

Explore Low-Dimensional Structures for Constrained and Controllable Diffusion Models in Scientific Applications

Liyue Shen

Assistant Professor

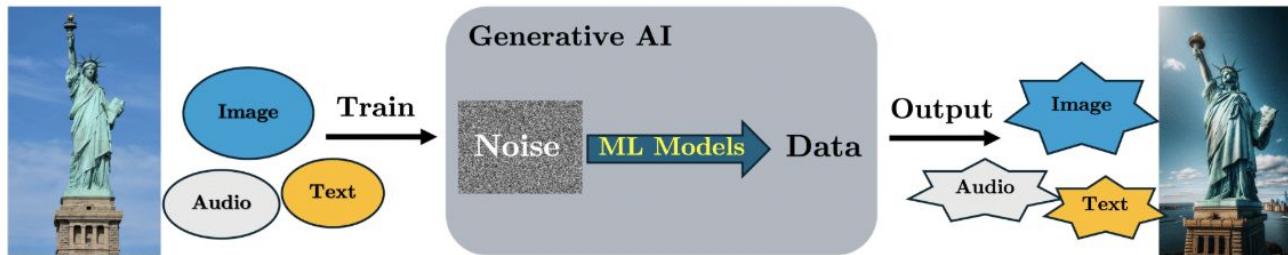
Electrical and Computer Engineering
University of Michigan



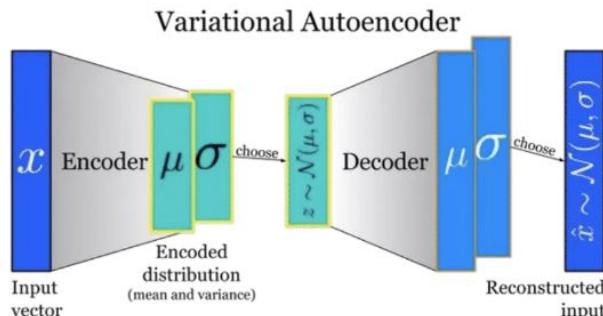
This Tutorial

- Session I: Introduction of Basic Low-dimensional Models
- Session II: Understanding Low-Dimensional Structures in Representation Learning
- Session III: Understanding Low-Dimensional Structures in Diffusion Generative Models
 - Low-Dimensional Models for Understanding Generalization in Diffusion Models - Qing Qu
 - *Explore Low-Dimensional Structures for Constrained and Controllable Diffusion Models in Scientific Applications - Liyue Shen*
- Session IV: Designing Deep Networks for Pursuing Low-Dimensional Structures
- Session V: Panel Discussion

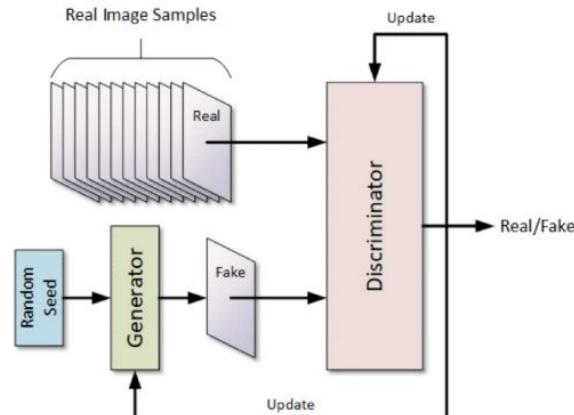
Recap: The Family of Generative Models



Generative models in the past:

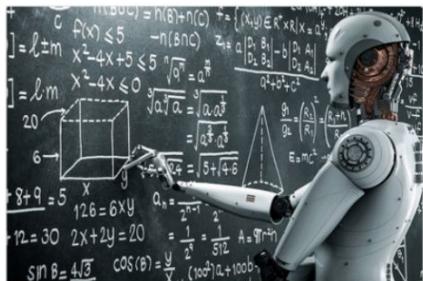


(a) VAE (Kingma & Wellings, 2013):
poor generation quality.



(b) GAN (Goodfellow et al. 2014):
unstable to train on large dataset.

Why is Diffusion Model?



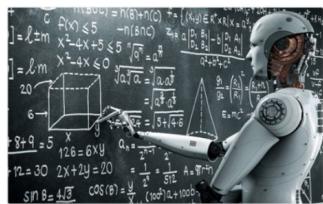
Mathematical Foundations



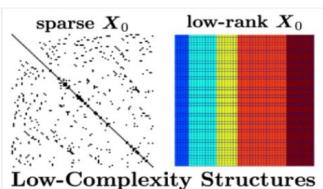
(Scientific) Applications

Low-dimensional structures are the key to address real-world high-dimensional data with small data regime in scientific applications

Application of Diffusion Model: Solving Inverse Problems



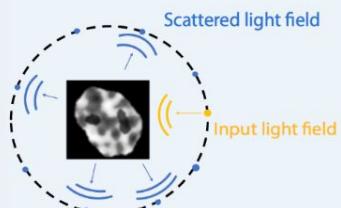
Mathematical Foundations



(Scientific) Applications

Inverse problems are common and important in scientific applications

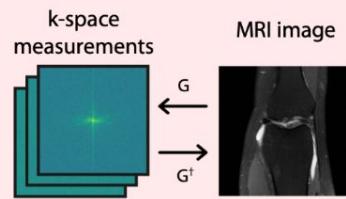
Linear inverse scattering



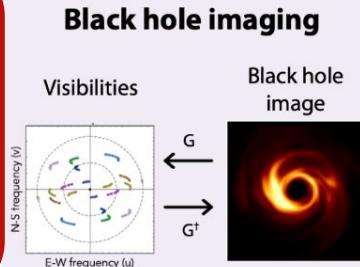
Scattered light field

input light field

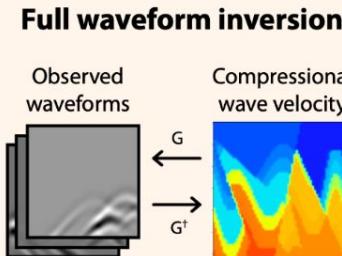
Compressed sensing MRI



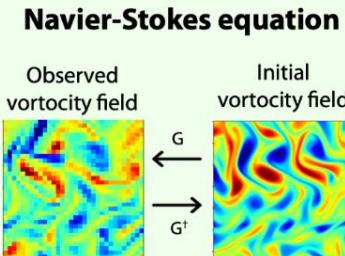
Black hole imaging



Full waveform inversion

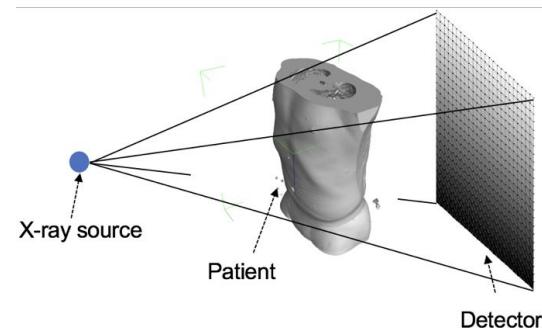


Navier-Stokes equation



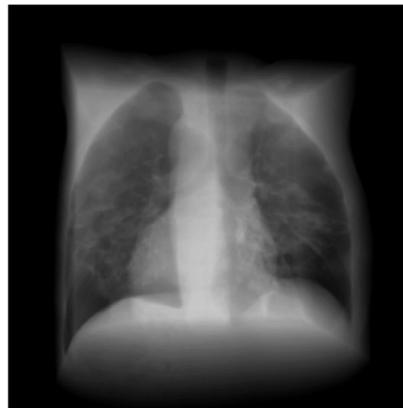
Medical Imaging

- X-ray Computed Tomography (CT) imaging
 - Reconstruct cross-sectional images of internal structures from X-ray projections at multiple views



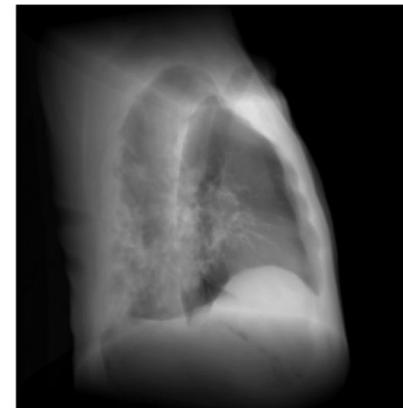
Medical Imaging

- X-ray Computed Tomography (CT) imaging
 - Reconstruct cross-sectional images of internal structures from X-ray projections at multiple views
- Reduce radiation dose in CT scanning
 - Sample sparse projections from fewer views



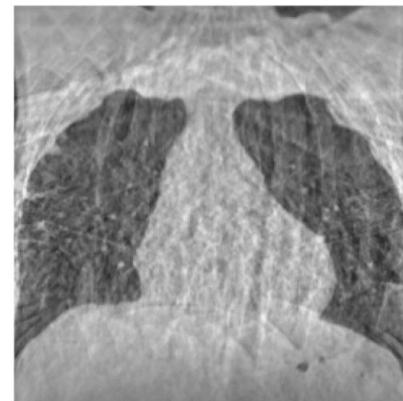
Dense-view (180)

6x



Sparse-view (30)

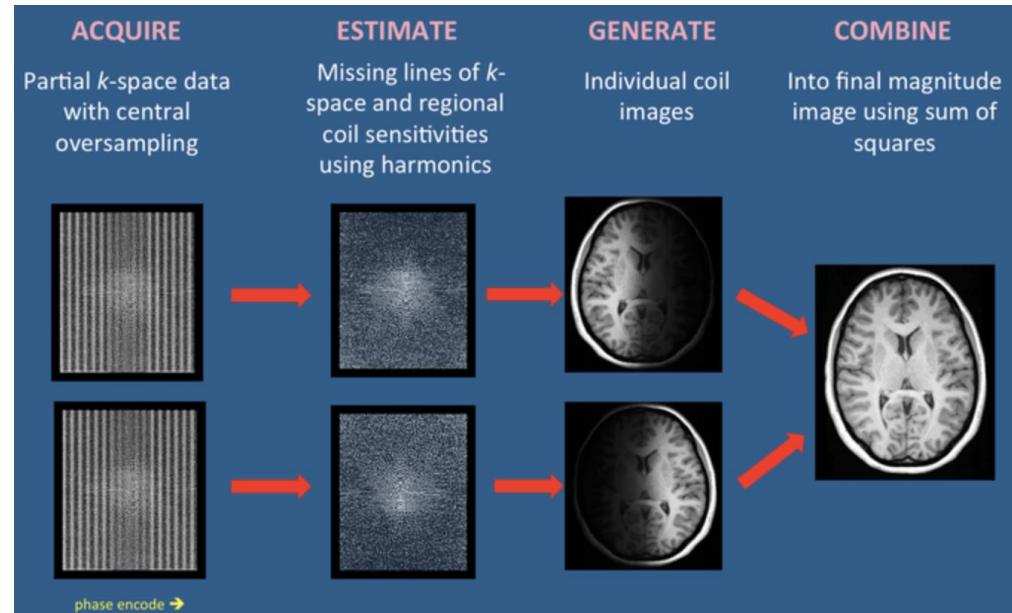
FBP



Reconstruction

Medical Imaging

- Magnetic Resonance Imaging (MRI)
 - Reconstruct cross-sectional images of internal structures from measurements in frequency space



Medical Imaging

- **Magnetic Resonance Imaging (MRI)**
 - Reconstruct cross-sectional images of internal structures from measurements in frequency space
- **Accelerate MRI scan**
 - Undersample frequency space data

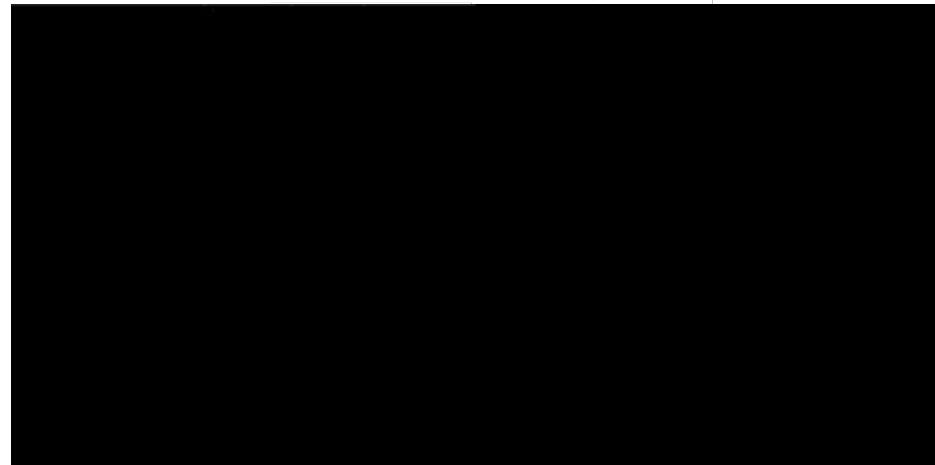
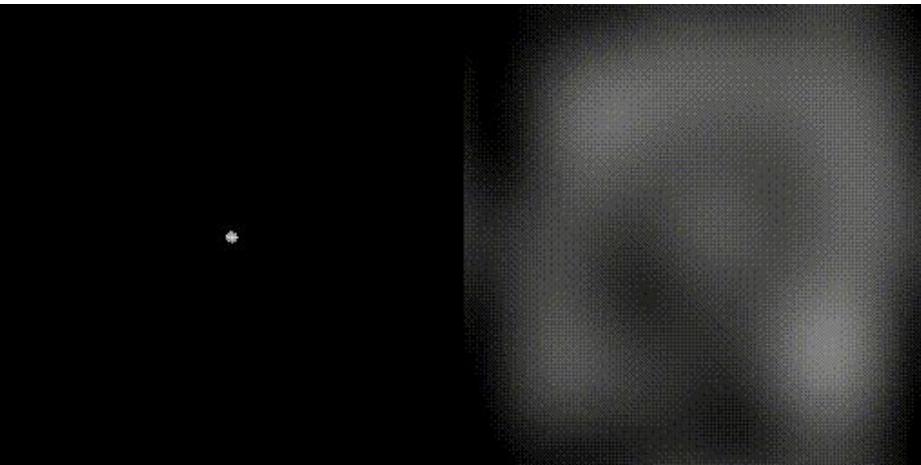
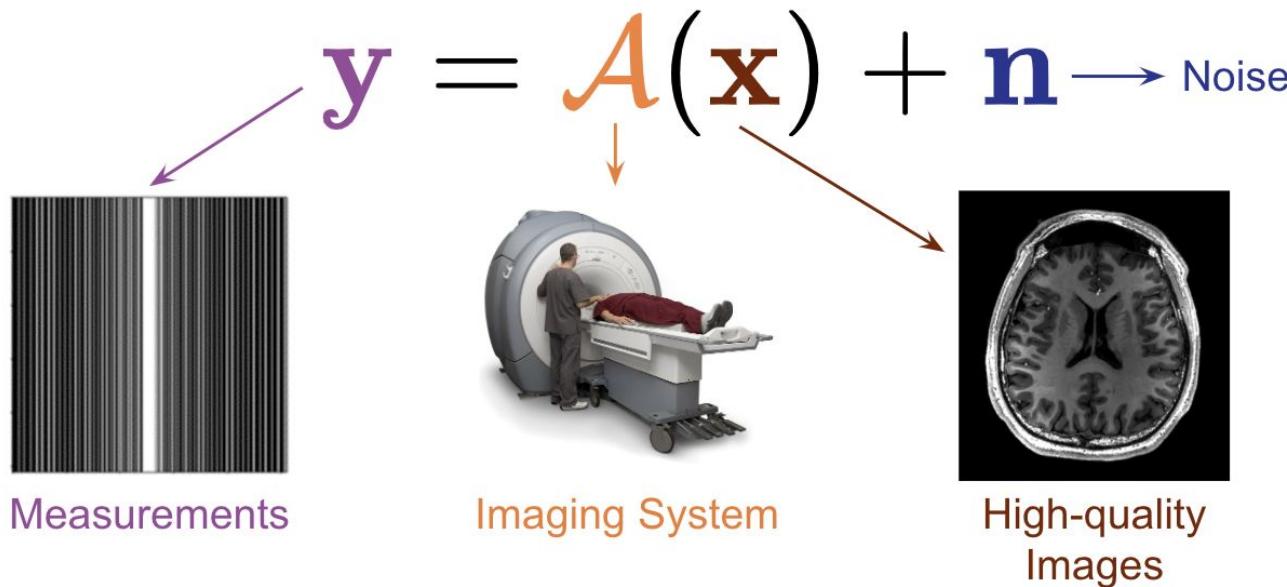


Image credit: <https://mriquestions.com/k-space-trajectories.html>

Inverse Problem in Medical Imaging

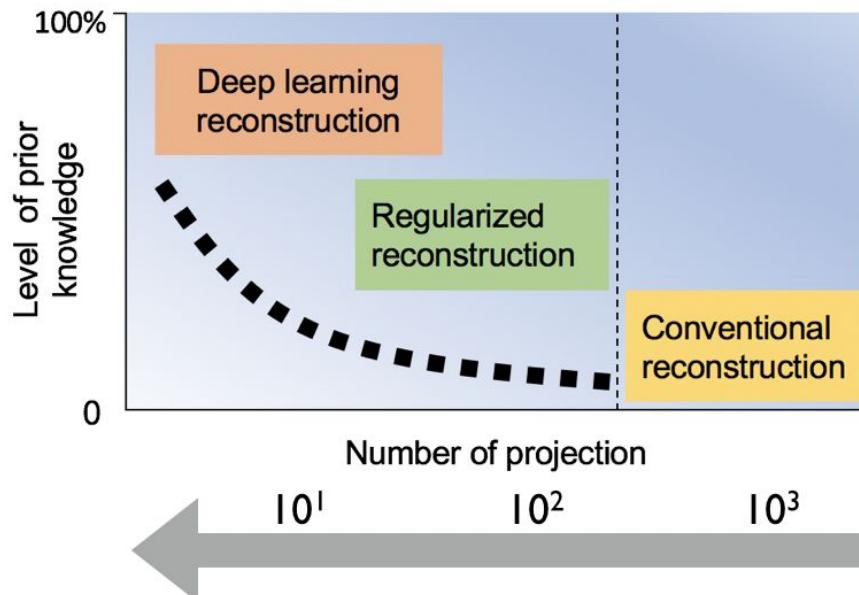
- Medical imaging process can be formulated as:



Inverse problem: reconstruct original image from sparse measurements

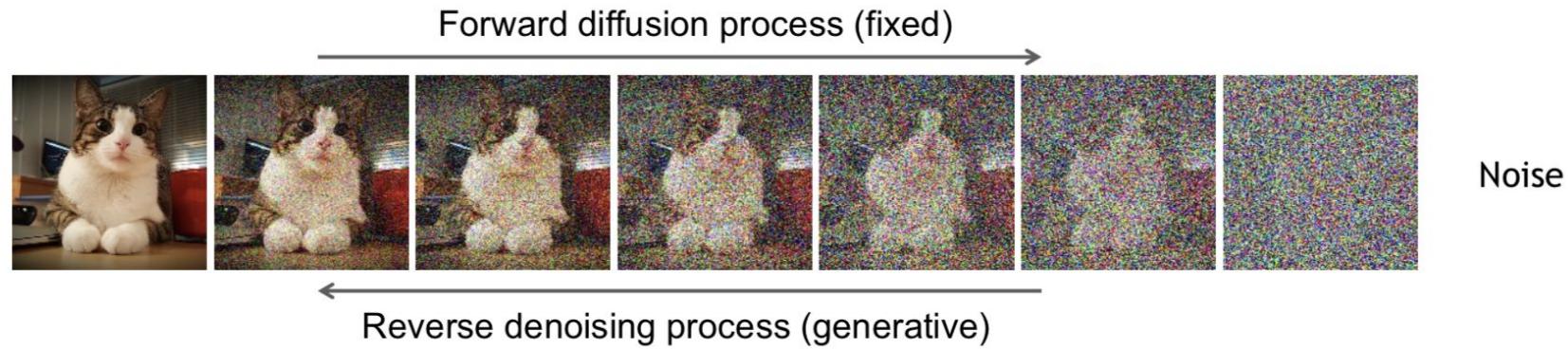
Leverage Prior Knowledge for Solving Ill-posed Inverse Problem

- Conventional reconstruction: require dense sampling
- Regularized reconstruction: sparsity in transformed domain
- Deep learning reconstruction: learned prior from a data-driven way



Diffusion model:
Provide prior knowledge of data distribution for solving inverse problems

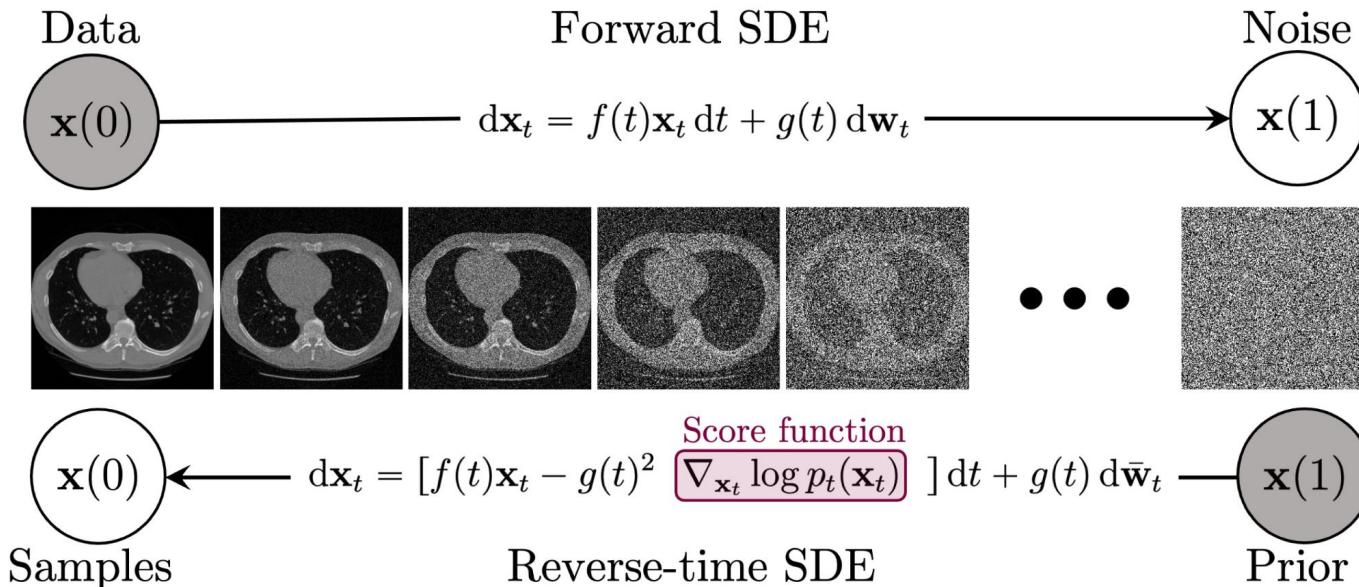
Recap: Diffusion Models



- Forward diffusion process with noise added
- Reverse denoising process for generative diffusion sampling

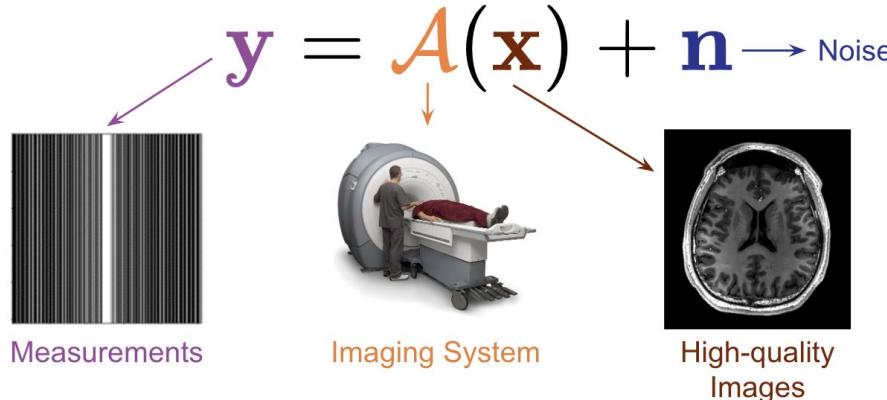
Recap: Score-based Diffusion Models

- Forward diffusion process as stochastic differential equation (SDE)
 - Generative reverse stochastic differential equation



Diffusion models learn data prior by modeling data distribution through unsupervised training

Diffusion Models for Solving Inverse Problems



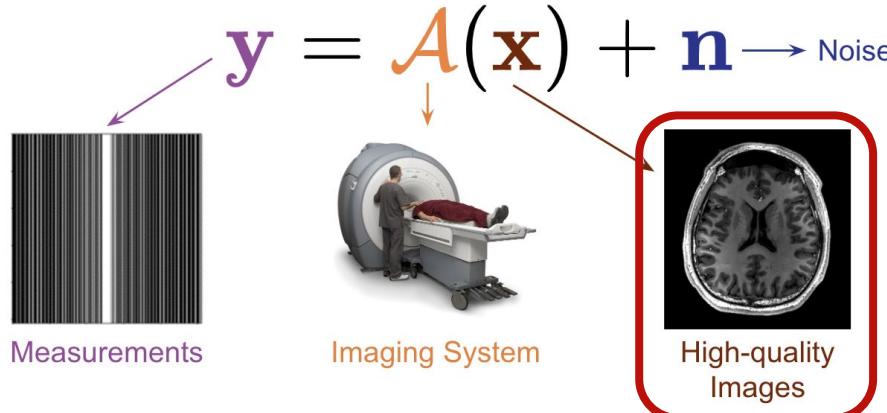
- From Bayes' rule:

$$p(\mathbf{x} | \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} | \mathbf{x}) / \int p(\mathbf{x})p(\mathbf{y} | \mathbf{x}) d\mathbf{x}.$$

- Compute the posterior score function:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x} | \mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{y} | \mathbf{x})$$

Diffusion Models for Solving Inverse Problems



- From Bayes' rule:

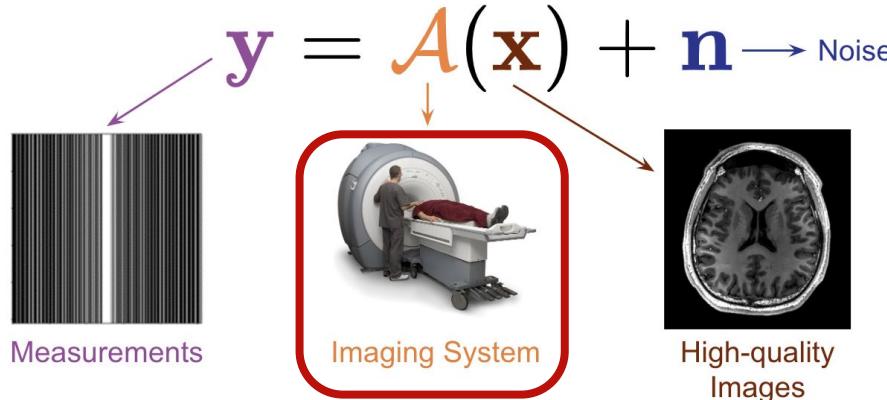
$$p(\mathbf{x} | \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} | \mathbf{x}) / \int p(\mathbf{x})p(\mathbf{y} | \mathbf{x}) d\mathbf{x}.$$

- Compute the posterior score function:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x} | \mathbf{y}) = \boxed{\nabla_{\mathbf{x}} \log p(\mathbf{x})} + \nabla_{\mathbf{x}} \log p(\mathbf{y} | \mathbf{x})$$

Score function of data distribution

Diffusion Models for Solving Inverse Problems



- From Bayes' rule:

$$p(\mathbf{x} | \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} | \mathbf{x}) / \int p(\mathbf{x})p(\mathbf{y} | \mathbf{x}) d\mathbf{x}.$$

- Compute the posterior score function:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x} | \mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \boxed{\nabla_{\mathbf{x}} \log p(\mathbf{y} | \mathbf{x})}$$

Known forward process

Forward Process of Sparse-sampling Medical

- Forward model:

- Physics-based transformation
- Sparse sampling mask of observed measurements

x: image

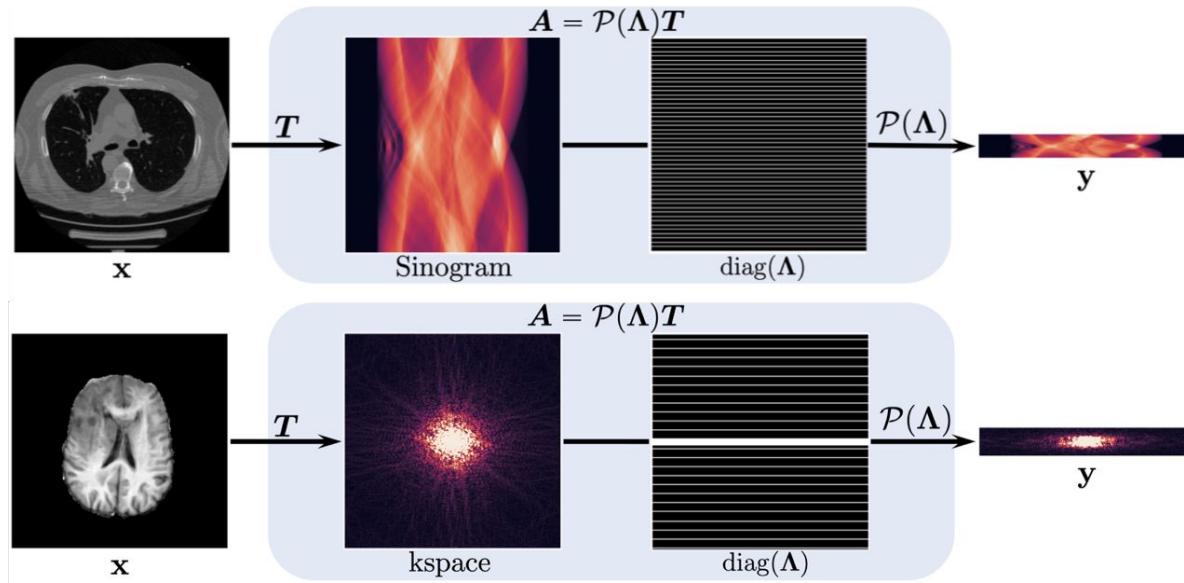
y: observed measurements

A: forward model

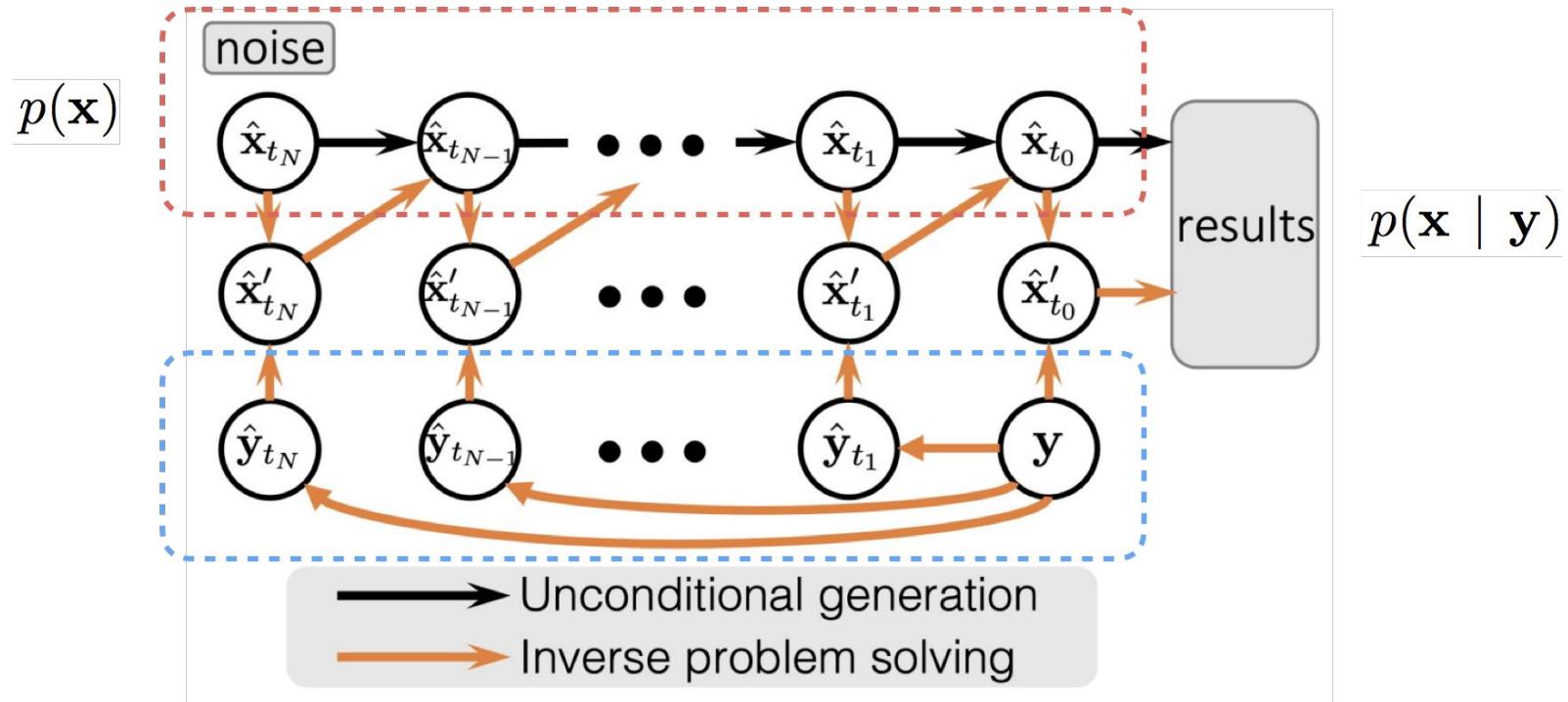
T: transformation

Λ : sampling mask

P: dimension reduction operator



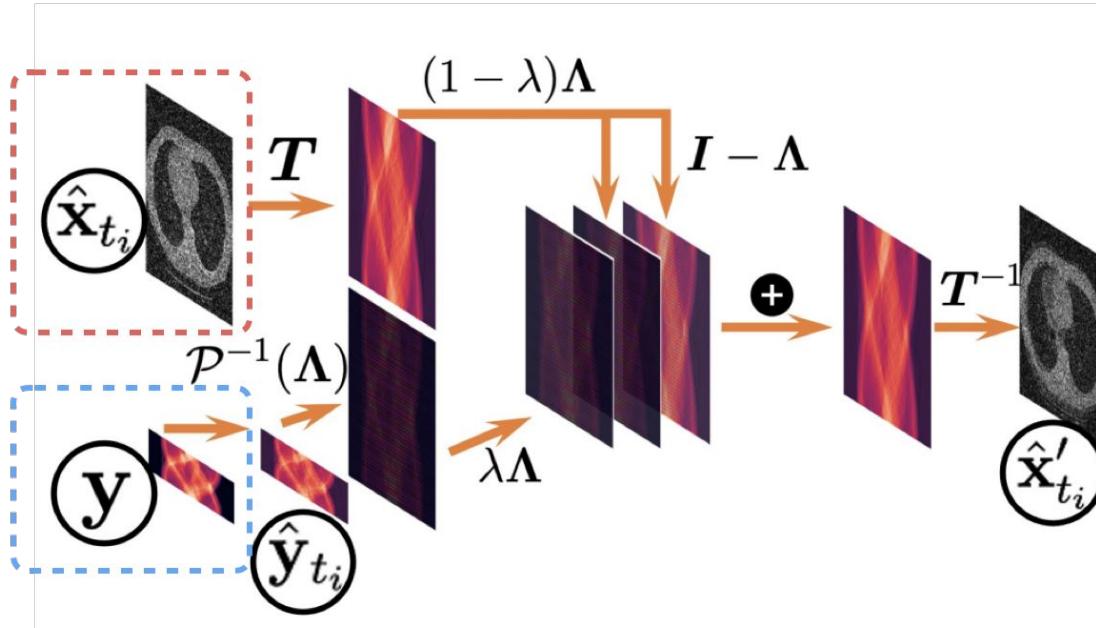
Diffusion Models for Solving Inverse Problems



Posterior sampling consistent with data distribution prior and observed measurements

Optimization Objective

$$\hat{\mathbf{x}}'_{t_i} = \arg \min_{\mathbf{z} \in \mathbb{R}^n} \{(1 - \lambda) \|\mathbf{z} - \hat{\mathbf{x}}_{t_i}\|_T^2 + \min_{\mathbf{u} \in \mathbb{R}^n} \lambda \|\mathbf{z} - \mathbf{u}\|_T^2\} \quad s.t. \quad \mathbf{A}\mathbf{u} = \hat{\mathbf{y}}_{t_i}$$

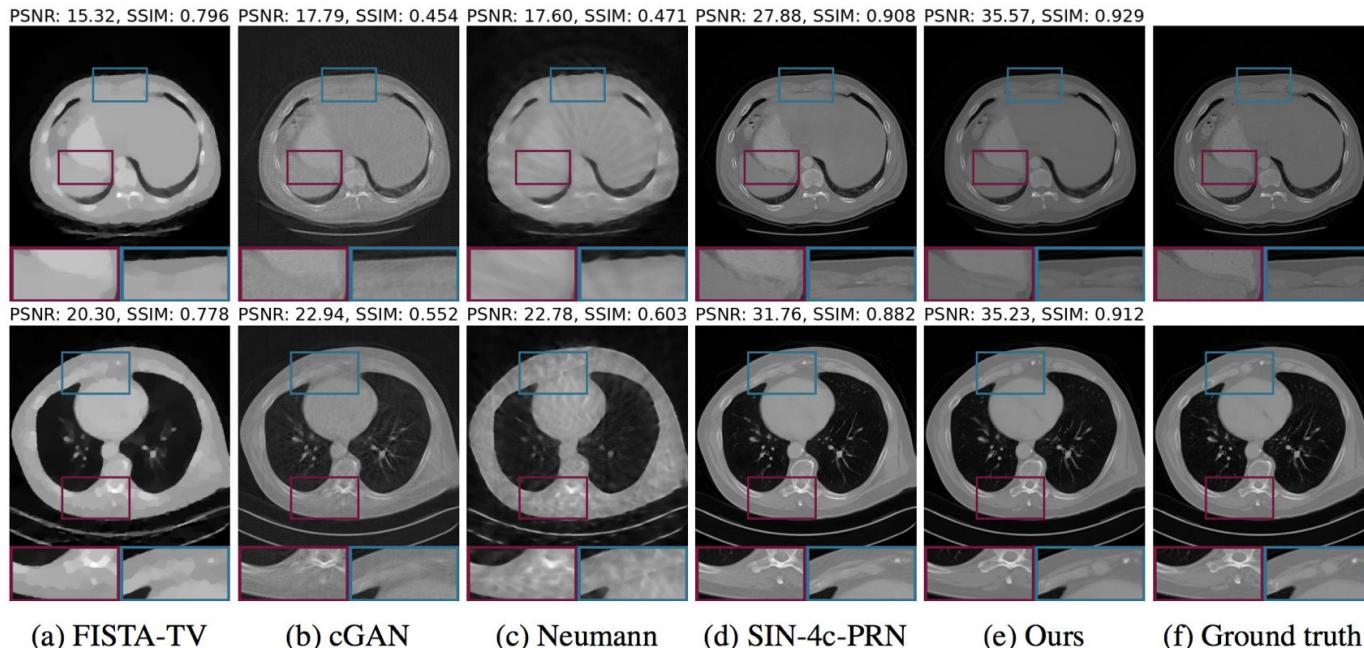


x: image
y: observed measurements
A: forward model
T: transformation
 Λ : sampling mask
P: dimension reduction operator

Posterior sampling consistent with **data distribution prior** and **observed measurements**

Advantages of Diffusion Inverse Solvers

- No assumption of measurements sampling during training
- Comparable or better performance to supervised learning methods



(a) FISTA-TV

(b) cGAN

(c) Neumann

(d) SIN-4c-PRN

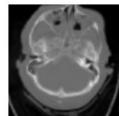
(e) Ours

(f) Ground truth

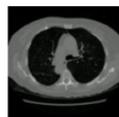
Advantages of Diffusion Inverse Solvers

- Generalizable to solve different inverse problems at inference time
 - Different number of projections in CT reconstruction
 - Different acceleration factors in MRI reconstruction
 - A single training model captures data distribution prior of multi-anatomic sites and multi-modal images

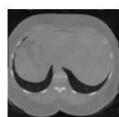
Multiple anatomic sites



Head CT

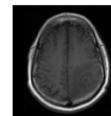


Lung CT

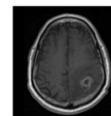


Abdominal CT

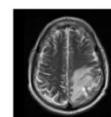
Multiple imaging modalities



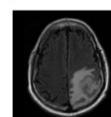
T1 MRI



T1-contrast MRI



T2 MRI

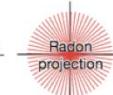


FLAIR MRI

Different sampling processes



Different projections in CT



Different sampling masks in MRI

Diffusion Inverse Solvers in Medical Imaging

- Lots of literatures:
 - Song et al., "Solving Inverse Problems in Medical Imaging with Score-Based Generative Models", ICLR, 2022
 - Jalal et al., "Robust Compressed Sensing MRI with Deep Generative Priors", NeurIPS, 2021
 - Chung and Ye, "Score-based diffusion models for accelerated MRI", Medical Image Analysis, 2022
 - Chung et al., "Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction", CVPR, 2022
 - Peng et al., "Towards performant and reliable undersampled MR reconstruction via diffusion model sampling", arXiv, 2022
 - Xie and Li, "Measurement-conditioned Denoising Diffusion Probabilistic Model for Under-sampled Medical Image Reconstruction", arXiv, 2022
 - Luo et al, "MRI Reconstruction via Data Driven Markov Chain with Joint Uncertainty Estimation", arXiv, 2022
 -

Diffusion Inverse Solvers beyond Medical Imaging

Linear

(a) Inpainting



(c) Gaussian deblur

(b) Super-resolution

(d) Motion deblur



Non-linear

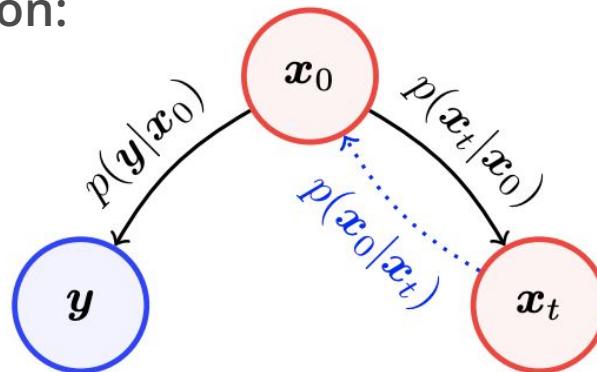
(e) Phase retrieval

(f) Non-uniform deblur



Diffusion Posterior Sampling for General Inverse Problems

Posterior mean estimation:



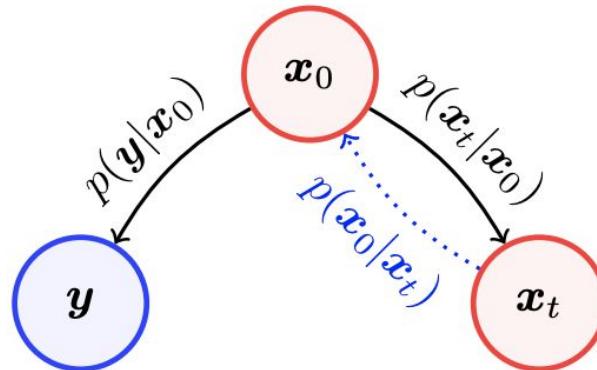
$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \boxed{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y} | \mathbf{x}_t)}$$

$$\hat{\mathbf{x}}_0 := \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t] = \frac{1}{\sqrt{\bar{\alpha}(t)}} (\mathbf{x}_t + (1 - \bar{\alpha}(t)) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t))$$

$$p(\mathbf{y} | \mathbf{x}_t) \simeq p(\mathbf{y} | \hat{\mathbf{x}}_0), \quad \text{where} \quad \hat{\mathbf{x}}_0 := \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t] = \mathbb{E}_{\mathbf{x}_0 \sim p(\mathbf{x}_0 | \mathbf{x}_t)} [\mathbf{x}_0]$$

Diffusion Posterior Sampling for General Inverse Problems

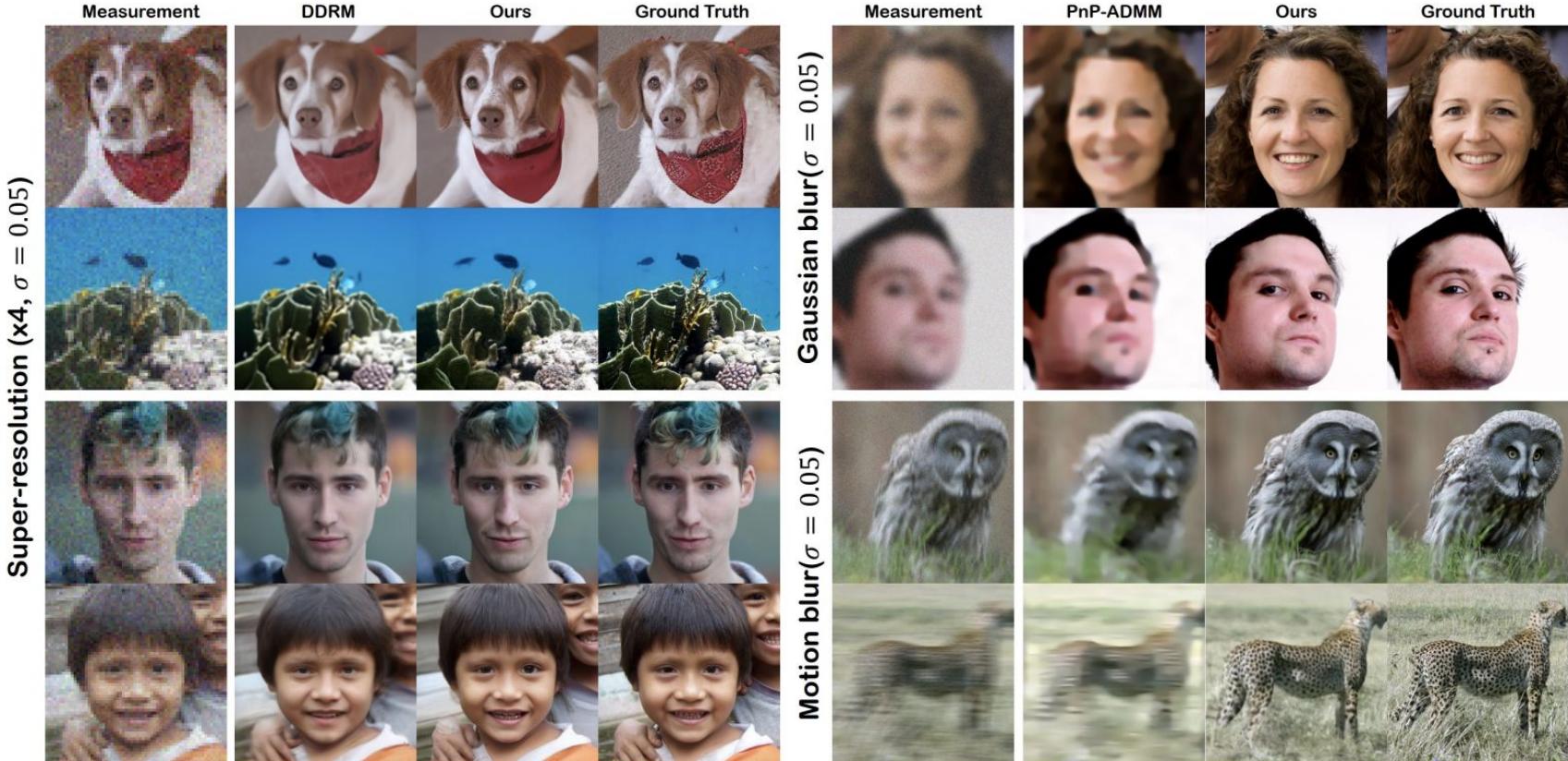
Take backpropagation through network:



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

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t) &\simeq \nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \hat{\mathbf{x}}_0) \\ &\simeq -\frac{1}{\sigma^2} \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0(\mathbf{x}_t))\|_2^2\end{aligned}$$

Diffusion Posterior Sampling for General Inverse Problems



Diffusion Inverse Solvers

Category	Method	SVD	Pseudo inverse	Linear	Gradient
Linear guidance	DDRM	✓	✓	✓	--
	DDNM	✗	✓	✓	--
	ΠGDM	✗	✓	✗	--
General guidance	DPS	✗	✗	✗	✓
	LGD	✗	✗	✗	✓
	DPG	✗	✗	✗	✗
	SCG	✗	✗	✗	✗
	EnKG	✗	✗	✗	✗
Variable-splitting	DiffPIR	✗	✗	✗	✓
	PnP-DM	✗	✗	✗	✓
	DAPS	✗	✗	✗	✓
Variational Bayes	RED-diff	✗	✗	✗	✓
Sequential Monte Carlo	FPS	✗	✗	✓	--
	MCGDiff	✓	✓	✓	--

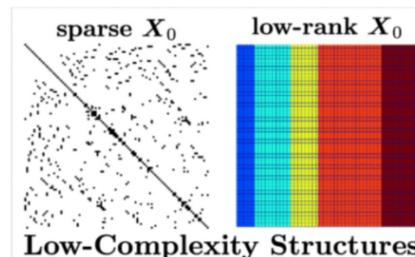
Challenges of Diffusion Inverse Solvers in Scientific Applications

Efficiency



(Scientific)
Applications

Generalizability

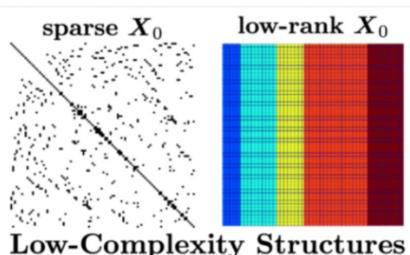


Controllability

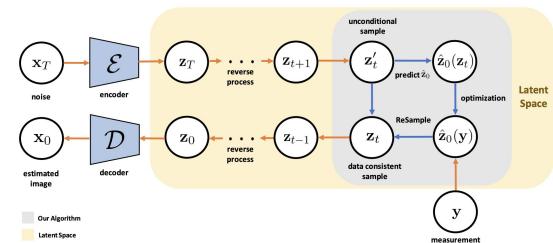
Challenges of Diffusion Inverse Solvers in Scientific Applications

Generalizability

(Scientific)
Applications



Efficiency



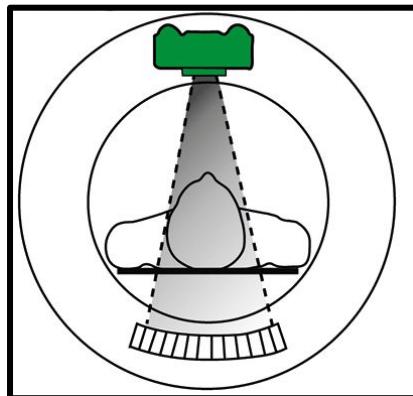
Controllability

Limitations of Diffusion Inverse Solvers

- Requires large-scale training data
 - 50k ~ 300k images
- Slow training and sampling process
 - ~1000 sampling steps
- Expensive GPU memory cost for high-dimensional data
 - Most models only apply to 2D images
-

Limitations of Diffusion Inverse Solvers

- Real-world applications is mostly about **high-dimensional and high-resolution** data!
 - 3D volumetric CT images with standard resolution of 512^3



Projections

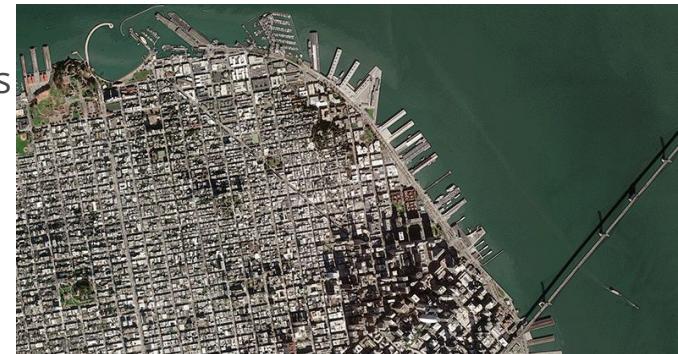


CT volumetric image

Limitations of Diffusion Inverse Solvers

- Real-world applications is mostly about **high-dimensional and high-resolution data!**

- 3D volumetric CT images with standard resolution of 512^3
- 4D MRI images with time sequence
- High-resolution satellite images with thousands of pixels
-

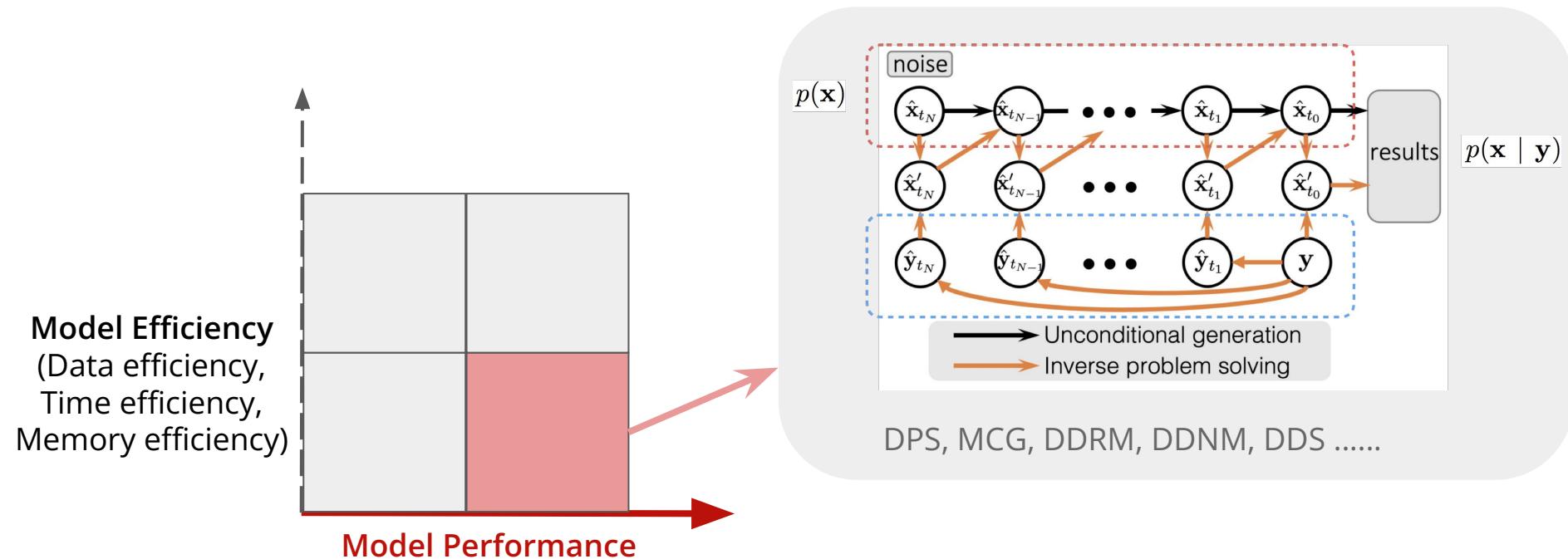


How to enable diffusion model for solving large-scale inverse problems?



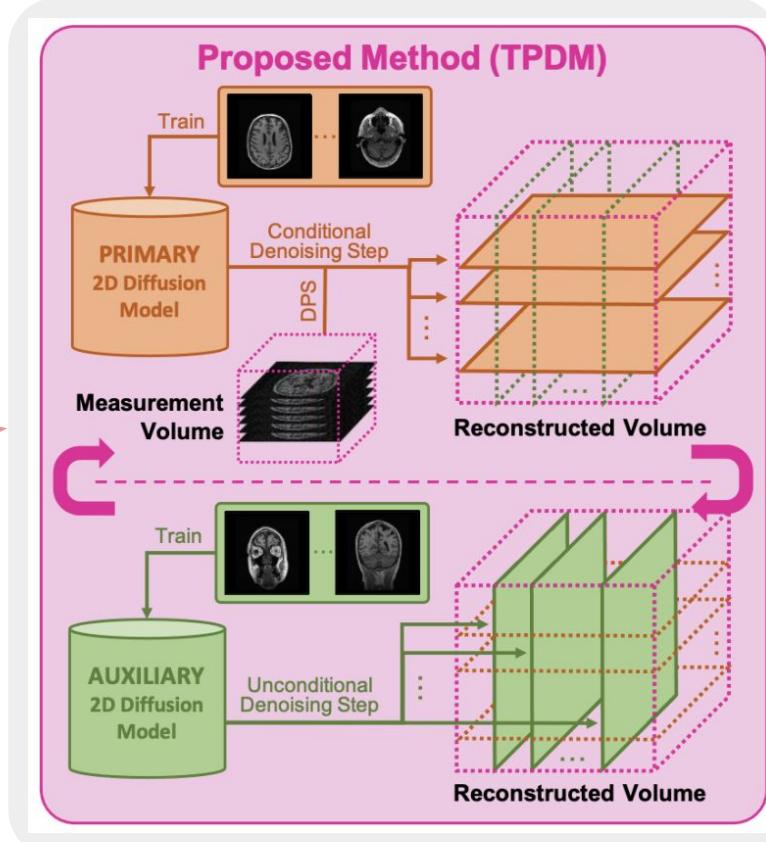
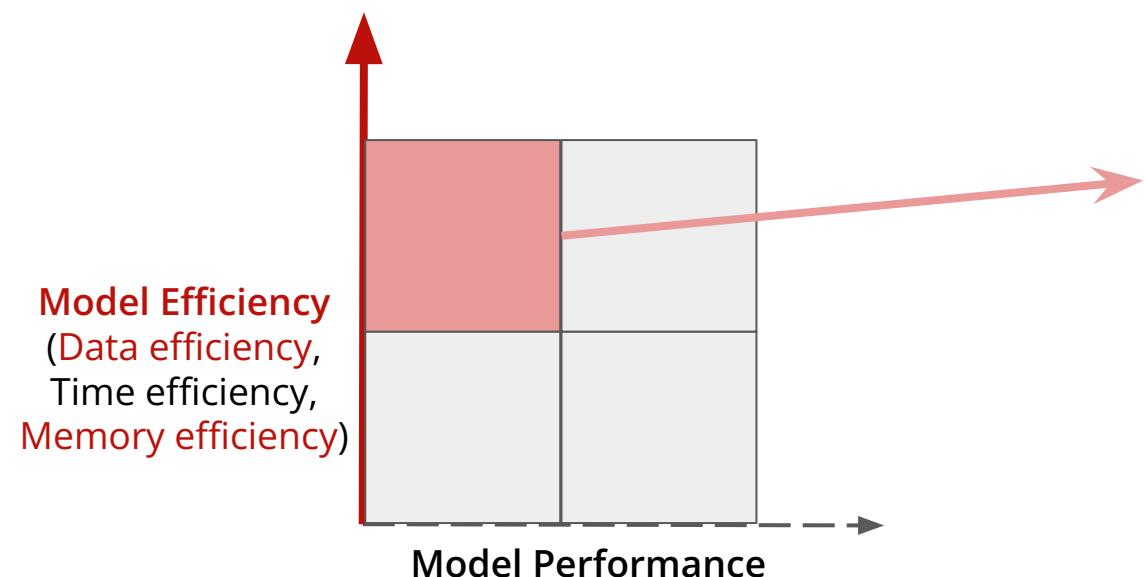
Enhance Diffusion Model Efficiency

- Focus on improving model performance
- Achieve impressive results for 2D images



Enhance Diffusion Model Efficiency

- Improve data and memory efficiency for tackling high-dimensional data
 - Solve 3D imaging problems by using 2D diffusion priors



DiffusionMBIR: Model-based Iterative

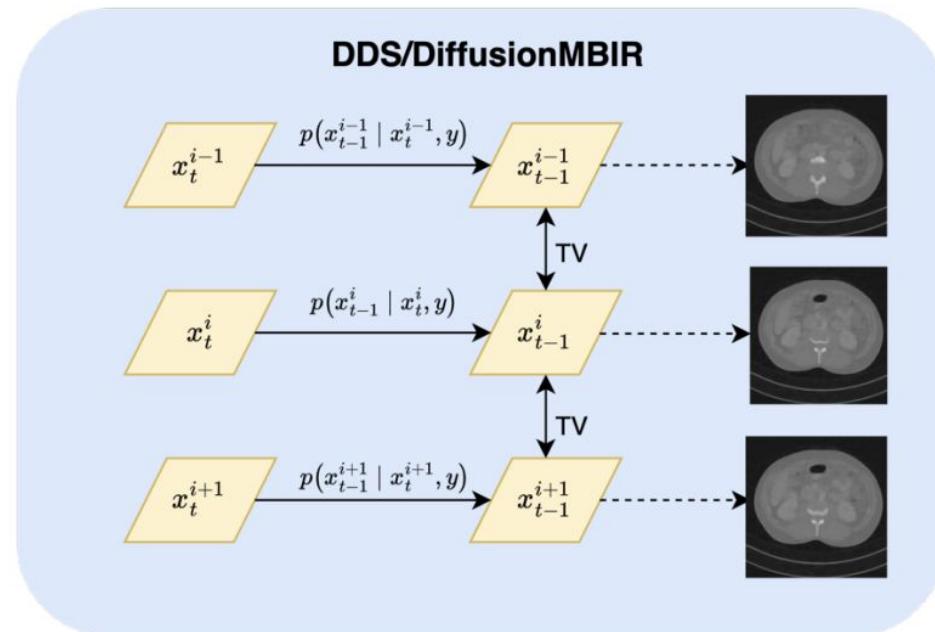
- Solve 3D imaging problems by using 2D diffusion priors
- Regularized objective:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \|\mathbf{D}_z \mathbf{x}\|_1,$$

- Iterative solver:
 - Parallel denoising for each slice
 - Z-directional TV prior to impose consistency

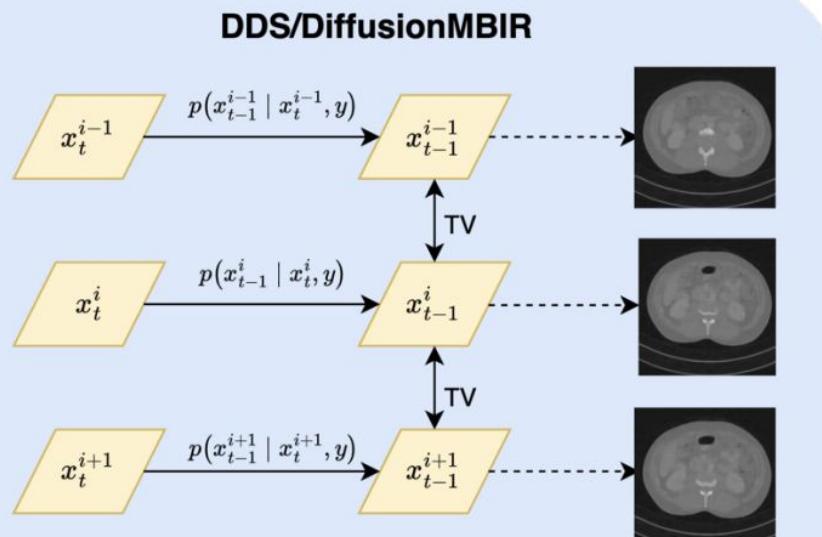
$$\mathbf{x}'_{i-1} \leftarrow \text{Solve}(\mathbf{x}_i, \mathbf{s}_{\theta^*}),$$

$$\mathbf{x}_{i-1} \leftarrow \operatorname{argmin}_{\mathbf{x}'_{i-1}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}'_{i-1}\|_2^2 + \|\mathbf{D}_z \mathbf{x}'_{i-1}\|_1.$$



Slice-by-slice Reconstruction with 2D Diffusion Priors

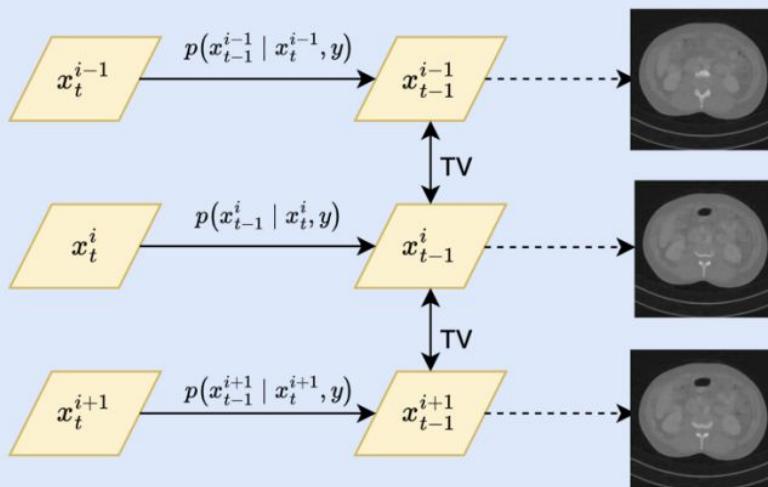
- Limitation:
 - Rely on additional regularization or impractical assumption
 - Compromise model performance on cross-slice consistency
 - Increase inference time (TPDM: 24 to 36 hours per volume!)



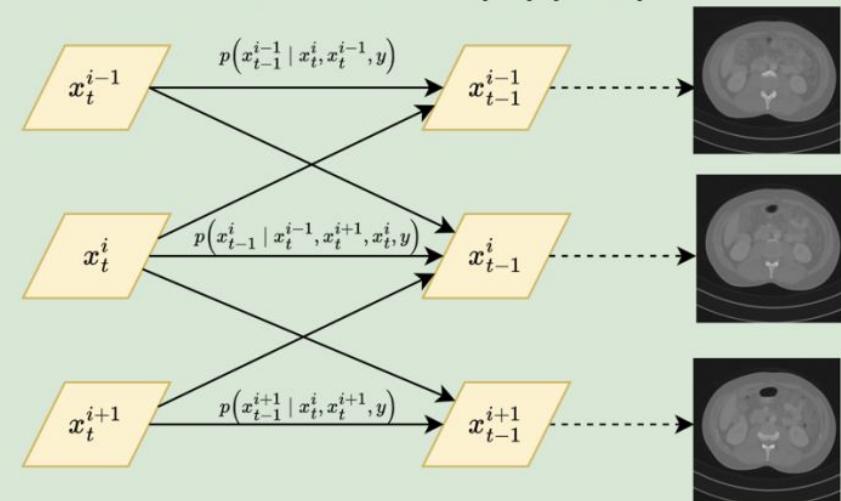
DiffusionBlend: Position-aware Diffusion Score Blending

- Learn 3D-patch image prior by incorporating cross-slice dependency
- Blend learned diffusion score between groups of slices

DDS/DiffusionMBIR

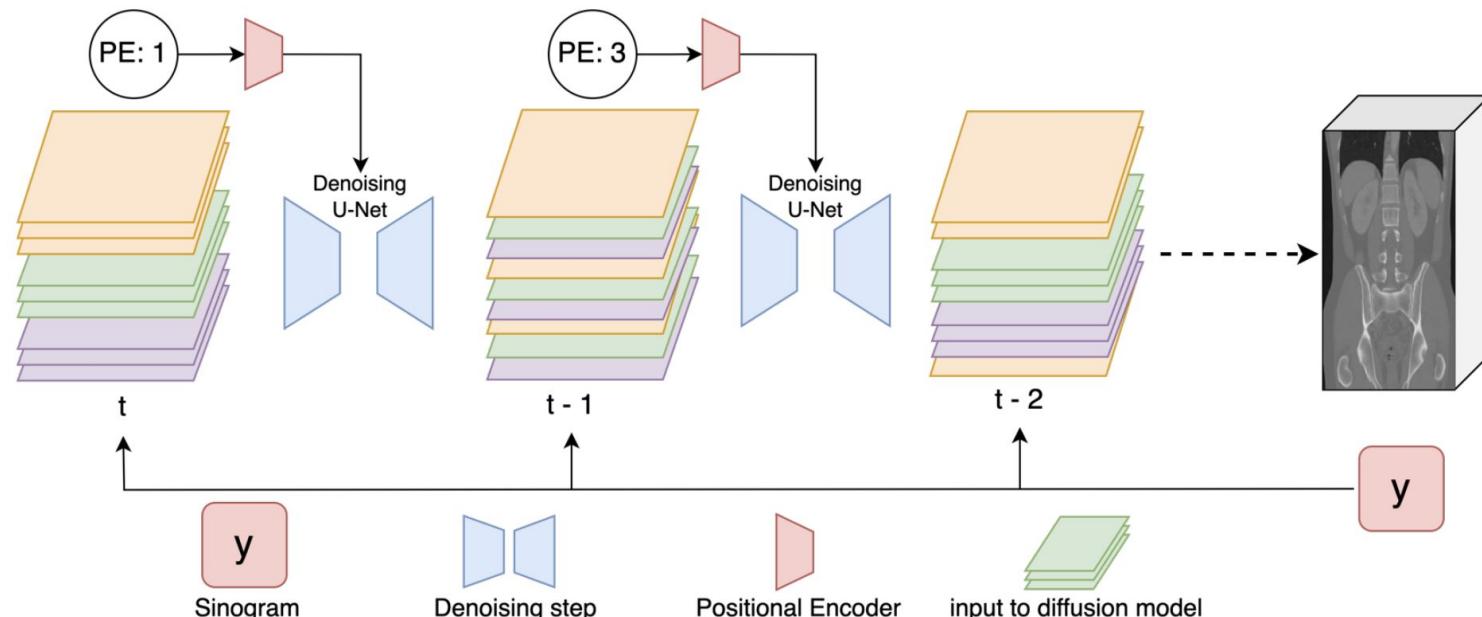


DiffusionBlend(++) (Ours)



DiffusionBlend: Position-aware Diffusion Score Blending

- Enforce cross-slice consistency for 3D volume without hand-crafted regularization
- Positional encoding: separation between the slices



DiffusionBlend: Position-aware Diffusion Score Blending

- Enable 3D CT reconstruction with high-dimensional 3D image ($256 \times 256 \times 500$)
- Enhance cross-slice consistency on z-axis



FBP

DDS

DiffusionBlend++

Ground Truth

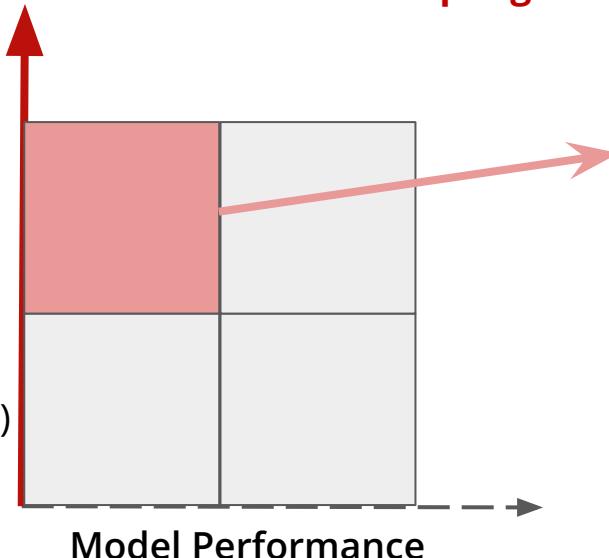
DiffusionBlend: Position-aware Diffusion Score Blending

- Achieve SOTA in DIS methods for (ultra-)sparse-view CT reconstruction

Method	Sparse-View CT Reconstruction on AAPM						Sparse-View CT Reconstruction on LIDC					
	8 views		6 views		4 views		8 Views		6 Views		4 Views	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
FBP	14.64	0.325	13.18	0.268	11.16	0.236	14.78	0.206	14.10	0.181	13.10	0.165
FBP-UNet	27.34	0.878	25.12	0.827	24.10	0.810	28.87	0.858	26.59	0.793	25.37	0.744
DiffusionMBIR	29.86	0.908	28.12	0.875	25.68	0.843	33.29	0.922	31.69	0.903	29.21	0.868
TPDM	-	-	-	-	-	-	28.12	0.833	25.78	0.804	22.29	0.735
DDS 2D	33.64	0.950	32.33	0.939	30.25	0.916	31.60	0.898	29.99	0.871	28.03	0.830
DDS	33.97	0.934	32.95	0.879	30.89	0.932	32.51	0.920	30.83	0.898	27.61	0.828
DiffusionBlend (Ours)	36.45	0.958	35.23	0.952	33.98	0.944	34.47	0.934	31.48	0.908	28.24	0.859
DiffusionBlend++ (Ours)	37.87	0.968	36.66	0.963	34.27	0.955	35.66	0.947	33.97	0.935	31.38	0.913

Enhance Diffusion Model Efficiency

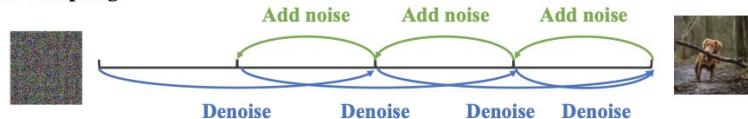
- Improve time efficiency
 - Enable sampling as few as 1~4 steps for image generation
 - **How to improve time efficiency for high-dimensional data sampling?**



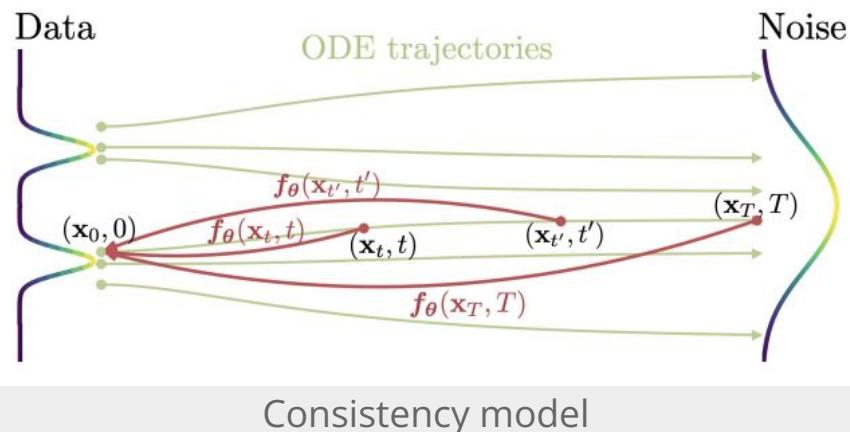
Deterministic sampling:



Stochastic sampling:



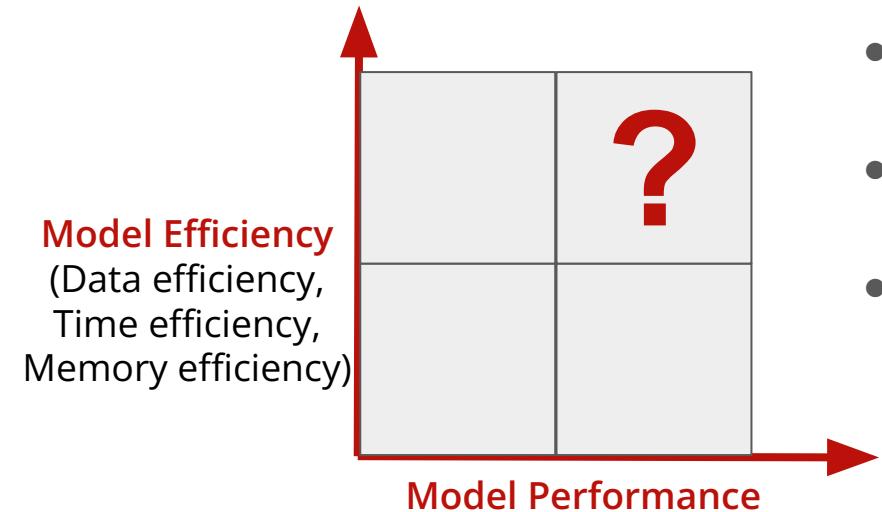
Distillation of diffusion model



Enable Diffusion Model for High-dimensional High-resolution Data

Open Question:

Can we achieve better model efficiency while maintaining model performance?



- Improve data efficiency:
 - Require less training data
- Improve memory efficiency:
 - Reduce memory cost
- Improve time efficiency:
 - Faster training/sampling process

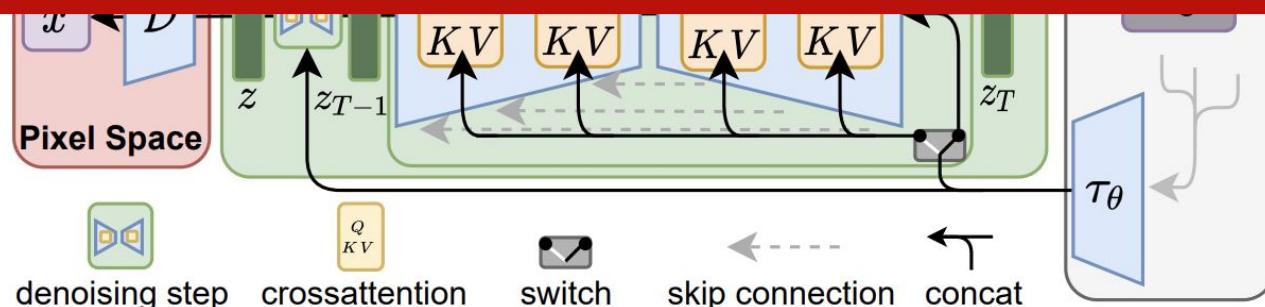
Latent Diffusion Model (LDM)

- Train diffusion model in low-dimensional latent space
 - Encoder-decoder models project data to low-dimensional latent space
 - Enable diffusion model training on limited computational resources while retaining quality and flexibility

Can the efficient LDMs be used as prior for solving inverse problems?

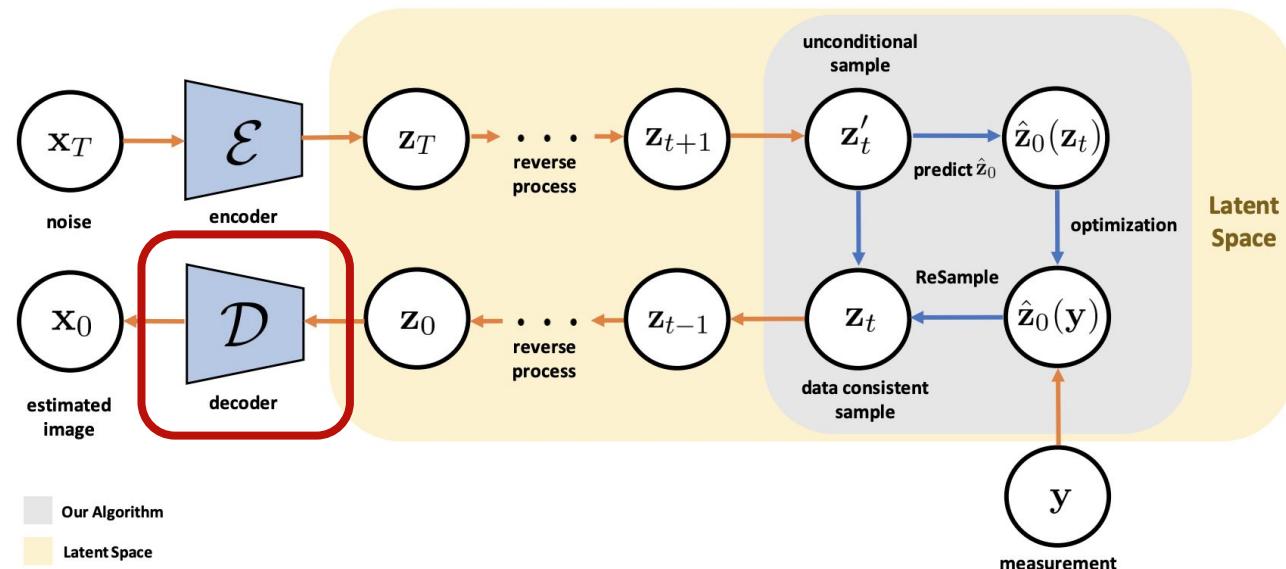


Challenge: How to guarantee data consistency with LDM prior?



Challenge: Guarantee Data Consistency with LDM

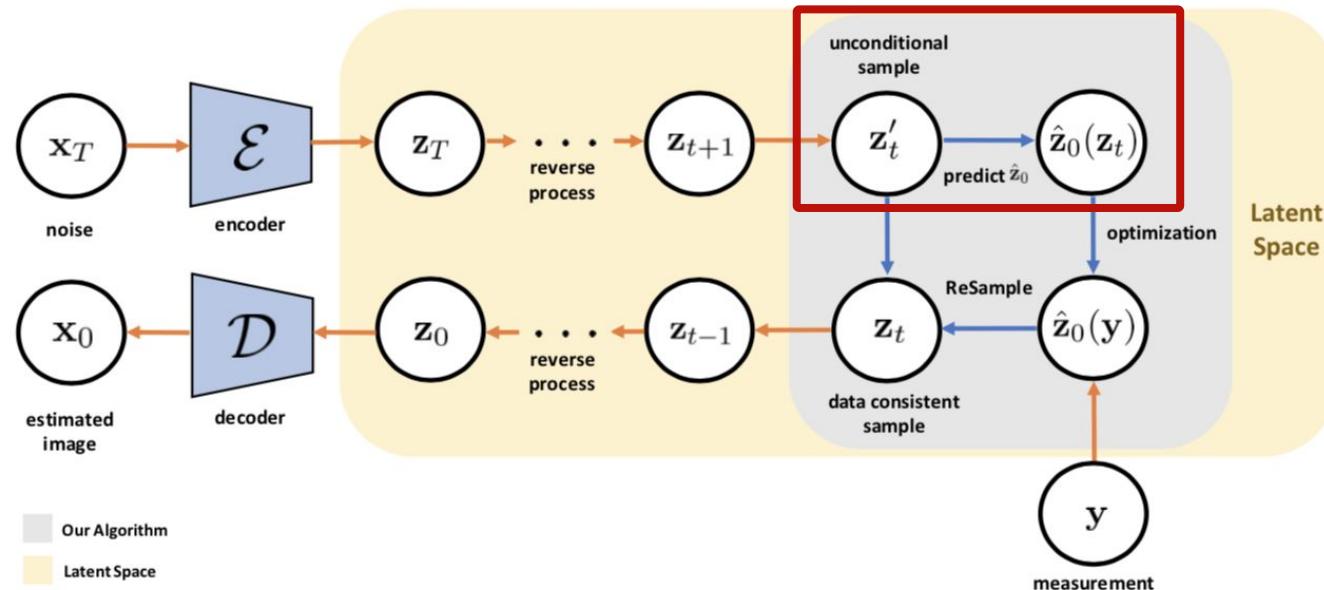
- Non-linearity from pretrained decoder
- ReSample: solve **general (linear and nonlinear)** inverse problems with **pre-trained LDMs**



ReSample

- Posterior mean estimation:

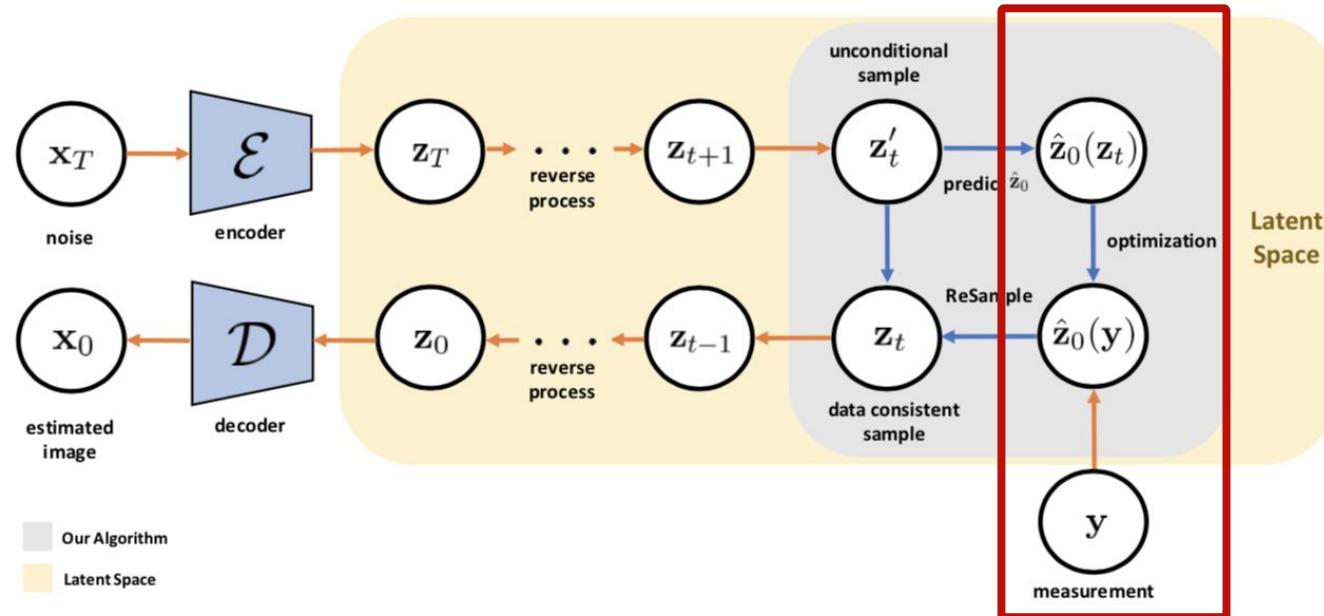
$$\hat{\mathbf{z}}_0(\mathbf{z}_t) = \mathbb{E}[\mathbf{z}_0 | \mathbf{z}_t] = \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{z}_t + (1 - \bar{\alpha}_t) \nabla \log p(\mathbf{z}_t))$$



ReSample

- Optimization objective:

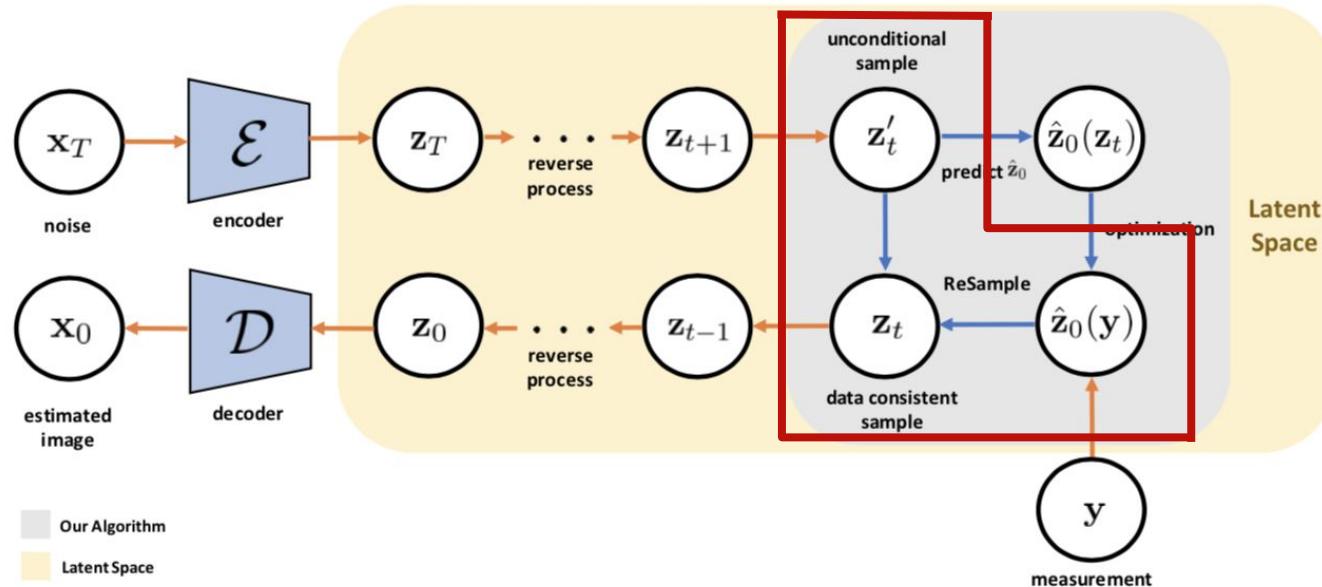
$$\hat{\mathbf{z}}_0(\mathbf{y}) = \arg \min_{\mathbf{z}} \left(\frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}))\|_2^2 \right)$$



ReSample

- Stochastic resampling combines **prior consistency** and **data consistency**

$$\mathbf{z}_t = \text{StochasticResample}(\hat{\mathbf{z}}_0(\mathbf{y}), \mathbf{z}'_t, \gamma)$$



Proposition 1 (Stochastic Encoding). *Since the sample $\hat{\mathbf{z}}_t$ given $\hat{\mathbf{z}}_0(\mathbf{y})$ and measurement \mathbf{y} is conditionally independent of \mathbf{y} , we have that*

$$p(\hat{\mathbf{z}}_t | \hat{\mathbf{z}}_0(\mathbf{y}), \mathbf{y}) = p(\hat{\mathbf{z}}_t | \hat{\mathbf{z}}_0(\mathbf{y})) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \hat{\mathbf{z}}_0(\mathbf{y}), (1 - \bar{\alpha}_t) \mathbf{I}).$$

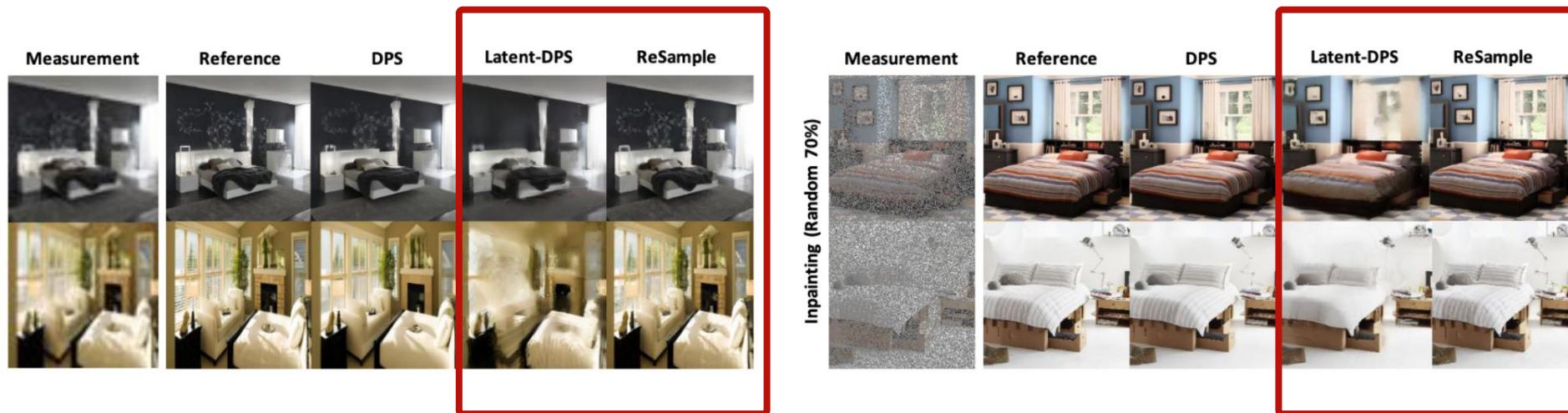
Proposition 2 (Stochastic Resampling). *Suppose that $p(\mathbf{z}'_t | \hat{\mathbf{z}}_t, \hat{\mathbf{z}}_0(\mathbf{y}), \mathbf{y})$ is normally distributed such that $p(\mathbf{z}'_t | \hat{\mathbf{z}}_t, \hat{\mathbf{z}}_0(\mathbf{y}), \mathbf{y}) = \mathcal{N}(\mu_t, \sigma_t^2)$. If we let $p(\hat{\mathbf{z}}_t | \hat{\mathbf{z}}_0(\mathbf{y}), \mathbf{y})$ be a prior for μ_t , then the posterior distribution $p(\hat{\mathbf{z}}_t | \mathbf{z}'_t, \hat{\mathbf{z}}_0(\mathbf{y}), \mathbf{y})$ is given by*

$$p(\hat{\mathbf{z}}_t | \mathbf{z}'_t, \hat{\mathbf{z}}_0(\mathbf{y}), \mathbf{y}) = \mathcal{N}\left(\frac{\sigma_t^2 \sqrt{\bar{\alpha}_t} \hat{\mathbf{z}}_0(\mathbf{y}) + (1 - \bar{\alpha}_t) \mathbf{z}'_t}{\sigma_t^2 + (1 - \bar{\alpha}_t)}, \frac{\sigma_t^2 (1 - \bar{\alpha}_t)}{\sigma_t^2 + (1 - \bar{\alpha}_t)} \mathbf{I}\right).$$

Hyperparameter controlling the tradeoff between data consistency and prior consistency

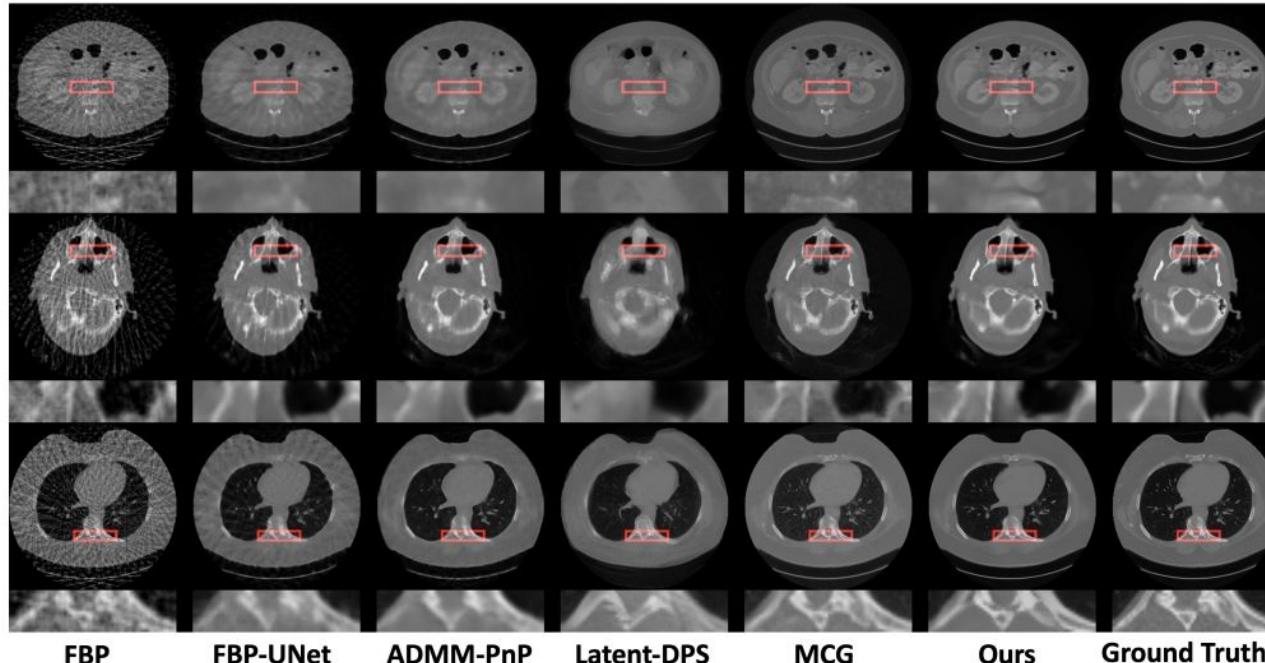
ReSample

- Results in natural images:
 - Tasks: super resolution, inpainting, Gaussian deblurring, non-linear deblurring
 - Datasets: FFHQ dataset, CelebA-HQ dataset, LSUN-Bedroom dataset



ReSample

- Data efficiency: use pre-trained encoder / decoder from natural images
- Fine-tune latent diffusion models using **2000 CT images**

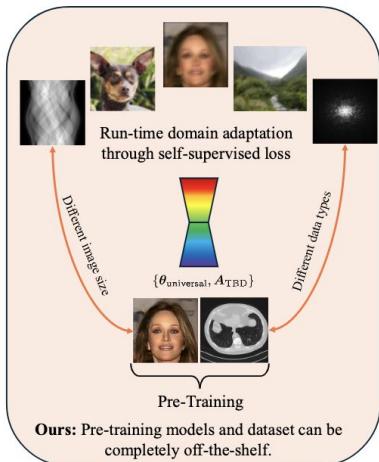


ReSample

- Achieve SOTA results for sparse-view CT image reconstruction using **low-dimensional prior in latent space**
- Outperform pixel-space diffusion models trained with more data samples

Method	Abdominal		Head		Chest	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Latent-DPS	26.80 ± 1.09	0.870 ± 0.026	28.64 ± 5.38	0.893 ± 0.058	25.67 ± 1.14	0.822 ± 0.033
MCG (Chung et al., 2022)	29.41 ± 3.14	0.857 ± 0.041	28.28 ± 3.08	0.795 ± 0.116	27.92 ± 2.48	0.842 ± 0.036
DPS (Chung et al., 2023a)	27.33 ± 2.68	0.715 ± 0.031	24.51 ± 2.77	0.665 ± 0.058	24.73 ± 1.84	0.682 ± 0.113
PnP-UNet (Gilton et al., 2021)	<u>32.84 ± 1.29</u>	<u>0.942 ± 0.008</u>	<u>33.45 ± 3.25</u>	<u>0.945 ± 0.023</u>	29.67 ± 1.14	<u>0.891 ± 0.011</u>
FBP	26.29 ± 1.24	0.727 ± 0.036	26.71 ± 5.02	0.725 ± 0.106	24.12 ± 1.14	0.655 ± 0.033
FBP-UNet (Jin et al., 2017)	32.77 ± 1.21	0.937 ± 0.013	31.95 ± 3.32	0.917 ± 0.048	29.78 ± 1.12	0.885 ± 0.016
ReSample (Ours)	35.91 ± 1.22	0.965 ± 0.007	37.82 ± 5.31	0.978 ± 0.014	31.72 ± 0.912	0.922 ± 0.011

Challenges of Diffusion Inverse Solvers in Scientific Applications

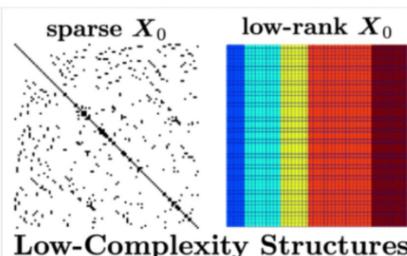


Efficiency



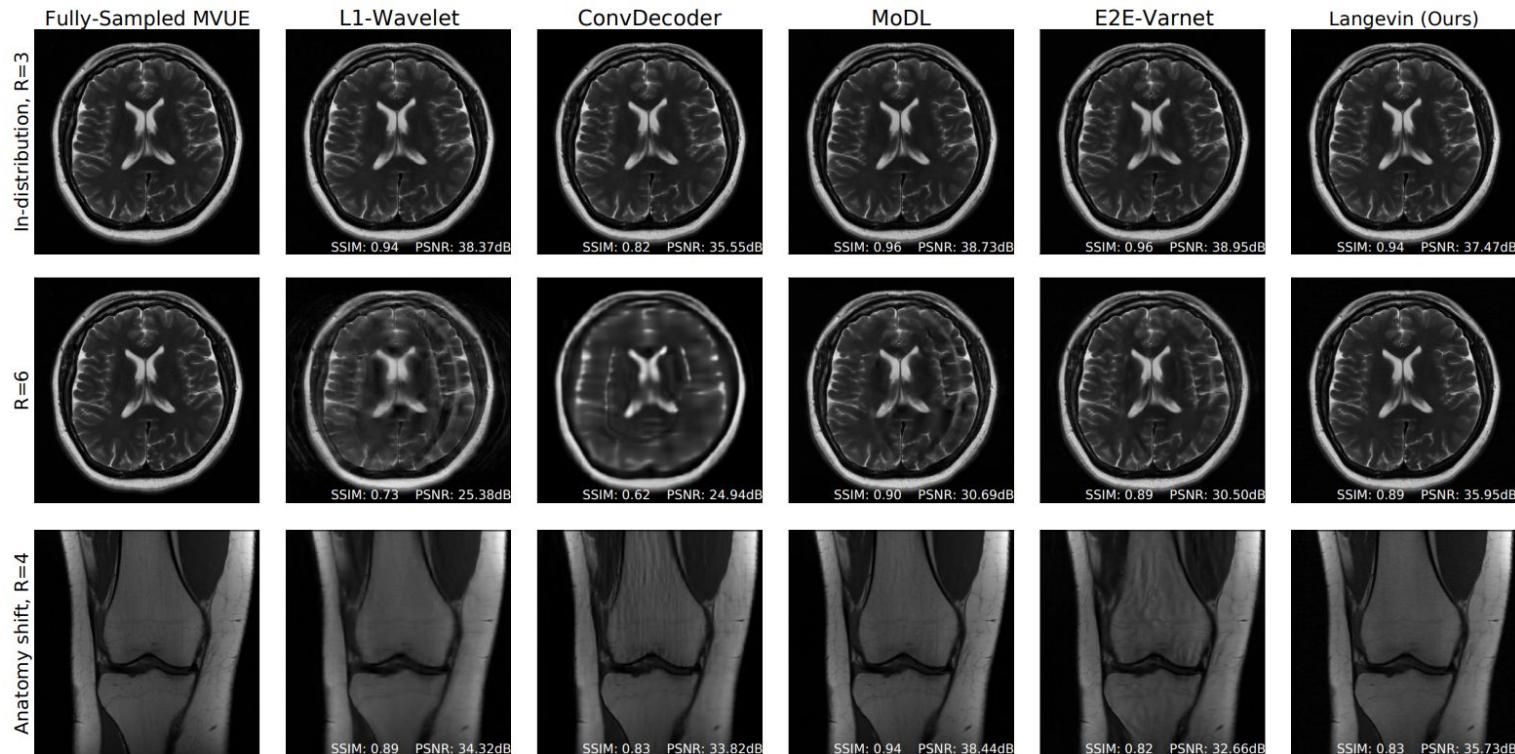
(Scientific)
Applications

Generalizability



Controllability

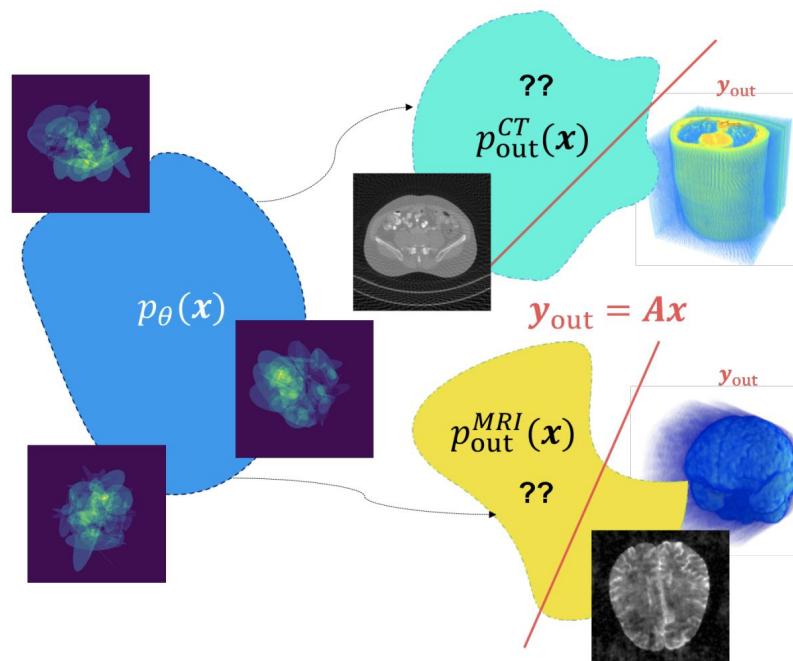
Generalizability of Diffusion Priors



How generalizable is the learned diffusion prior?

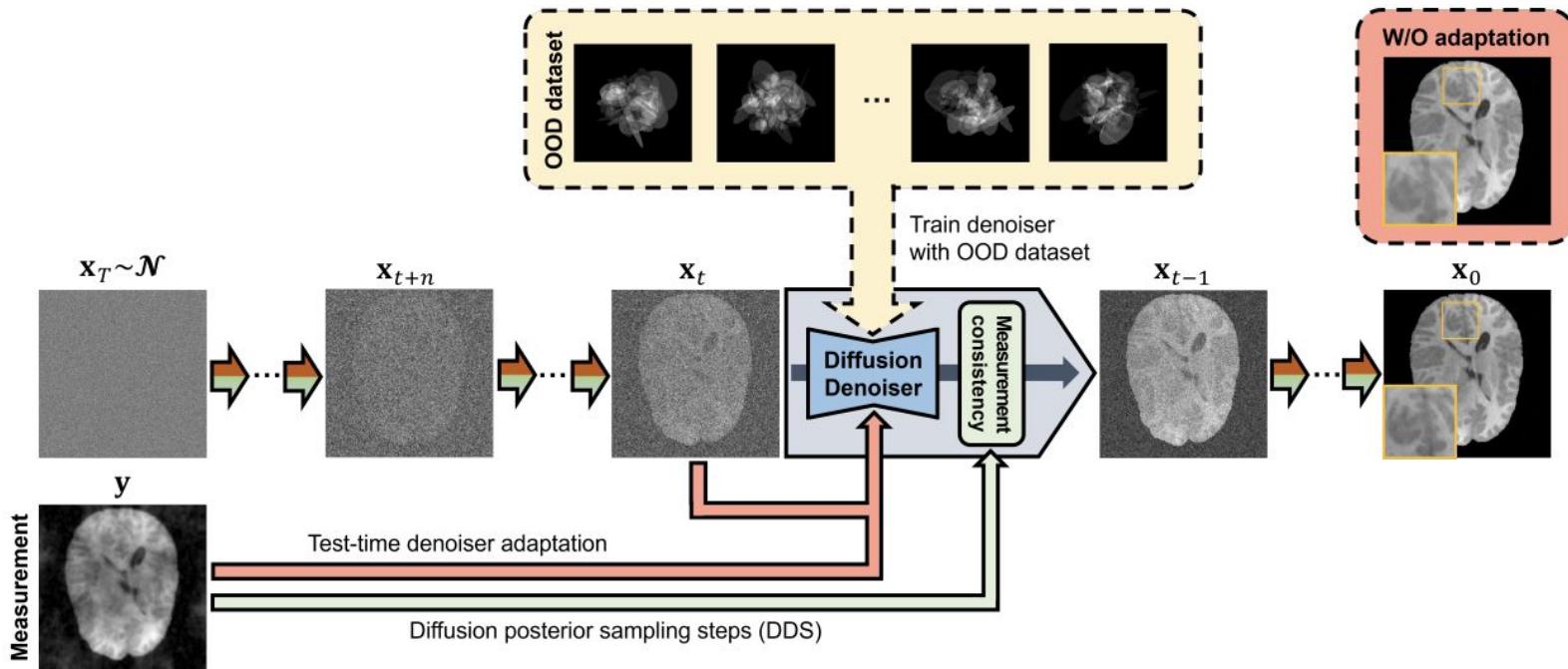
Out-of-Distribution (OOD) Inverse Problems

- Pretrained diffusion model learns prior
- At test-time, obtain measurements from unknown OOD distributions



Steerable Conditional Diffusion for OOD Adaptation

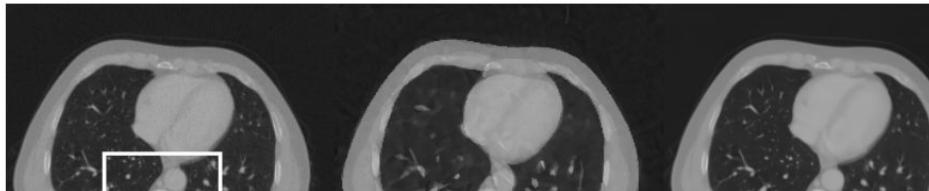
- Self-supervised loss for LoRA adaptation
- Enables DIS model adaptation for OOD tasks using a single corrupted measured data



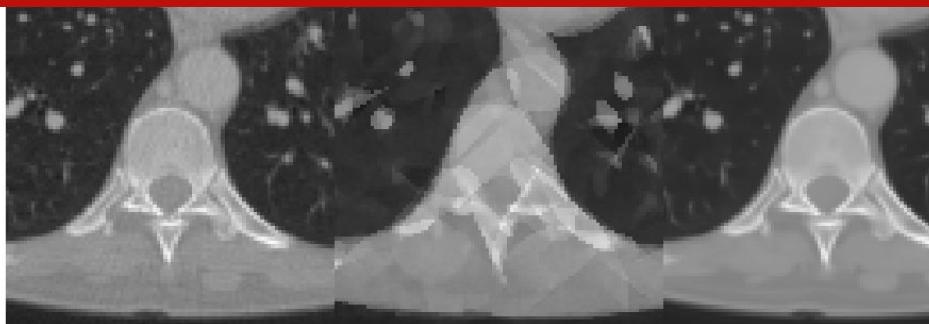
Steerable Conditional Diffusion for OOD Adaptation

- Training: synthetic ellipses images
- Test: computed tomography images

Ground Truth OOD Diffusion Prior Correct Diffusion Prior

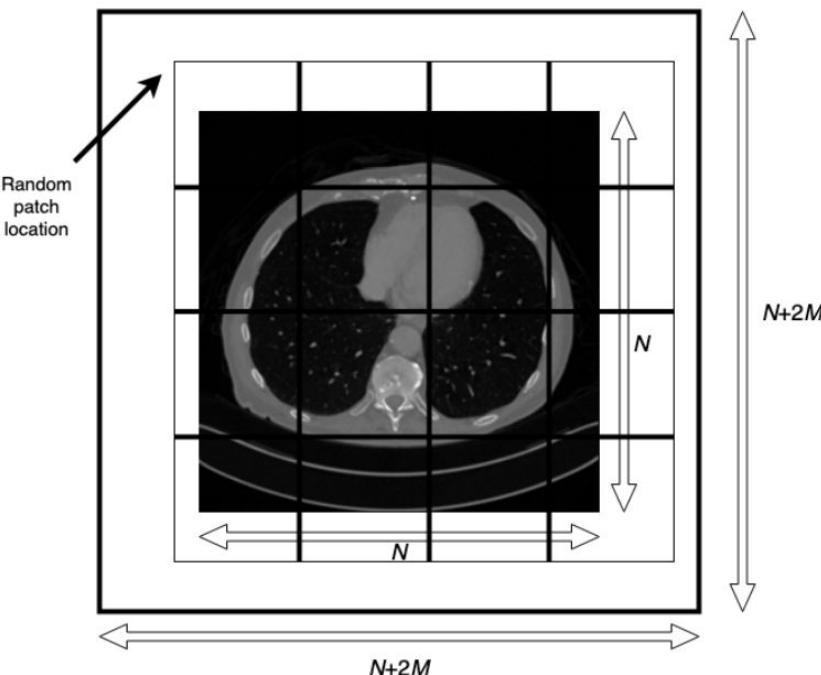


Would low-dimensional image patches preserve more transferable features in OOD scenarios?



PaDIS: Patch-based Diffusion Inverse Solver

- Learn score function of whole image through scores of patches with positional encoding
- Zero-padding boundary region to enable shifting patch locations

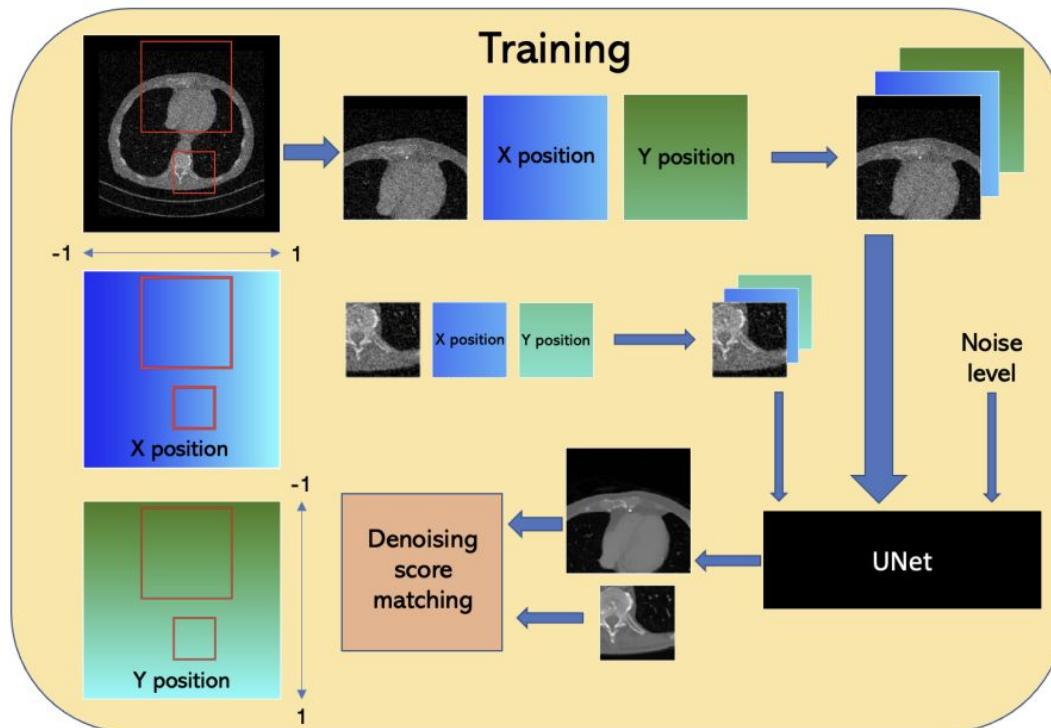


$$p(\mathbf{x}) = \prod_{i,j=1}^{i,j=M} p_{i,j,B}(\mathbf{x}_{i,j,B}) \prod_{r=1}^{(k+1)^2} p_{i,j,r}(\mathbf{x}_{i,j,r})/Z,$$

$$\nabla \log p(\mathbf{x}) = \sum_{i,j=1}^{i,j=M} \left(\mathbf{s}_{i,j,B}(\mathbf{x}_{i,j,B}) + \sum_{r=1}^{(k+1)^2} \mathbf{s}_{i,j,r}(\mathbf{x}_{i,j,r}) \right)$$

PaDIS: Patch-based Diffusion Inverse Solver

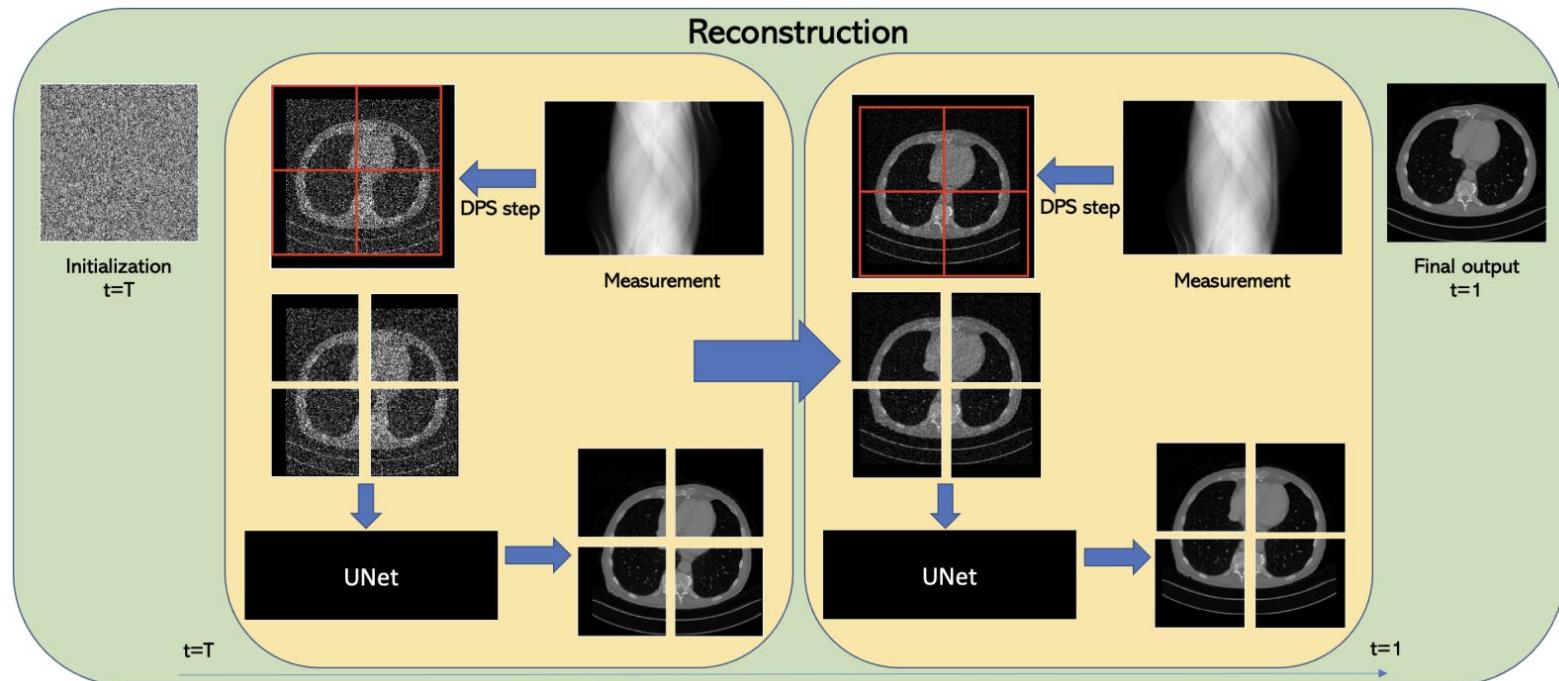
- Improve training efficiency



- **Memory efficiency**
 - Only input image patches
- **Data efficiency**
 - 2000~3000 training samples
- **Training Time efficiency**
 - Patch image model: ~12 hours
 - Whole image model: 24~36 hours

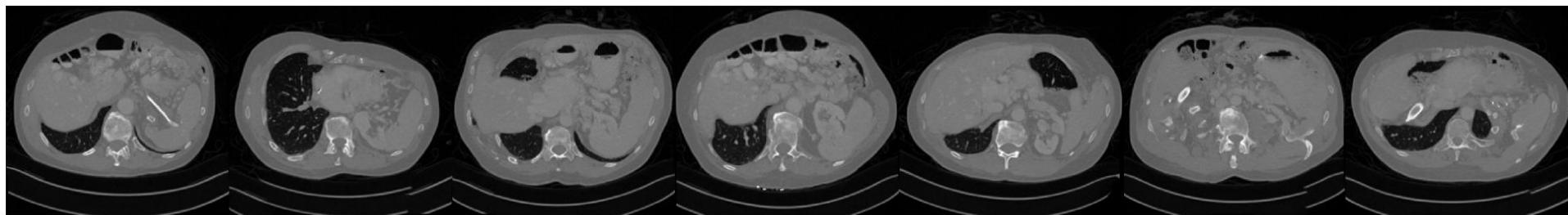
PaDIS: Patch-based Diffusion Inverse Solver

- Random partition in each sampling step
- No boundary artifacts because of shifting patch locations



PaDIS: Patch-based Diffusion Inverse Solver

- Generate entire images via positional encoding



PaDIS: Patch-based Diffusion Inverse Solver

- For in-distribution inverse problems
- Comparable or even better results as whole-image DIS models

Method	CT, 20 Views		CT, 8 Views		Deblurring		Superresolution	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Baseline	24.93	0.595	21.39	0.415	24.54	0.688	25.86	0.739
ADMM-TV	26.82	0.724	23.09	0.555	28.22	0.792	25.66	0.745
PnP-ADMM [42]	26.86	0.607	22.39	0.489	28.82	0.818	26.61	0.785
PnP-RED [46]	27.99	0.622	23.08	0.441	29.91	0.867	26.36	0.766
Whole image diffusion	32.84	0.835	25.74	0.706	30.19	0.853	29.17	0.827
Langevin dynamics [1]	33.03	0.846	27.03	0.689	30.60	0.867	26.83	0.744
Predictor-corrector [19]	32.35	0.820	23.65	0.546	28.42	0.724	26.97	0.685
VE-DDNM [7]	31.98	0.861	27.71	0.759	-	-	26.01	0.727
Patch Averaging [23]	33.35	0.850	28.43	0.765	29.41	0.847	27.67	0.802
Patch Stitching [66]	32.87	0.837	26.71	0.710	29.69	0.849	27.50	0.780
PaDIS (Ours)	33.57	0.854	29.48	0.767	30.80	0.870	29.47	0.846

PaDIS: Patch-based Diffusion Inverse Solver

- For out-of-distribution (OOD) inverse problems with mismatching distribution (training data: synthetic ellipses, testing data: CT images)
- Low-dimensional patch-based diffusion model training enhance generalization and robustness for OOD tasks**

Method	CT, 20 Views		CT, 60 Views		Deblurring		Superresolution	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Baseline	24.93	0.613	30.15	0.784	23.93	0.666	25.42	0.724
ADMM-TV	26.81	0.750	31.14	0.862	27.58	0.773	25.22	0.729
PnP-ADMM (Xu et al., 2020)	30.20	0.838	36.75	0.932	28.98	0.815	27.29	0.796
PnP-RED (Hu et al., 2022)	27.12	0.682	32.68	0.876	28.37	0.793	27.73	0.809
Whole image, naive	28.11	0.800	33.10	0.911	25.85	0.742	25.65	0.742
Patches, naive (Hu et al., 2024)	27.44	0.719	33.97	0.934	26.77	0.782	26.12	0.759
Self-supervised, whole (Barbano et al., 2023)	33.19	0.861	40.47	0.957	29.50	0.831	27.07	0.701
Self-supervised, patch (Ours)	33.77	0.874	41.45	0.969	30.34	0.860	28.10	0.827
Whole image, correct*	33.99	0.886	41.67	0.969	29.87	0.851	28.33	0.801
Patches, correct*	34.02	0.889	41.70	0.967	30.12	0.865	28.49	0.835

*not available in practice for mismatched distribution inverse problems

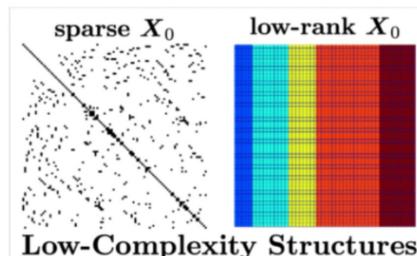
Challenges of Diffusion Inverse Solvers in Scientific Applications

Efficiency

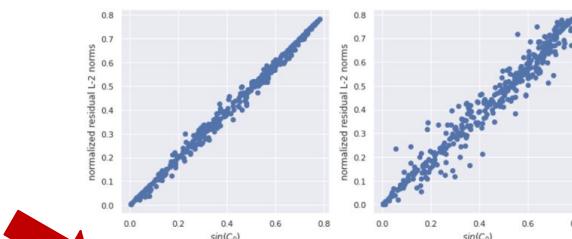


(Scientific)
Applications

Generalizability



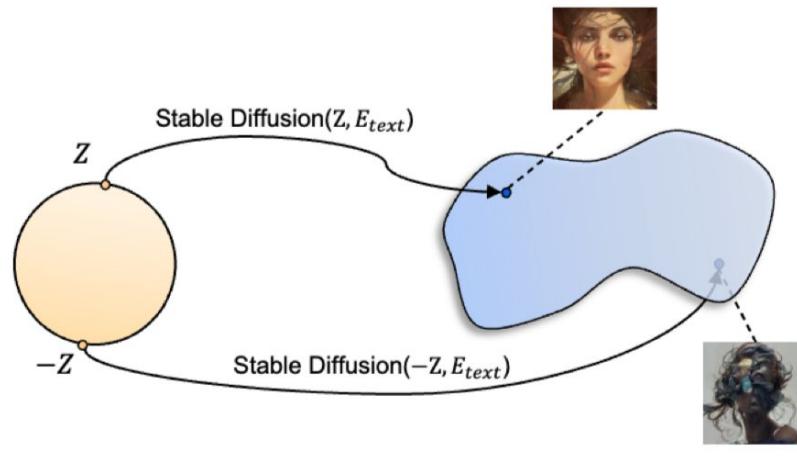
Low-Complexity Structures



Controllability

Motivations

- Diffusion models provide potential sampling from posterior distribution
- Initial noise space of diffusion model offers an inherent (low-dimensional) latent structures of high-dimensional data!



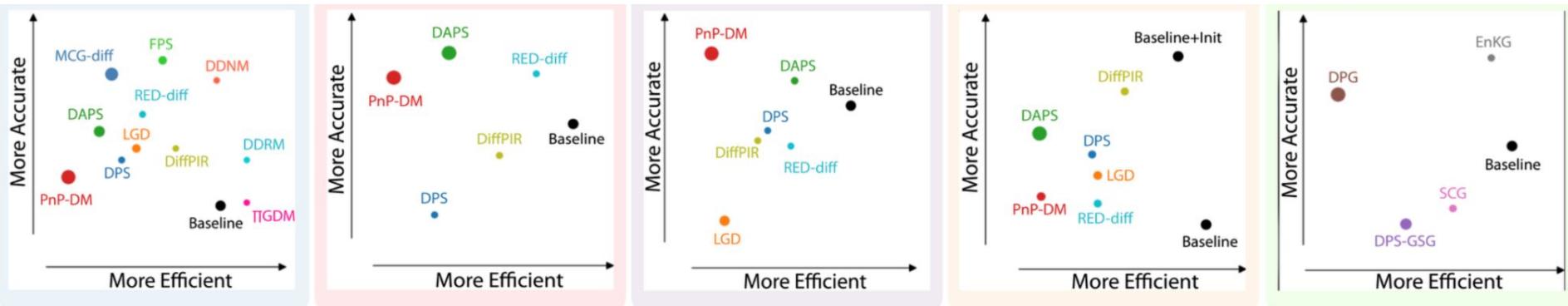
(a) Stable Diffusion



(b) CelebA-HQ

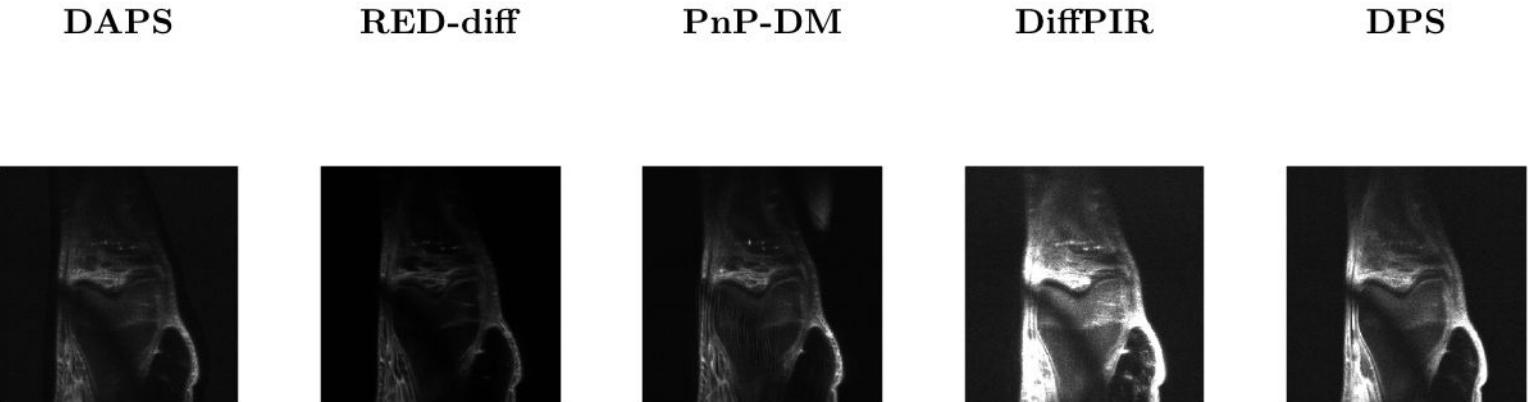
Motivations

- Diffusion models provide potential sampling from posterior distribution
- Initial noise space of diffusion model offers an inherent (low-dimensional) latent structures of high-dimensional data!
- Uncertainty estimation is desired in real-world applications
 - Existing benchmarks focus on measuring accuracy of a random sample against groundtruth



Motivations

- Diffusion models provide potential sampling from posterior distribution
- Initial noise space of diffusion model offers an inherent (low-dimensional) latent structures of high-dimensional data!
- Uncertainty estimation is desired in real-world applications
 - Existing benchmarks focus on measuring accuracy of a random sample against groundtruth
 - Diffusion posterior sampling from initial noise space offers more statistical information about sampled posterior distribution

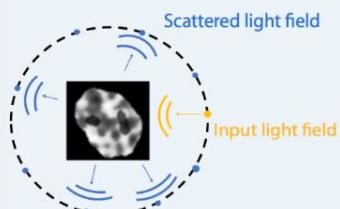


Motivations

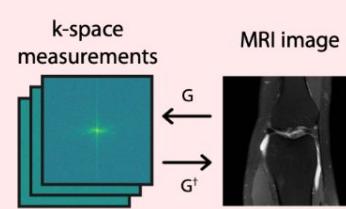
- Diffusion models provide sampling from posterior distribution
- Initial noise space of diffusion model offers an inherent low-dimensional latent structures of high-dimensional data!
- Scientific applications always require to control sampled data satisfying certain (physical) constraints

How to guarantee controllable and constrained sampling with diffusion models?

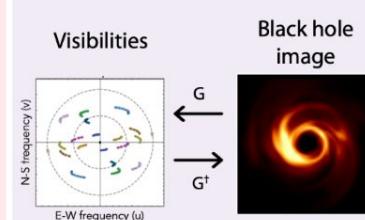
Linear inverse scattering



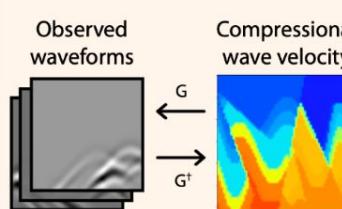
Compressed sensing MRI



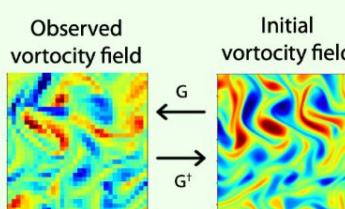
Black hole imaging



Full waveform inversion



Navier-Stokes equation



Controllable and Constrained Sampling: Toy Example

- Given a target image, how to conduct constrained sampling around target image with controlled mean and variance



Target Image



Controllable and Constrained Sampling: Toy Example

- Exploring initial noise perturbation with a controller algorithm can achieve this constrained sampling!



Target Image

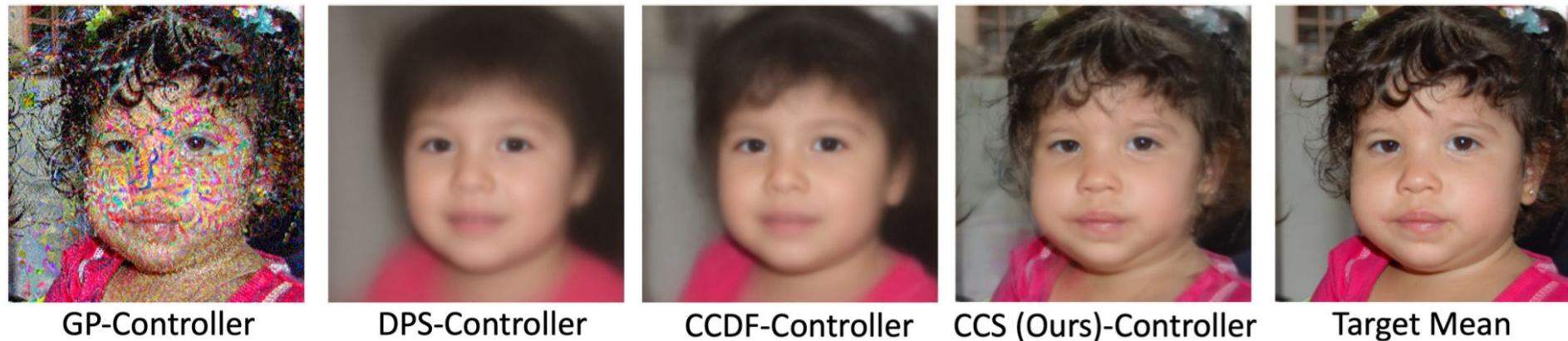
Controllable and Constrained Sampling: Toy Example

- Despite diverse samples, the sample mean closely matches the desired target image (at the constrained distance)



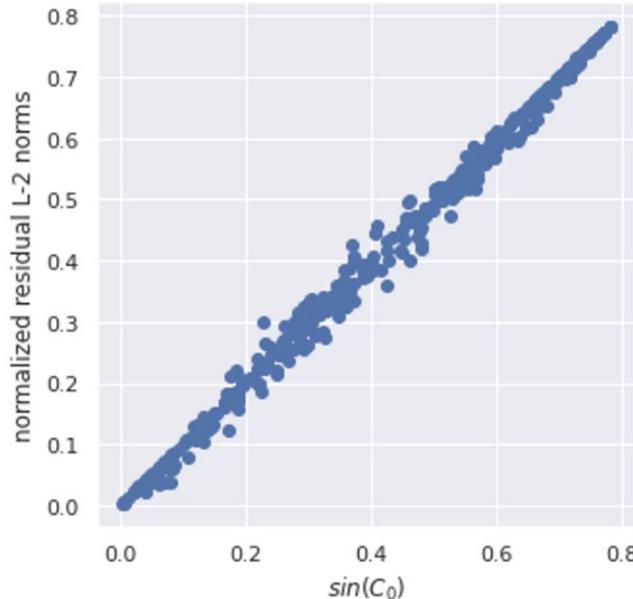
Controllable and Constrained Sampling: Toy Example

- Compared with other methods, CCS gives sample mean that better matches the desired target image (at the constrained distance)

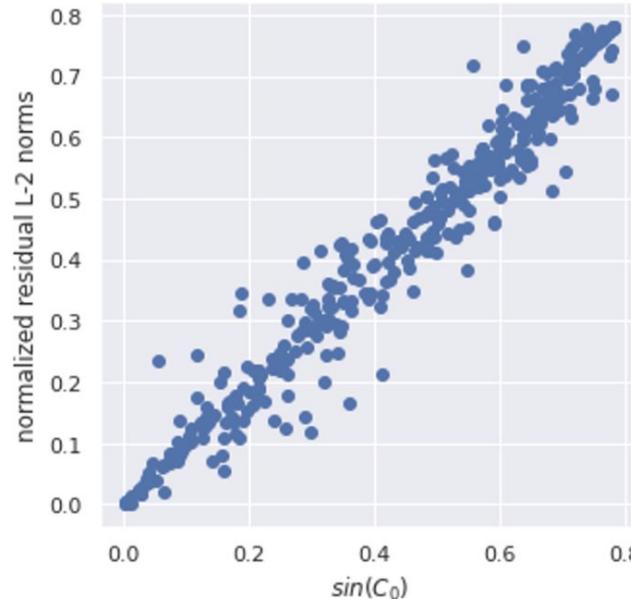


Controllable and Constrained Sampling with Diffusion Models

- Linearity between the magnitude of initial noise perturbation and sample-target image distance



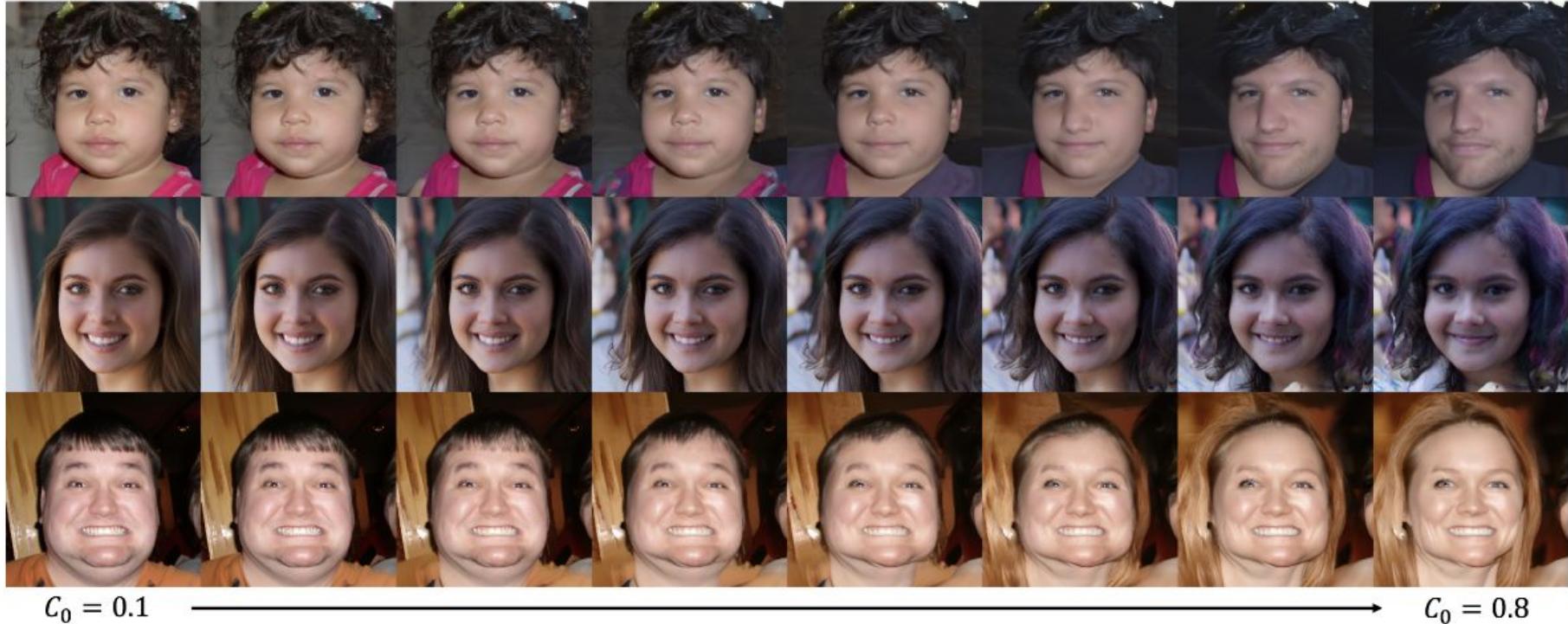
FFHQ dataset with pixel diffusion model



Celeba-HQ dataset with latent diffusion model

Controllable and Constrained Sampling with Diffusion Models

- Linearity when increasing the magnitude of initial noise perturbation

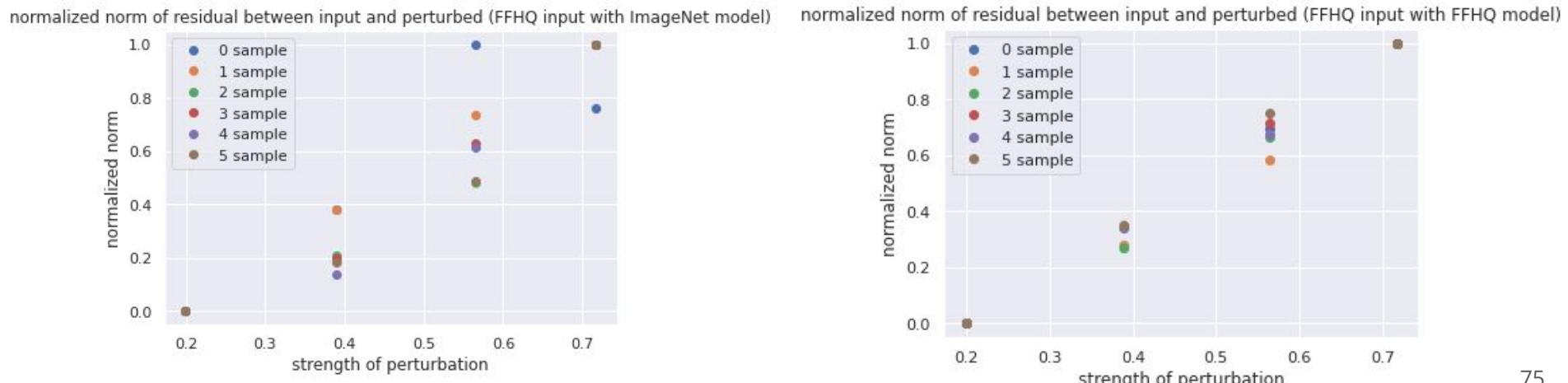


Implications of the Linearity Score

- However, the linearity of perturbation for different centers is different across initial noise manifold

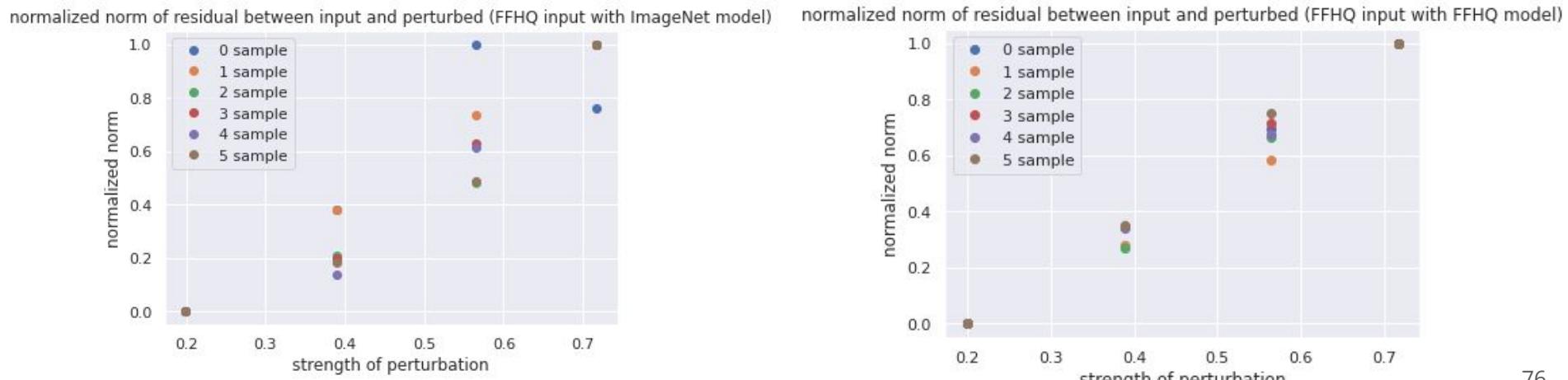
Implications of the Linearity Score

- However, the linearity of perturbation for different centers is different across initial noise manifold
- Out-of-distribution (OOD) data shows lower linearity score
 - OOD data lies on a low-probability manifold



Implications of the Linearity Score

- However, the linearity of perturbation for different centers is different across initial noise manifold
- Out-of-distribution (OOD) data shows lower linearity score
 - OOD data lies on a low-probability manifold
- **The linearity score reveals inherent properties for understanding the low-dimensional geometry of learned data manifold**
 - Transition from memorization regime to generalization regime



Designing low-dimensional tokenizers

- Pre-training considerations:
 - Improved linearity enhances generalization capability
 - Perturbation in latent space [1,2]
- Post-training considerations:
 - Improved linearity enhances controllability
 - Decoder post-training for alignment [1,2]

[1] Qiu, et al., Image Tokenizer Needs Post-Training, arXiv 2025

[2] Zheng, et al., Diffusion Transformers with Representation Autoencoders, arXiv 2025

[3] Song, et al., CCS: Controllable and Constrained Sampling with Diffusion Models via Initial Noise Perturbation, NeurIPS 2025

Designing low-dimensional tokenizers

- Pre-training considerations:
 - Improved linearity enhances generalization capability
 - Perturbation in latent space [1,2]
- Post-training considerations:
 - Improved linearity enhances controllability
 - Decoder post-training for alignment [1,2]
- More thoughts...
 - Improving linearity as much as possible? No!
 - If fully linear, a diffusion model collapses to a linear model
 - Tradeoff between fine-details and controllability/generalizability

Our Blog Post:

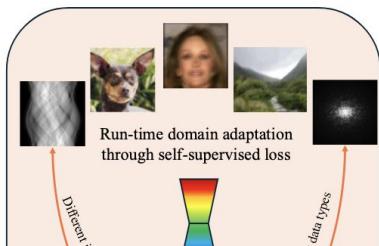


[1] Qiu, et al., Image Tokenizer Needs Post-Training, arXiv 2025

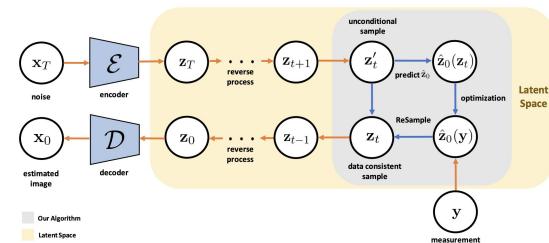
[2] Zheng, et al., Diffusion Transformers with Representation Autoencoders, arXiv 2025

[3] Song, et al., CCS: Controllable and Constrained Sampling with Diffusion Models via Initial Noise Perturbation, NeurIPS 2025

Challenges of Diffusion Inverse Solvers in Scientific Applications



Efficiency

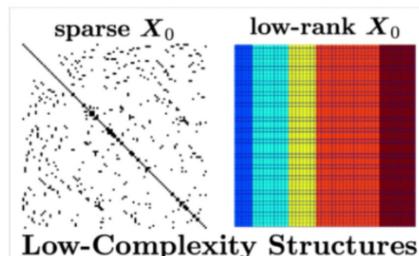


Low-dimensional structures are the key to address real-world high-dimensional data with small data regime in scientific applications

Ours: Pre-training models and dataset can be completely off-the-shelf.

(Scientific)
Applications

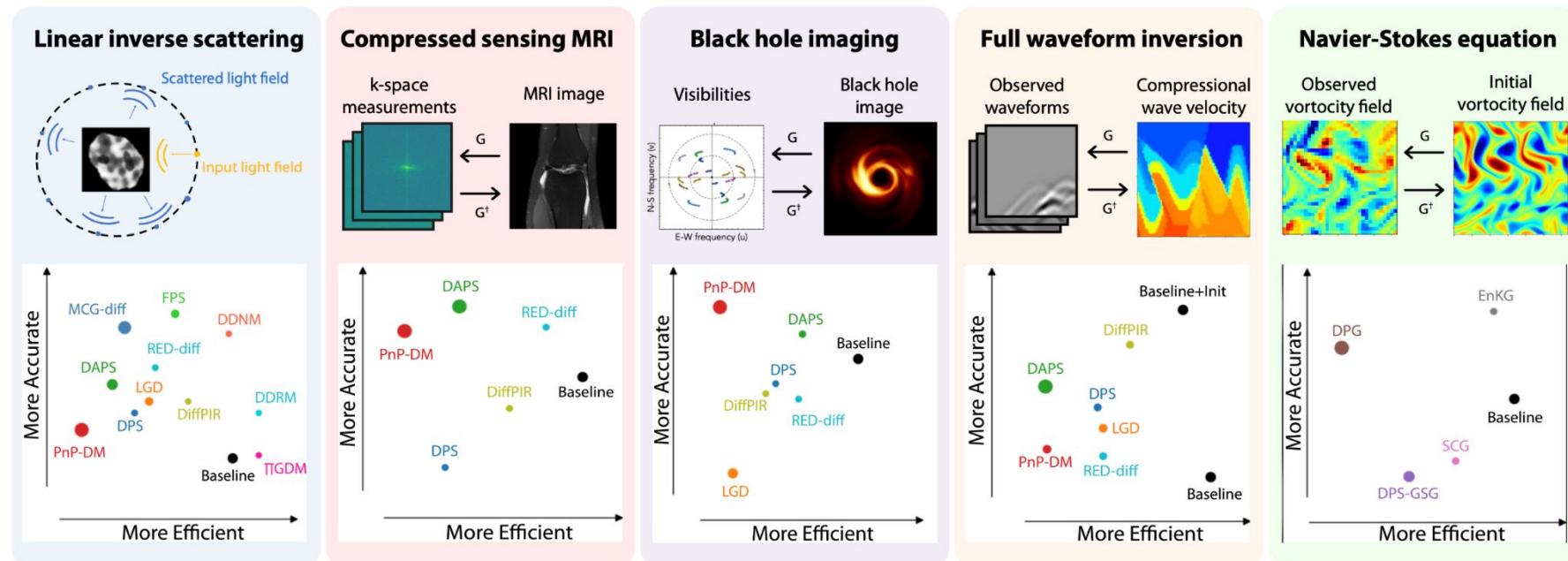
Generalizability



Controllability

DIS for Solving Various Inverse Problems in Scientific Applications

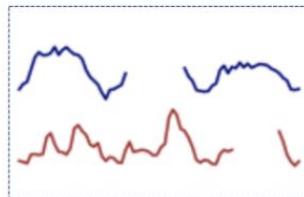
- Computational imaging
 - Dynamic systems
 -



CSDI: Time Series Imputation

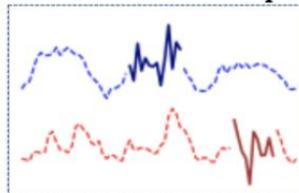
- Scientific applications in finance, meteorology and healthcare
-

Conditional observations \mathbf{x}_0^{co}



$$p_{\theta}(\mathbf{x}_{t-1}^{\text{ta}} | \mathbf{x}_t^{\text{ta}}, \mathbf{x}_0^{\text{co}})$$

Random noise \mathbf{x}_T^{ta}



\mathbf{x}_T^{ta}

\mathbf{x}_T^{ta}

\mathbf{x}_t^{ta}

\mathbf{x}_t^{ta}

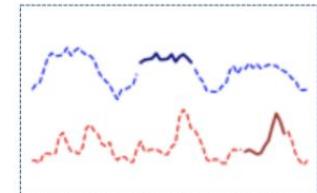
$\mathbf{x}_{t-1}^{\text{ta}}$

$\mathbf{x}_{t-1}^{\text{ta}}$

\mathbf{x}_0^{ta}

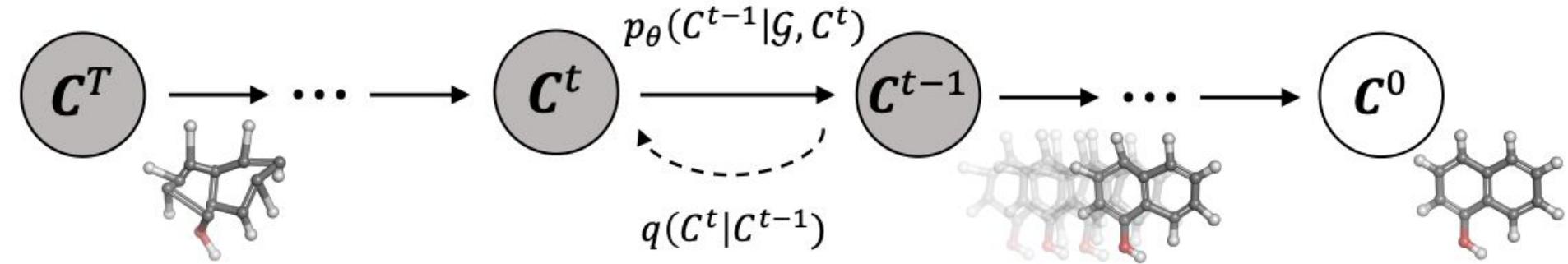
\mathbf{x}_0^{ta}

Imputation targets \mathbf{x}_0^{ta}



GeoDiff: Molecular Conformation Prediction

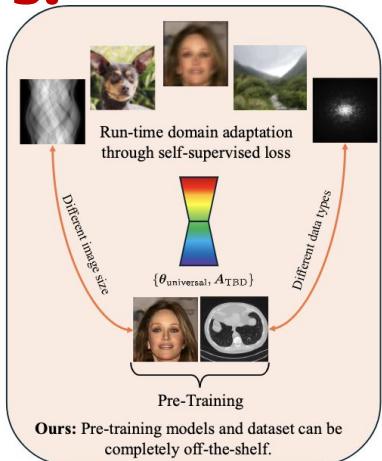
- Scientific applications in cheminformatics and drug discovery
-



~~Challenges of Diffusion Inverse Solvers in Scientific Applications~~

Opportunities

s!

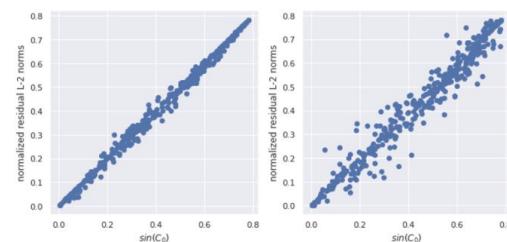
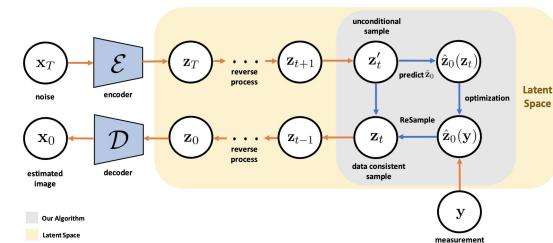


Efficiency



(Scientific)
Applications

Generalizability



Controllability

Diffusion models can benefit numerous scientific applications!

Acknowledgement

Thanks!

- Support from the NSF, DARPA, Hyundai America Technical Center, Inc. (HATCI), Michigan MICDE Catalyst Grant, Michigan MIDAS PODS Grant
- All PhD/MS/UG students in the [BioMed-AI Lab](#) at U-M
- Collaborators at U-M EECS / BME / Radiology / Radiation Oncology...
- Contact: liyues@umich.edu



Jeff Fessler



Qing Qu



Bowen Song



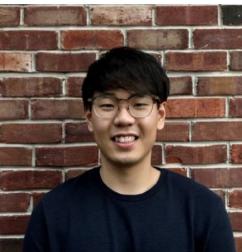
Jason Hu



Chenwei Wu



Lixuan Chen



Soo-Min Kwon



National
Science
Foundation

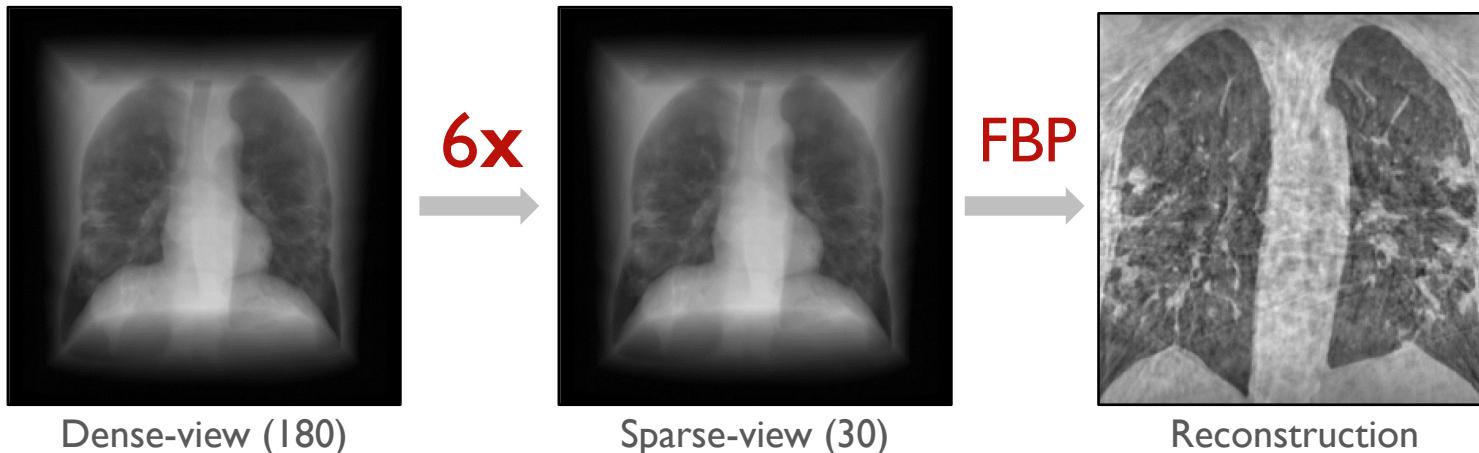


HYUNDAI
MOTOR GROUP

HYUNDAI AMERICA TECHNICAL CENTER, INC.

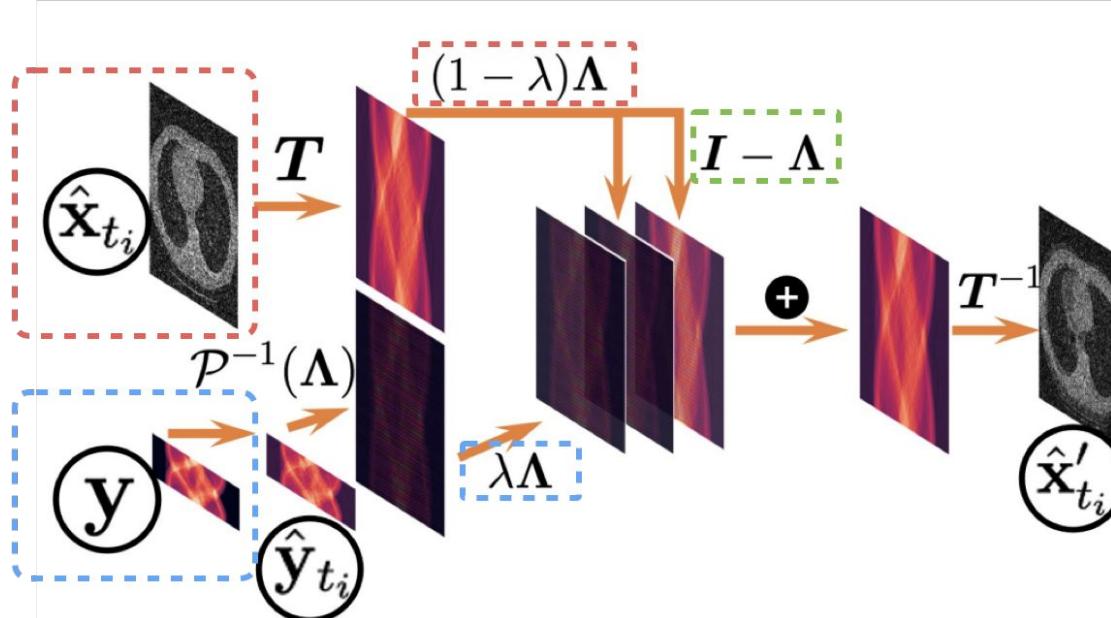
Medical Imaging

- X-ray Computed Tomography (CT) imaging
 - Reconstruct cross-sectional images of internal structures from X-ray projections at multiple views
- Reduce radiation dose in CT scanning
 - Sample sparse projections from fewer views



Optimization Solution

$$\hat{\mathbf{x}}'_{t_i} = \mathbf{T}^{-1} [(\mathbf{I} - \Lambda)\mathbf{T}\hat{\mathbf{x}}_{t_i} + (1 - \lambda)\Lambda\mathbf{T}\hat{\mathbf{x}}_{t_i} + \lambda\Lambda\mathcal{P}^{-1}(\Lambda)\hat{\mathbf{y}}_{t_i}]$$

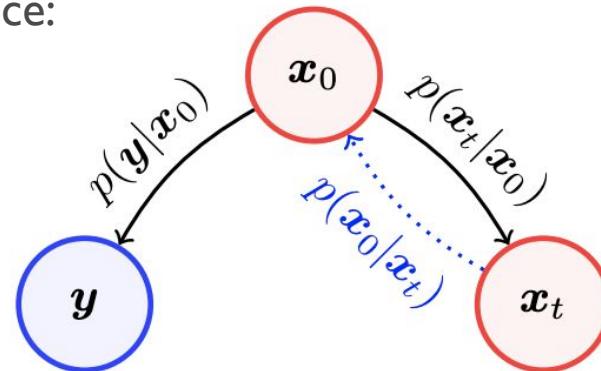


x: image
y: observed measurements
A: forward model
T: transformation
 Λ : sampling mask
P: dimension reduction operator

Posterior sampling consistent with **data distribution prior** and **observed measurements**

Diffusion Posterior Sampling for General Inverse Problems

Conditional independence:



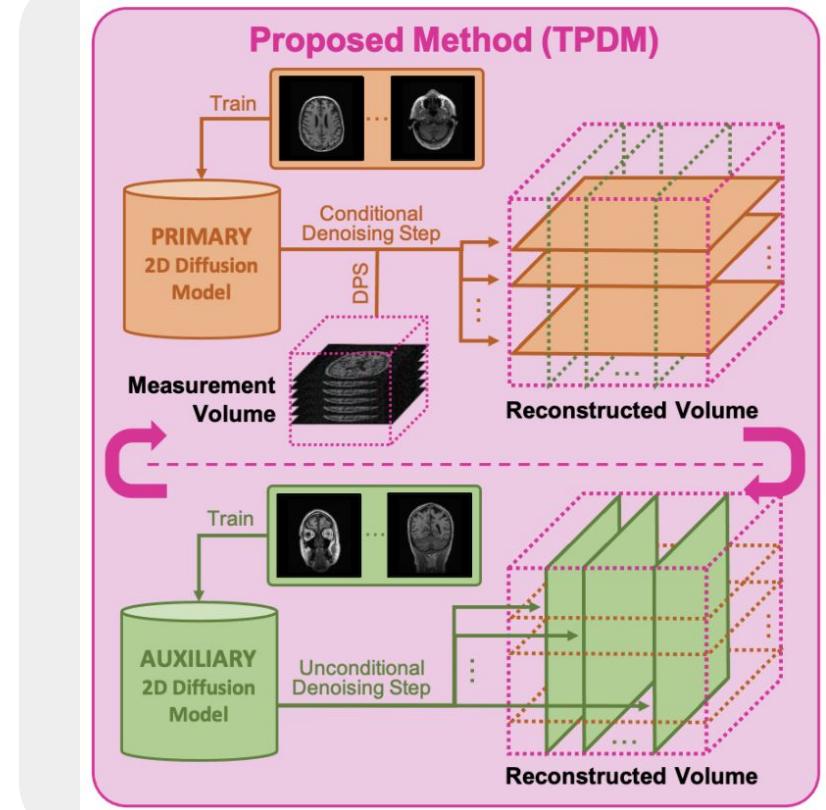
$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \boxed{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y} | \mathbf{x}_t)}$$

$$\begin{aligned} p(\mathbf{y} | \mathbf{x}_t) &= \int p(\mathbf{y} | \mathbf{x}_0, \mathbf{x}_t) p(\mathbf{x}_0 | \mathbf{x}_t) d\mathbf{x}_0 \\ &= \int p(\mathbf{y} | \mathbf{x}_0) p(\mathbf{x}_0 | \mathbf{x}_t) d\mathbf{x}_0, \end{aligned}$$

Two Perpendicular 2D Diffusion Models (TPDM)

- Leverage auxiliary 2D diffusion model

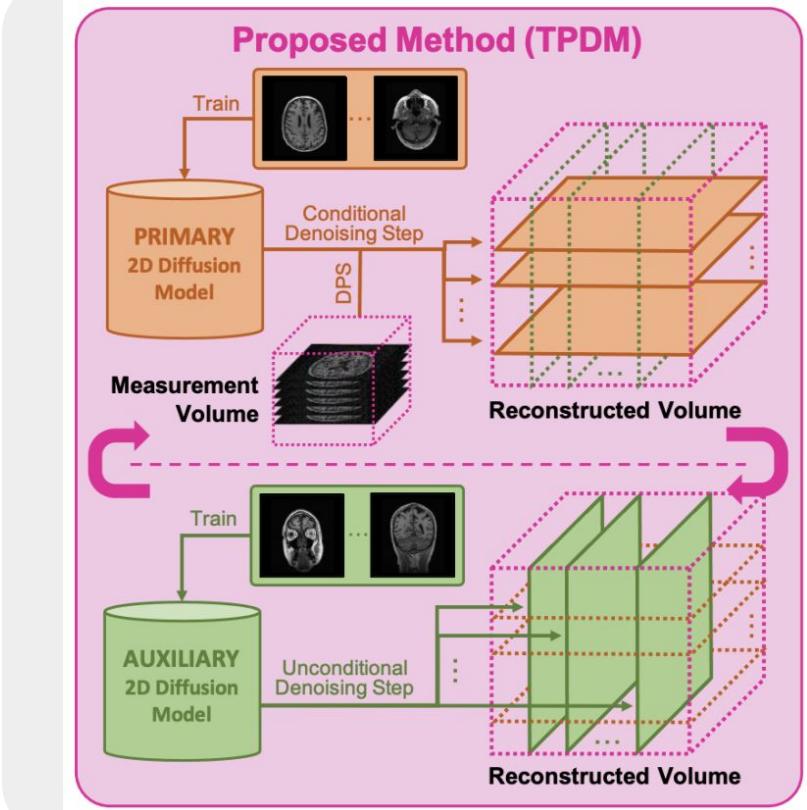
$$p_{\theta, \phi}(\mathbf{x}) = q_{\theta}^{(p)}(\mathbf{x})^{\alpha} q_{\phi}^{(a)}(\mathbf{x})^{\beta} / Z$$



Two Perpendicular 2D Diffusion Models (TPDM)

- Leverage auxiliary 2D diffusion model
- 2D slice independence assumption

$$\begin{aligned} p_{\theta, \phi}(\mathbf{x}) &= q_{\theta}^{(p)}(\mathbf{x})^{\alpha} q_{\phi}^{(a)}(\mathbf{x})^{\beta} / Z \\ &= [q_{\theta}^{(p)}(\mathbf{x}_{[:, :, 1]}) q_{\theta}^{(p)}(\mathbf{x}_{[:, :, 2]}) \cdots q_{\theta}^{(p)}(\mathbf{x}_{[:, :, d_3}])]^{\alpha} \\ &\times [q_{\phi}^{(a)}(\mathbf{x}_{[1, :, :]}) q_{\phi}^{(a)}(\mathbf{x}_{[2, :, :]}) \cdots q_{\phi}^{(a)}(\mathbf{x}_{[d_1, :, :]})]^{\beta} / Z \end{aligned}$$

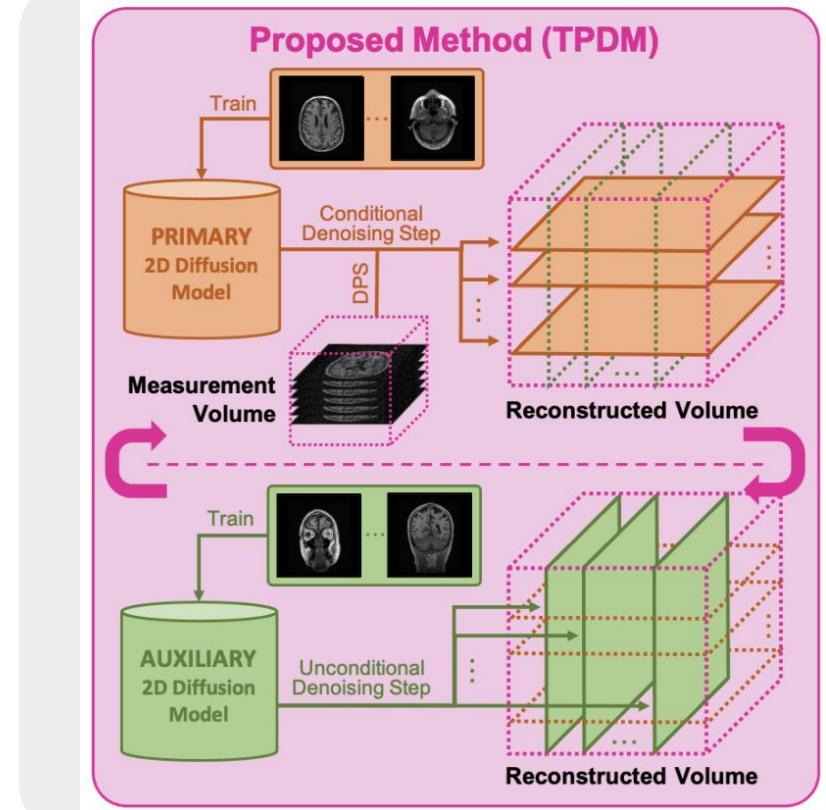


Two Perpendicular 2D Diffusion Models (TPDM)

- Leverage auxiliary 2D diffusion model
- 2D slice independence assumption

$$\begin{aligned}
 p_{\theta, \phi}(\mathbf{x}) &= q_{\theta}^{(p)}(\mathbf{x})^{\alpha} q_{\phi}^{(a)}(\mathbf{x})^{\beta} / Z \\
 &= [q_{\theta}^{(p)}(\mathbf{x}_{[:, :, 1]}) q_{\theta}^{(p)}(\mathbf{x}_{[:, :, 2]}) \cdots q_{\theta}^{(p)}(\mathbf{x}_{[:, :, d_3}])]^{\alpha} \\
 &\times [q_{\phi}^{(a)}(\mathbf{x}_{[1, :, :]}) q_{\phi}^{(a)}(\mathbf{x}_{[2, :, :]}) \cdots q_{\phi}^{(a)}(\mathbf{x}_{[d_1, :, :]})]^{\beta} / Z
 \end{aligned}$$

$$\begin{aligned}
 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) &= \alpha \nabla_{\mathbf{x}_t} \log q^{(p)}(\mathbf{x}_t) + \beta \nabla_{\mathbf{x}_t} \log q^{(a)}(\mathbf{x}_t) \\
 &= \alpha \sum_{i=1}^{d_3} \nabla_{\mathbf{x}_t} \log q^{(p)}(\mathbf{x}_{t,[:, :, i]}) + \beta \sum_{i=1}^{d_1} \nabla_{\mathbf{x}_t} \log q^{(a)}(\mathbf{x}_{t,[i, :, :]}) \\
 &\simeq \alpha \sum_{i=1}^{d_3} \mathbf{s}_{\theta^*}^{3D}(\mathbf{x}_{t,[:, :, i]}) + \beta \sum_{i=1}^{d_1} \mathbf{s}_{\phi^*}^{3D}(\mathbf{x}_{t,[i, :, :]})
 \end{aligned}$$

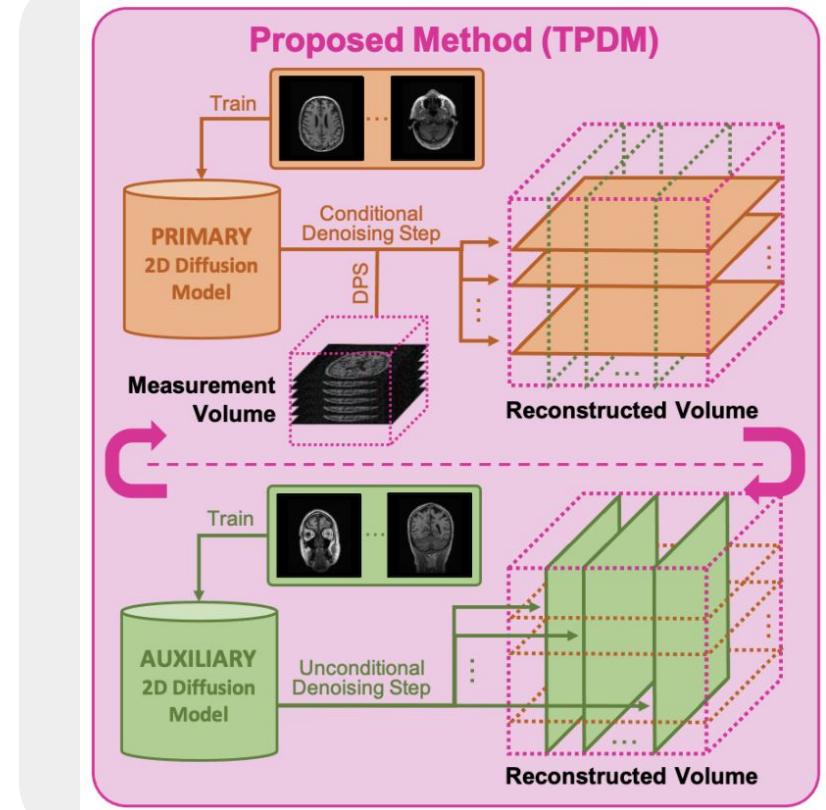


Two Perpendicular 2D Diffusion Models (TPDM)

- Leverage auxiliary 2D diffusion model
- 2D slice independence assumption
- Not true in medical imaging!

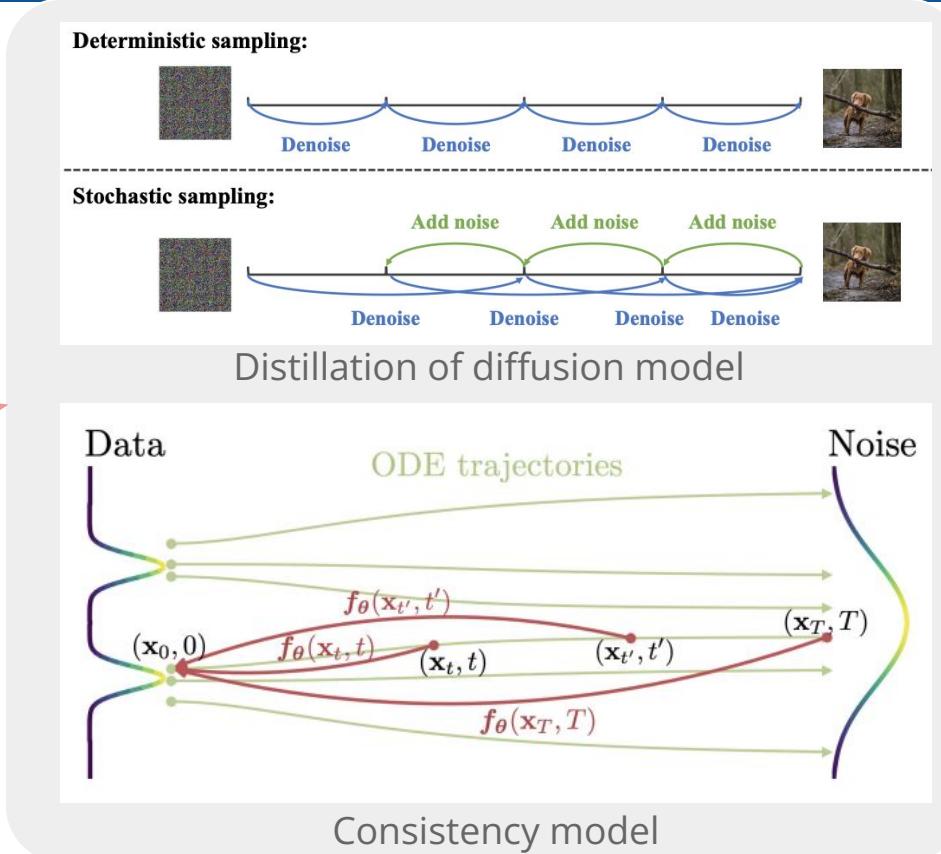
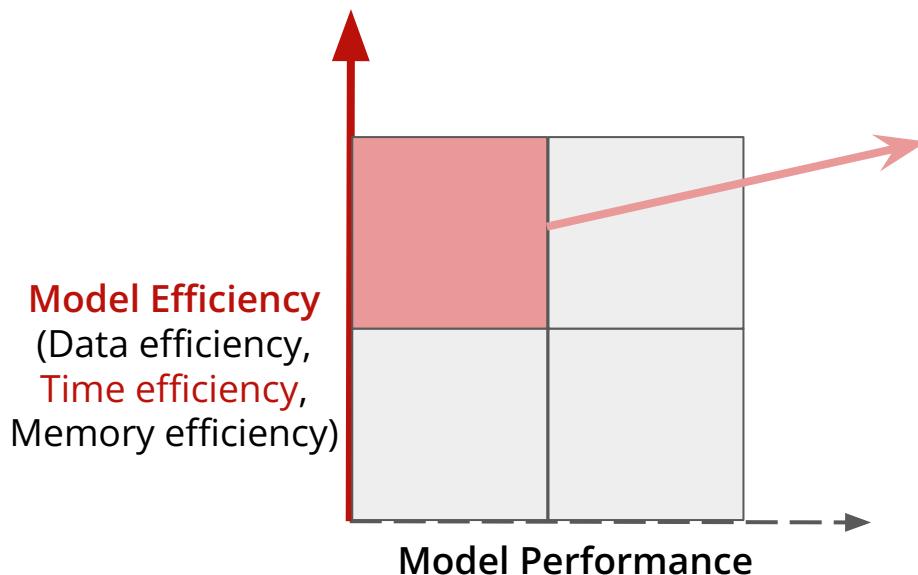
$$\begin{aligned} p_{\theta, \phi}(\mathbf{x}) &= q_{\theta}^{(p)}(\mathbf{x})^{\alpha} q_{\phi}^{(a)}(\mathbf{x})^{\beta} / Z \\ &= [q_{\theta}^{(p)}(\mathbf{x}_{[:, :, 1]}) q_{\theta}^{(p)}(\mathbf{x}_{[:, :, 2]}) \cdots q_{\theta}^{(p)}(\mathbf{x}_{[:, :, d_3}])]^{\alpha} \\ &\times [q_{\phi}^{(a)}(\mathbf{x}_{[1, :, :]}) q_{\phi}^{(a)}(\mathbf{x}_{[2, :, :]}) \cdots q_{\phi}^{(a)}(\mathbf{x}_{[d_1, :, :]})]^{\beta} / Z \end{aligned}$$

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) &= \alpha \nabla_{\mathbf{x}_t} \log q^{(p)}(\mathbf{x}_t) + \beta \nabla_{\mathbf{x}_t} \log q^{(a)}(\mathbf{x}_t) \\ &= \alpha \sum_{i=1}^{d_3} \nabla_{\mathbf{x}_t} \log q^{(p)}(\mathbf{x}_{t,[:, :, i]}) + \beta \sum_{i=1}^{d_1} \nabla_{\mathbf{x}_t} \log q^{(a)}(\mathbf{x}_{t,[i, :, :]}) \\ &\simeq \alpha \sum_{i=1}^{d_3} \mathbf{s}_{\theta^*}^{3D}(\mathbf{x}_{t,[:, :, i]}) + \beta \sum_{i=1}^{d_1} \mathbf{s}_{\phi^*}^{3D}(\mathbf{x}_{t,[i, :, :]}) \end{aligned}$$



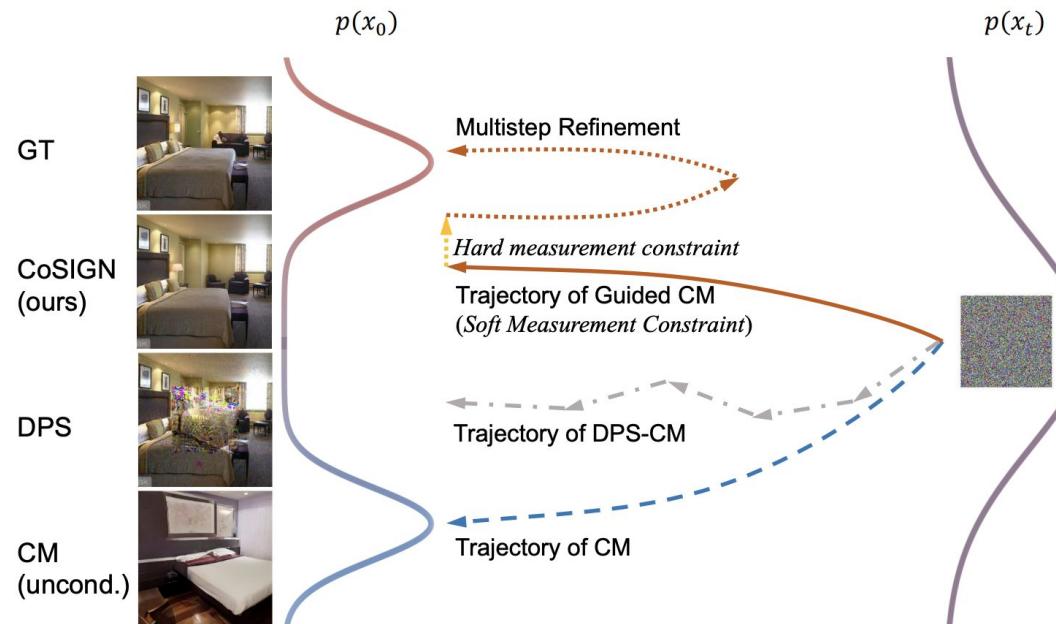
Enhance Diffusion Model Efficiency

- Improve time efficiency
 - Enable sampling as few as 1~4 steps for image generation
- Nontrivial to integrate with DIS!



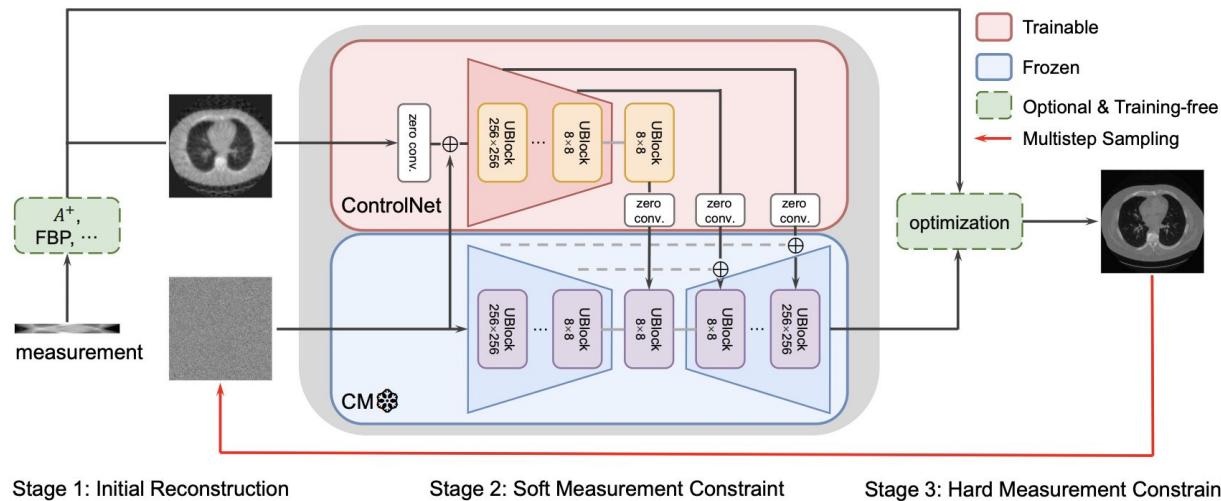
Leverage Consistency Model as Data Prior

- Push the boundary of inference steps to 1-2 NFEs
- Existing DIS methods fail to achieve good measurement consistency in 1-2 NFEs



CoSIGN: Few-Step Guidance of Consistency Model to Solve

- Enforce two types of constraints during sampling process:
 - Soft measurement constraint with ControlNet
 - Hard measurement constraint via optimization



ReSample

- Algorithm:

Algorithm 1 ReSample: Solving Inverse Problems with Latent Diffusion Models

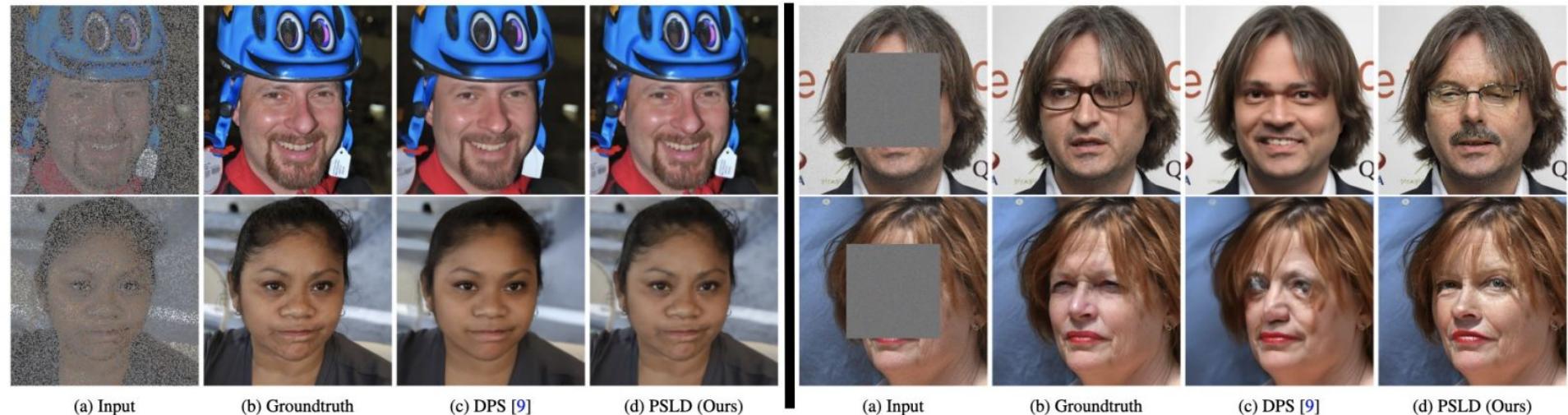
Require: Measurements \mathbf{y} , $\mathcal{A}(\cdot)$, Encoder $\mathcal{E}(\cdot)$, Decoder $\mathcal{D}(\cdot)$, Score function $s_\theta(\cdot, t)$, Pretrained LDM Parameters $\beta_t, \bar{\alpha}_t, \eta, \delta$, Hyperparameter γ to control σ_t^2 , Time steps to perform resample C

$\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Initial noise vector

for $t = T - 1, \dots, 0$ **do**

- $\epsilon_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- $\hat{\mathbf{z}}_0 = \frac{1}{\sqrt{\bar{\alpha}_{t+1}}}(\mathbf{z}_{t+1} + \sqrt{1 - \bar{\alpha}_{t+1}}s_\theta(\mathbf{z}_{t+1}, t+1))$ ▷ Predict $\hat{\mathbf{z}}_0$
- $\mathbf{z}'_t = \sqrt{\bar{\alpha}_t}\hat{\mathbf{z}}_0 + \sqrt{1 - \bar{\alpha}_t - \eta\delta^2}s_\theta(\mathbf{z}_{t+1}, t+1) + \eta\delta\epsilon_1$ ▷ Unconditional DDIM step
- if** $t \in C$ **then**
 - $\hat{\mathbf{z}}_0(\mathbf{z}_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{z}'_t + \sqrt{(1 - \bar{\alpha}_t)}s_\theta(\mathbf{z}'_t, t))$ ▷ Use Tweedie's formula
 - $\hat{\mathbf{z}}_0(\mathbf{y}) \in \arg \min_{\mathbf{z}} \frac{1}{2}\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}))\|_2^2$ ▷ Solve with initial point $\hat{\mathbf{z}}_0(\mathbf{z}_t)$
 - $\mathbf{z}_t = \text{StochasticResample}(\hat{\mathbf{z}}_0(\mathbf{y}), \mathbf{z}'_t, \gamma)$ ▷ Map back to t
- else**
 - $\mathbf{z}_t = \mathbf{z}'_t$ ▷ Unconditional sampling if not resampling
 - $\mathbf{x}_0 = \mathcal{D}(\mathbf{z}_0)$ ▷ Output reconstructed image

- Solving linear inverse problems provably via posterior sampling with LDMs
- Goodness Modified Latent DPS (GML-DPS)



- Solving linear inverse problems provably via posterior sampling with LDMs
- Goodness Modified Latent DPS (GML-DPS)
 - Explain measurements when passed through decoder

$$\nabla_{\mathbf{z}_t} \log p(\mathbf{y}|\mathbf{z}_t) = \underbrace{\nabla_{\mathbf{z}_t} \log p(\mathbf{y}|\hat{\mathbf{x}}_0 = \mathcal{D}(\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t]))}_{\text{DPS vanilla extension}} + \gamma_t \underbrace{\nabla_{\mathbf{z}_t} ||\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t] - \mathcal{E}(\mathcal{D}(\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t]))||^2}_{\text{"goodness" of } \mathbf{z}_0}$$

- Solving linear inverse problems provably via posterior sampling with LDMs
- Goodness Modified Latent DPS (GML-DPS)
 - Explain measurements when passed through decoder
 - Generated sample remains on the manifold of real data

$$\nabla_{\mathbf{z}_t} \log p(\mathbf{y}|\mathbf{z}_t) = \underbrace{\nabla_{\mathbf{z}_t} \log p(\mathbf{y}|\hat{\mathbf{x}}_0 = \mathcal{D}(\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t]))}_{\text{DPS vanilla extension}} + \gamma_t \underbrace{\nabla_{\mathbf{z}_t} ||\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t] - \mathcal{E}(\mathcal{D}(\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t]))||^2}_{\text{"goodness" of } \mathbf{z}_0}$$

- Solving **linear** inverse problems provably via posterior sampling with LDMs
- **Goodness Modified Latent DPS (GML-DPS)**
 - Explain measurements when passed through decoder
 - Generated sample remains on the manifold of real data
 - Gluing objective to ensure that denoising update and measurement-matching update converge

$$\nabla_{\mathbf{z}_t} \log p(\mathbf{y}|\mathbf{z}_t) = \underbrace{\nabla_{\mathbf{z}_t} \log p(\mathbf{y}|\hat{\mathbf{x}}_0 = \mathcal{D}(\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t]))}_{\text{DPS vanilla extension}} + \gamma_t \underbrace{\nabla_{\mathbf{z}_t} \|\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t] - \mathcal{E}(\mathcal{D}(\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t]))\|^2}_{\text{"goodness" of } \mathbf{z}_0}$$

$\gamma_t \boxed{\nabla_{\mathbf{z}_t} \|\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t] - \mathcal{E}(\mathcal{A}^T \mathbf{y} + (\mathbf{I} - \mathcal{A}^T \mathcal{A}) \mathcal{D}(\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t]))\|^2}$

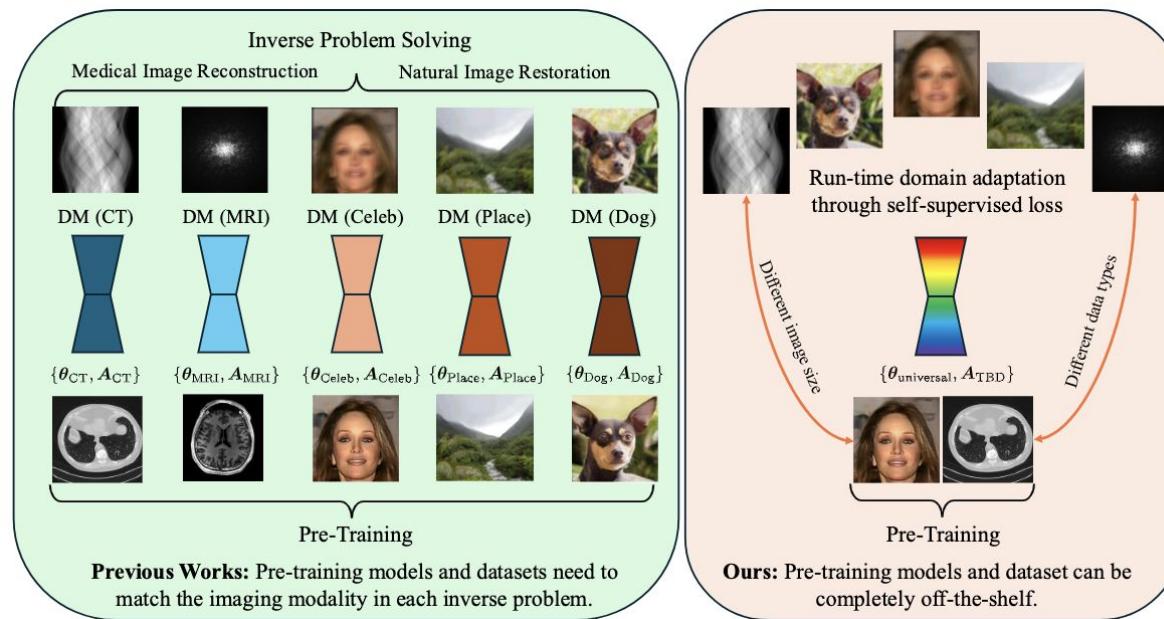
“gluing” of \mathbf{z}_0

Steerable Conditional Diffusion for OOD Adaptation

$q(\mathbf{x}_0)$	$\tilde{q}(\mathbf{x}_0)$	RED-DIFF		DDS		SCD (OURS)	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AAPM	AAPM	38.37 ± 0.09	0.941 ± 0.001	39.55 ± 0.08	0.951 ± 0.001	39.73 ± 0.07	0.952 ± 0.001
	LoDoPAB	31.09 ± 0.18	0.799 ± 0.006	31.20 ± 0.21	0.742 ± 0.007	34.21 ± 0.25	0.850 ± 0.008
	ELLIPSES	32.67 ± 0.10	0.772 ± 0.006	33.11 ± 0.11	0.787 ± 0.005	35.41 ± 0.10	0.954 ± 0.001
ELLIPSES	ELLIPSES	35.51 ± 0.09	0.878 ± 0.004	36.91 ± 0.09	0.972 ± 0.001	36.02 ± 0.08	0.968 ± 0.000
	AAPM	29.75 ± 0.06	0.801 ± 0.002	30.82 ± 0.06	0.847 ± 0.002	33.98 ± 0.12	0.883 ± 0.002
	LoDoPAB	31.25 ± 0.22	0.742 ± 0.008	31.80 ± 0.22	0.798 ± 0.009	33.42 ± 0.25	0.829 ± 0.008
BRAINWEB	AAPM	32.24 ± 0.07	0.850 ± 0.001	29.83 ± 0.07	0.779 ± 0.002	34.58 ± 0.17	0.910 ± 0.001

Versatile Diffusion Model

- Refining a single pretrained diffusion model to solve many inverse problems
- Domain adaptation at testing time through self-supervised learning
- Avoid domain-specific pre-training.



Theoretical Findings

- Approximated linear mapping of initial noise to images

$$\|x_0(x_T + \lambda \Delta x, T) - x_0(x_T, T)\| = \|\lambda \gamma_0(x_T)\| + o(\lambda) = \lambda \|\gamma_0(x_T)\| + o(\lambda)$$

- Design noise perturbations to make the samples centering around a target

$$\mathbb{E}[x'_0] = \mathbb{E}[\mathbb{E}[x'_0 | x'_T]] = x_0 + E[A\Delta x] = x_0 + AE[\Delta x]$$

Controllable and Constrained Sampling with Diffusion Models

Algorithm 1 (Full Inversion) CCS Sampling

Requires: target mean \mathbf{x}_0 , perturbation scale C_0 , number of diffusion model timesteps T

Step 0: Compute the DDIM inversion of \mathbf{x}_0 , i.e. $\mathbf{x}_T = \text{DDIM}^{-1}(\mathbf{x}_0, 0, T)$

Step 1: Sample noise $\epsilon \sim \mathcal{N}(0, I)$. Then compute

$$\theta = \cos^{-1} \left(\frac{\epsilon \cdot \mathbf{x}_T}{\|\epsilon\|_2 \|\mathbf{x}_T\|_2} \right)$$

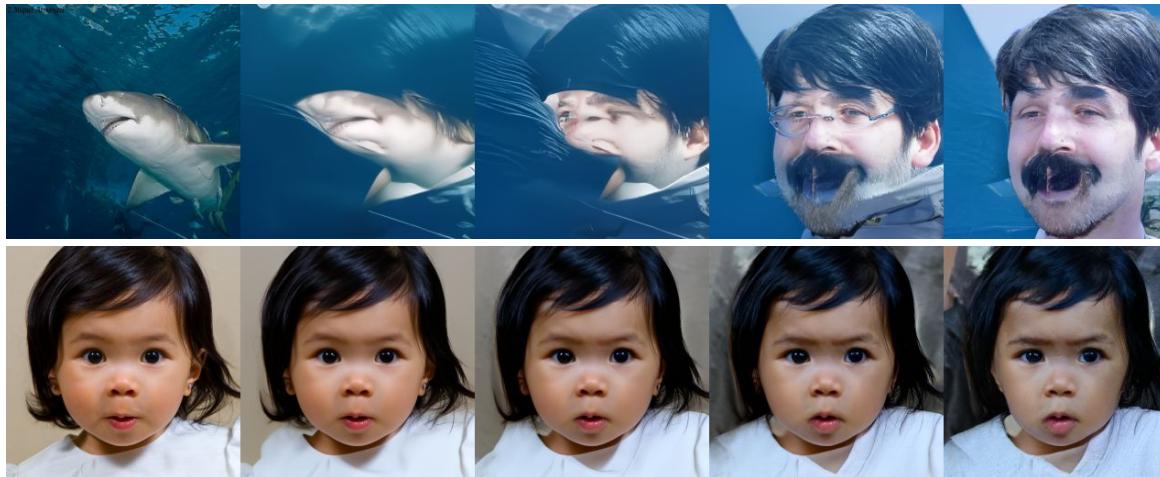
Step 2: Compute \mathbf{x}'_T using spherical interpolation formula:

$$\mathbf{x}'_T = \frac{\sin(C_0)}{\sin(\theta)} \cdot \epsilon + \frac{\sin(\theta - C_0)}{\sin(\theta)} \cdot \mathbf{x}_T$$

Step 3: Output sample $\mathbf{x}'_0 = \text{DDIM}(\mathbf{x}'_T, T, 0)$

Implications of the Linearity Score

- OOD data has a lower linearity score
 - OOD data lies on a low-probability manifold
 - Transition from Memorization Regime to Generalization Regime => Multi-delta distribution transits into ground truth data distribution => Linearity increases



Top row: ImageNet input image with increasing level of perturbation with FFHQ pretrained diffusion model
Bottom row: FFHQ input image with increasing level of perturbation with FFHQ pretrained diffusion model

Designing low-dimensional tokenizers

- Pre-training considerations:
 - Improved linearity => Improved Reconstruction/Generation gap=> Better generalization capability [3]
 - Perturbation in latent space helps, see RAE, RobustTok [1,2].
- Post-training considerations:
 - Better Linearity => Better Controllability [3]
 - Decoder post-training for alignment, see RAE, RobustTok [1,2].

[1] Qiu, et al., Image Tokenizer Needs Post-Training

[2] Zheng, et al., DIFFUSION TRANSFORMERS WITH REPRESENTATION AUTOENCODERS

[3] Song, et al., CCS: Controllable and Constrained Sampling with Diffusion Models via Initial Noise Perturbation, NeurIPS 2025

Designing low-dimensional tokenizers

- Improving linearity as much as possible? No.
- If fully linear, a diffusion model collapses to a linear model.
- Tradeoff between fine-details and controllability/generalizability.

[1] Qiu, et al., Image Tokenizer Needs Post-Training

[2] Zheng, et al., DIFFUSION TRANSFORMERS WITH REPRESENTATION AUTOENCODERS

[3] Song, et al., CCS: Controllable and Constrained Sampling with Diffusion Models via Initial Noise Perturbation, NeurIPS 2025