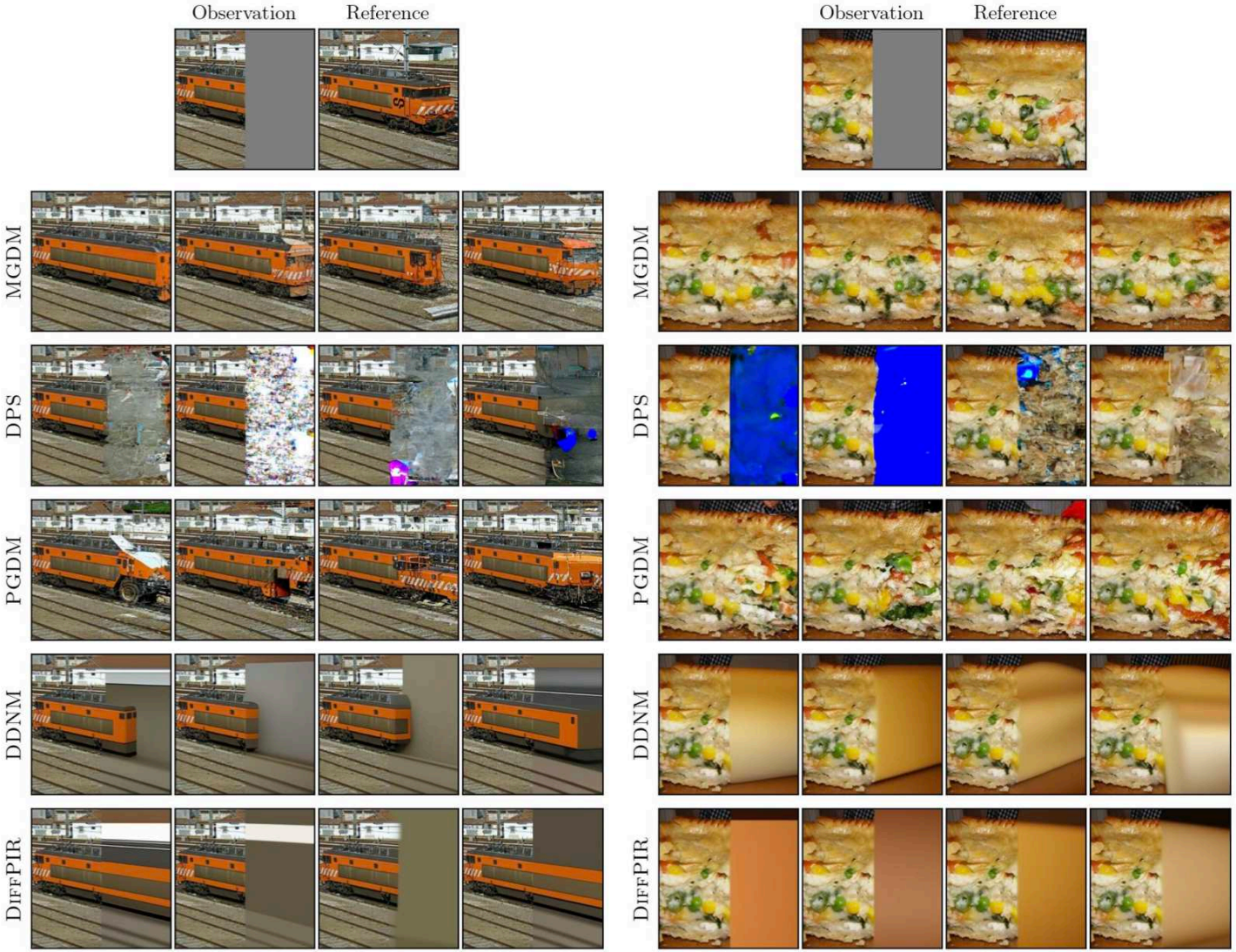
Some samples

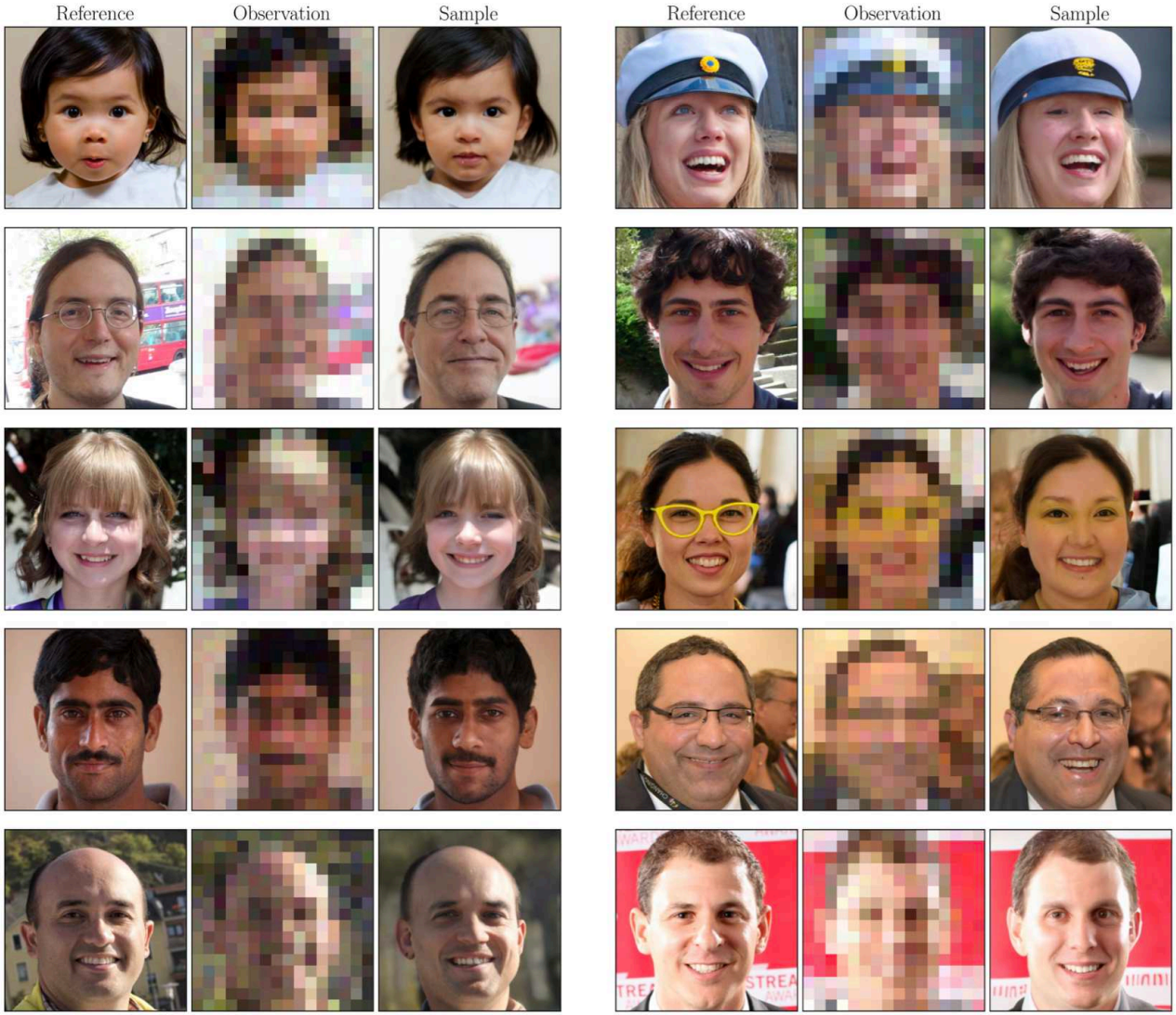

Reference
Observation
Sample
Reference
Observation
Sample


Reference
Observation
Sample
Reference
Observation
Sample
Catalyst


Reference
Observation
Sample
Reference
Observation
Sample
CROCK COOKER


Reference
Observation
Sample
Reference
Observation
Sample


Observation
Reference
Observation
Reference
MGDM
DPS
PGDM
DDNM
DiffPIR
MGDM
DPS
PGDM
DDNM
DiffPIR


Reference
Observation
Sample
Reference
Observation
Sample

Audio source separation

- Prior: diffusion model that generates 4 instruments: bass, drums, piano, guitar

$$\pi_0^y(x_0) \propto \mathrm{N}(y; \sum_{i=1}^{4} x_0^i, \sigma_y^2 I) p_0(x_0^1, x_0^2, x_0^3, x_0^4), \quad x_0 = [x_0^1, x_0^2, x_0^3, x_0^4]$$

$$\text{where} \quad y = \sum_{i=1}^{4} x_*^i$$

Audio source separation

- Prior: diffusion model that generates 4 instruments: bass, drums, piano, guitar

$$\pi_0^y(x_0) \propto \mathrm{N}(y; \sum_{i=1}^{4} x_0^i, \sigma_y^2 I) p_0(x_0^1, x_0^2, x_0^3, x_0^4), \quad x_0 = [x_0^1, x_0^2, x_0^3, x_0^4]$$

$$\text{where} \quad y = \sum_{i=1}^{4} x_*^i$$

Table 3: Mean SI-SDR$_\text{I}$ on `slakh2100` test dataset. The last row displays the mean over the four stems. Higher metrics are better.

Stems	MGDM	DPS	PGDM	DDNM	MSDM	ISDM	DEMUCS$_{512}$
Bass	18.49	16.50	16.41	14.94	17.12	19.36	17.16
Drums	18.07	18.29	18.14	19.05	18.68	20.90	19.61
Guitar	16.68	9.90	12.84	14.38	15.38	14.70	17.82
Piano	16.17	10.41	12.31	11.46	14.73	14.13	16.32
All	17.35	13.77	14.92	14.96	16.48	17.27	17.73

Audio source separation

- Prior: diffusion model that generates 4 instruments: bass, drums, piano, guitar

$$\pi_0^y(x_0) \propto \mathrm{N}(y; \sum_{i=1}^{4} x_0^i, \sigma_y^2 I) p_0(x_0^1, x_0^2, x_0^3, x_0^4), \quad x_0 = [x_0^1, x_0^2, x_0^3, x_0^4]$$

$$\text{where} \quad y = \sum_{i=1}^{4} x_*^i$$

Table 3: Mean SI-SDR$_\text{I}$ on `slakh2100` test dataset. The last row displays the mean over the four stems. Higher metrics are better.

Stems	MGDM	DPS	PGDM	DDNM	MSDM	ISDM	DEMUCS$_{512}$
Bass	18.49	16.50	16.41	14.94	17.12	19.36	17.16
Drums	18.07	18.29	18.14	19.05	18.68	20.90	19.61
Guitar	16.68	9.90	12.84	14.38	15.38	14.70	17.82
Piano	16.17	10.41	12.31	11.46	14.73	14.13	16.32
All	17.35	13.77	14.92	14.96	16.48	17.27	17.73

Independent prior: $$\pi_0^y(x_0) \propto \mathrm{N}(y; \sum_{i=1}^{4} x_0^i, \sigma_y^2 I) \prod_{i=1}^{4} p_0(x_0^i)$$

Parameters impact

	$R=1$	$R=2$	$R=4$	$R=6$	$R=1, G\gg 1$
Bass	15.46	18.07	18.53	18.49	19.89
Drums	16.28	17.93	18.19	18.07	18.95
Guitar	12.58	14.73	16.26	16.68	16.07
Piano	11.82	14.34	15.38	16.17	16.50
All	14.03	16.27	17.09	17.35	17.85

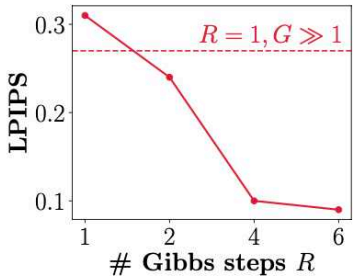


Figure 3: Performance of MGDM as a function of the number of Gibbs steps R. The setup $R=1, G\gg 1$ represents MGDM with $R=1$ and a number of gradient steps resulting in a runtime equivalent to using $R=6$. Left: Mean SI-SDR$_\text{I}$ for multisource–audio separation task on `slakh2100` test dataset. Right: Mean LPIPS for the phase retrieval task on `FFHQ`.