FMPlug: Plug-In Foundation Flow-Matching (FM) Priors for Inverse Problems

Ju Sun

Computer Science & Engineering, UMN

Dec 16, 2025

Exploiting Low-Dimensional Structures and Generative Models for Solving High-Dimensional Inverse Problems

CAMSAP 2025

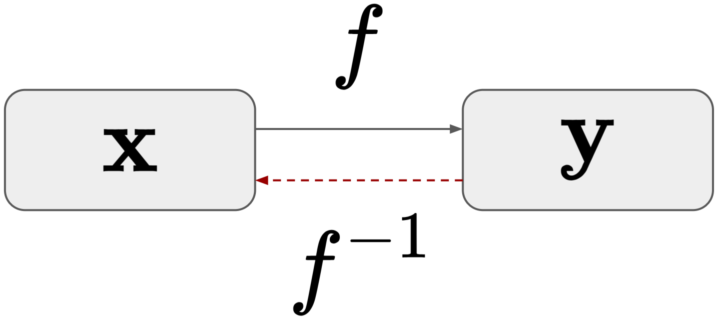
University of Minnesota
Driven to Discover[SM]

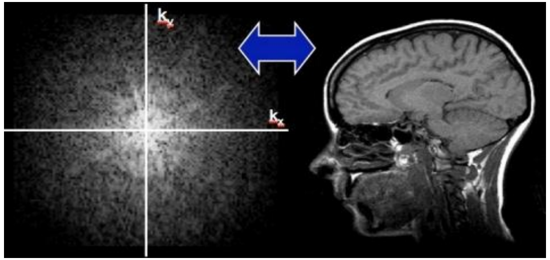
Inverse Problems

Inverse problems

Inverse problem: given $\mathbf{y} = f(\mathbf{x})$, recover $\mathbf{x}$

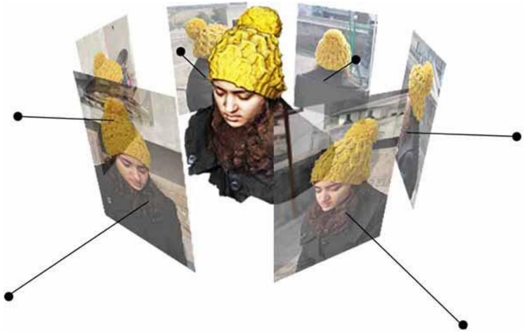
Image denoising



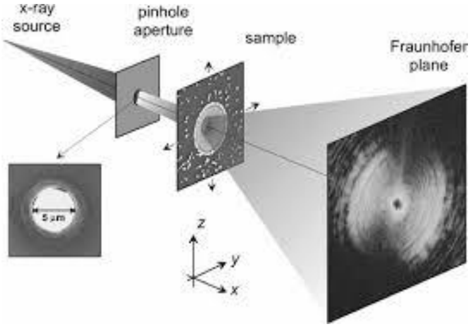


MRI reconstruction

Image super-resolution



3D reconstruction



Coherent diffraction imaging (CDI)

Traditional methods

Inverse problem: given $\mathbf{y} = f(\mathbf{x})$, recover $\mathbf{x}$

$$\min_{\mathbf{x}} \underbrace{\ell(\mathbf{y}, f(\mathbf{x}))}_{\text{data fitting}} + \lambda \underbrace{\mathrm{R}(\mathbf{x})}_{\text{regularizer}}$$

RegFit

Questions

- Which ℓ? (e.g., unknown/compound noise)
- Which R? (e.g., structures not amenable to math description)
- Speed

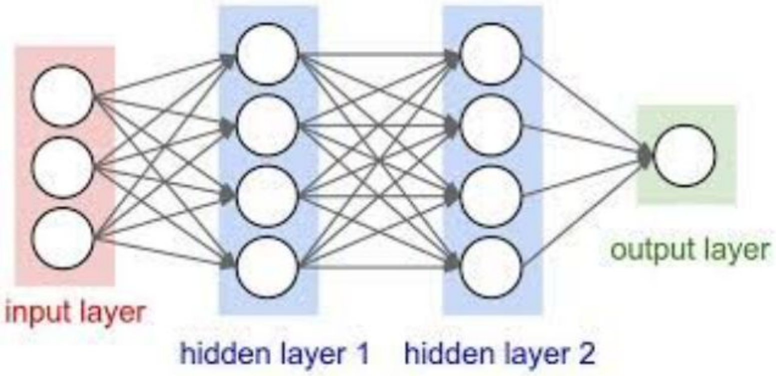
Deep learning has changed everything

With paired datasets $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1,\ldots,N}$

Direct inversion

Learn f^{-1} from $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1,\ldots,N}$

$\mathbf{y}_i$



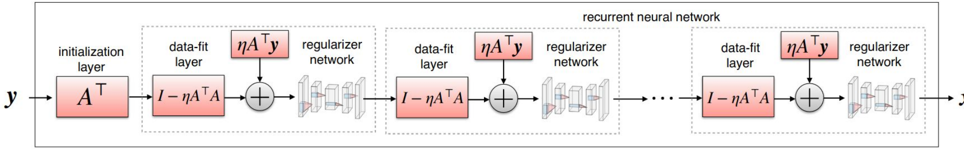
$\mathbf{x}_i$

Algorithm unrolling

$$\min_{\mathbf{x}} \ell(\mathbf{y}, f(\mathbf{x})) + \lambda \mathrm{R}(\mathbf{x})$$

$$\mathbf{x}^{k+1} = \mathcal{P}_R\left(\mathbf{x}^k - \eta \nabla^\top f(\mathbf{x}^k) \ell'(\mathbf{y}, f(\mathbf{x}^k))\right)$$

Idea: make $\mathcal{P}_R$ trainable



With object datasets only $\{\boldsymbol{x}_i\}_{i=1,\ldots,N}$

Model the distribution of the objects first, and then plug the prior in

GAN Inversion

Pretraining: $\mathbf{x}_i \approx G_\theta(\mathbf{z}_i) \ \forall i$

Deployment: $\min_{\mathbf{z}} \ \ell(\mathbf{y},\ f \circ G_\theta(\mathbf{z})) + \lambda R \circ G_\theta(\mathbf{z})$

Interleaving pretrained diffusion models

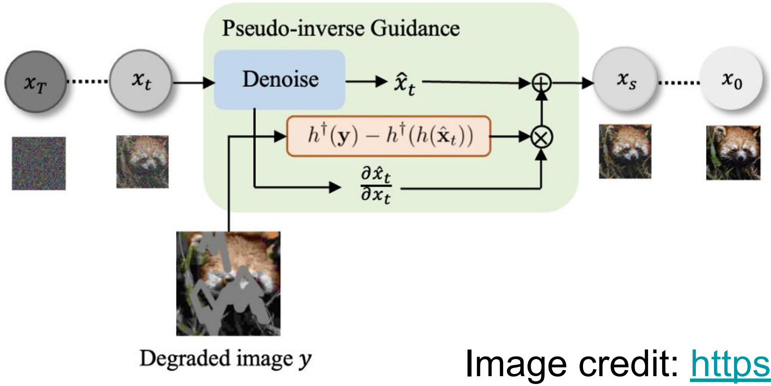


Image credit: https://arxiv.org/abs/2308.09388

Without datasets? **untrained/dataless** methods

Deep image prior (DIP) $\mathbf{x} \approx G_\theta(\mathbf{z})$ G_θ (and $\mathbf{z}$) trainable

$$\min_{\mathbf{x}} \underbrace{\ell(\mathbf{y},\, f(\mathbf{x}))}_{\text{data fitting}} + \lambda \underbrace{\mathrm{R}(\mathbf{x})}_{\text{regularizer}}$$

$$\Downarrow$$

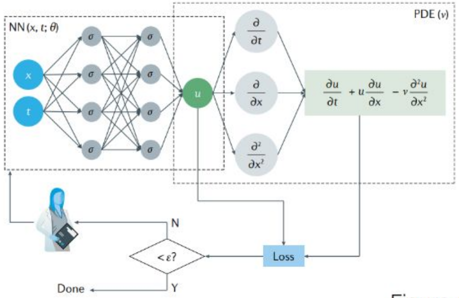
$$\min_{\theta} \ell(\mathbf{y},\, f \circ G_\theta(\mathbf{z})) + \lambda R \circ G_\theta(\mathbf{z})$$

No extra training data!

Neural implicit representation (NIR)

$$\mathbf{x} \approx \mathcal{D} \circ \overline{\mathbf{x}} \qquad \mathcal{D}: \text{discretization} \qquad \overline{\mathbf{x}}: \text{continuous function}$$

Physics-informed neural networks (PINN)



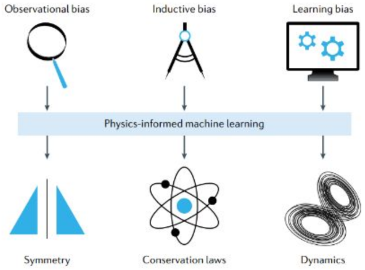


Figure credit: https://www.nature.com/articles/s42254-021-00314-5

Surveys

Deep Learning Techniques for Inverse Problems in Imaging

Gregory Ongie[*], Ajil Jalal[†], Christopher A. Metzler[‡]
Richard G. Baraniuk[§], Alexandros G. Dimakis[¶], Rebecca Willett[||]

https://arxiv.org/abs/2005.06001

But focused on linear IPs

Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing

Vishal Monga, *Senior Member, IEEE*, Yuelong Li, *Member, IEEE*, and Yonina C. Eldar, *Fellow, IEEE*

Focused on alg. unrolling

Untrained Neural Network Priors for Inverse Imaging Problems: A Survey

Deep Internal Learning: Deep Learning from a Single Input

Tom Tirer *Member,*

Understanding Untrained Deep Models for Inverse Problems: Algorithms and Theory

Ismail Alkhouri, Evan Bell, Avrajit Ghosh, Shijun Liang, Rongrong Wang, Saiprasad Ravishankar

Focused on single-instance methods

Theoretical Perspectives on Deep Learning Methods in Inverse Problems

Jonathan Scarlett, Reinhard Heckel, Miguel R. D. Rodrigues, Paul Hand, and Yonina C. Eldar

Focused on theories for linear IPs

This talk:

Solving Inverse Problems (IPs)
Using Pretrained Flow-Matching (FM) Models

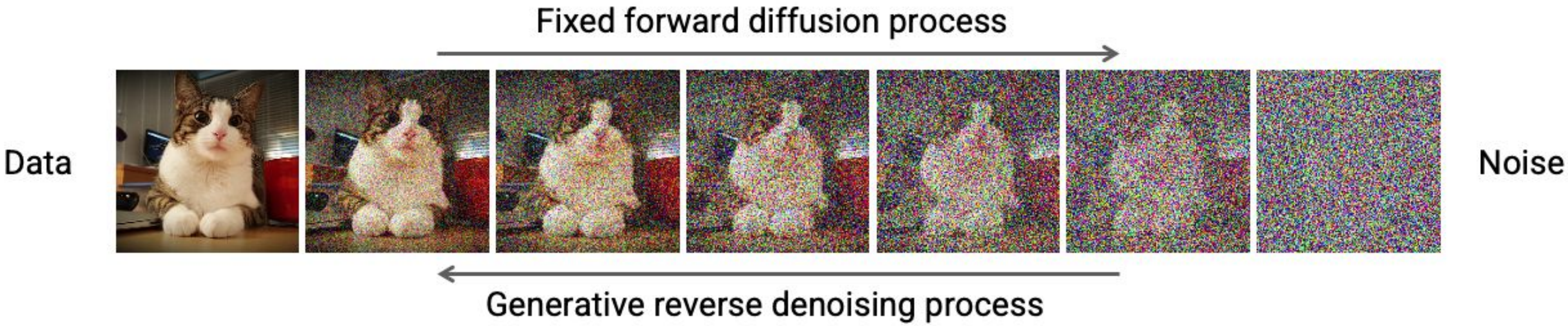
[Submitted on 30 Sep 2024]

A Survey on Diffusion Models for Inverse Problems

Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G. Dimakis, Mauricio Delbracio

Diffusion models

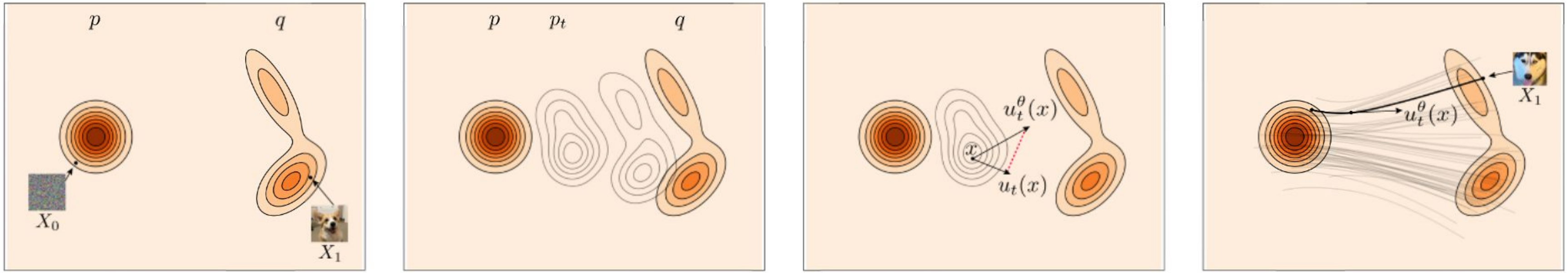
$$d\boldsymbol{x} = -\beta_t/2 \cdot \boldsymbol{x}dt + \sqrt{\beta_t}d\boldsymbol{w},$$



$$d\boldsymbol{x} = -\beta_t \left[\boldsymbol{x}/2 + \boxed{\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})}\right] dt + \sqrt{\beta_t}d\bar{\boldsymbol{w}}.$$

$$\cong \quad \boldsymbol{\varepsilon}_{\boldsymbol{\theta}}^{(t)}(\boldsymbol{x})$$

Flow-matching (FM) models



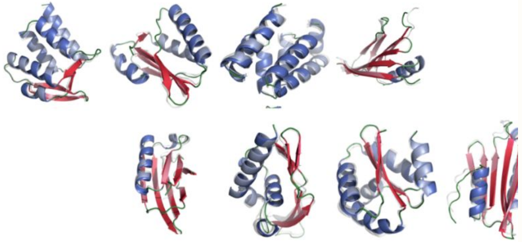
(a) Data. **(b)** Path design. **(c)** Training. **(d)** Sampling.

NeurIPS'25 tutorial: Flow Matching for Generative Modeling
https://neurips.cc/virtual/2024/tutorial/99531

Image credit: https://github.com/facebookresearch/flow_matching

Foundation FM-based generative models

Text-2-Video
MovieGen, Meta



Protein Generation
Huguet el a. 24

Image credit: https://neurips.cc/virtual/2024/tutorial/99531

Domain-specific vs. foundation FM models

FFHQ
70K

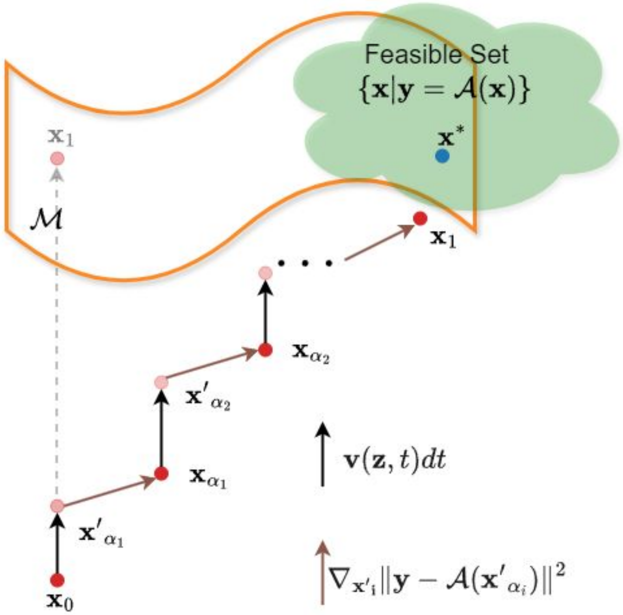
LSUN-Bedroom
3M

AFHQ
15K

LAION-400M image/text Status: Released	Formerly known as crawling@home (C@H), an openly accessible 400M image-text-pair dataset.
LAION5B image/text Status: Released	A dataset consisting of 5.85 billion CLIP-filtered image-text pairs, featuring several nearest neighbor indices, an improved web-interface for exploration and subset generation, and detection scores for watermark, NSFW, and toxic content detection.

How do people solve IPs with these pretrained generative models?

Solving IPs with foundation FM models - I



interleaving approach

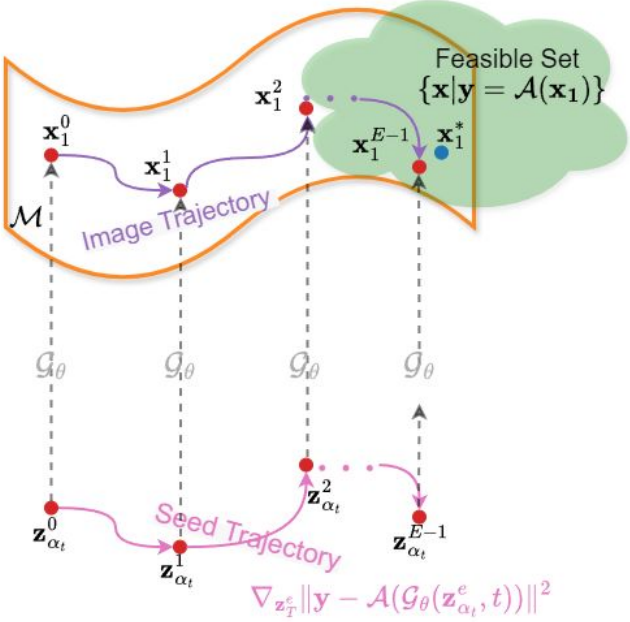
Algorithm 1 A sample algorithm of the interleaving approach

Input: ODE steps T, measurement $\boldsymbol{y}$, forward model $\mathcal{A}$

Initialize $\boldsymbol{z}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $i = 0$ to $T - 1$ **do**
 $t_i \leftarrow i/T$
 $\boldsymbol{v} \leftarrow \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, t_i)$ ▷ learned velocity
 $\boldsymbol{z}'_{i+1} \leftarrow \boldsymbol{z}_i + 1/T \cdot \boldsymbol{v}$ ▷ discrete integration
 $\boldsymbol{z}_{i+1} \leftarrow (\boldsymbol{y}, \mathcal{A})$-driven update of $\boldsymbol{z}'_{i+1}$ ▷ reducing $\ell(\boldsymbol{y}, \mathcal{A}(\boldsymbol{z}))$ starting from $\boldsymbol{z}'_{i+1}$
end for

Output: Estimated $\widehat{\boldsymbol{x}}$

Solving IPs with foundation FM models - II



plug-in approach

Algorithm 2 A sample algorithm of the plug-in approach

Input: Total iterations E, ODE steps T, measurement $\boldsymbol{y}$, forward model $\mathcal{A}$

Initialize $\boldsymbol{z}^{(0)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $e = 0$ to $E - 1$ **do**
    $\boldsymbol{z}_0 \leftarrow \boldsymbol{z}^{(e)}$
    for $i = 0$ to $T - 1$ **do** ▷ whole integration path
        $t_i \leftarrow i/T$
        $\boldsymbol{v} \leftarrow \boldsymbol{v_\theta}(\boldsymbol{z}_i, t_i)$
        $\boldsymbol{z}_{i+1} \leftarrow \boldsymbol{z}_i + 1/T \cdot \boldsymbol{v}$
    end for
    $\boldsymbol{z}^{(e+1)} \leftarrow \boldsymbol{z}^{(e)} - \eta^{(e)} \cdot \nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}_{i+1})$ ▷ reducing loss
end for

Output: Estimation $\widehat{\boldsymbol{x}} = \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z}^{(E-1)})$

Surprise?

	PSNR↑	SSIM↑	LPIPS↓	CLIPIQA↑
DIP	27.5854	0.7179	0.3898	0.2396
D-Flow (DS)	**28.1389**	**0.7628**	**0.2783**	**0.5871**
D-Flow (FD)	25.0120	0.7084	0.5335	0.3607
D-Flow (FD-S)	25.1453	0.6829	0.5213	0.3228
FlowDPS (DS)	22.1191	0.5603	<u>0.3850</u>	<u>0.5417</u>
FlowDPS (FD)	22.1404	0.5930	0.5412	0.2906
FlowDPS (FD-S)	22.0538	0.5920	0.5408	0.2913

Table 1: Comparison between foundation FM, domain-specific FM, and untrained priors for Gaussian deblurring the on AFHQ-Cat dataset (resolution: 256×256). DS: domain-specific FM; FD: foundation FM; FD-S: strengthened foundation FM; DIP: deep image prior. **Bold**: best, & <u>underline</u>: second best, for each metric/column. The foundation model is Stable Diffusion V3 here.

Foundation priors <<

Domain-specific, and even untrained priors

Attempts to strengthen the priors don't quite work

- Interleaving Approach
  - Prompt as guidance
- Plug-in Approach
  - Mixture initialization:

$z_0 = \sqrt{\alpha} \cdot y_0 + \sqrt{1-\alpha} \cdot z$ $\quad y_0$ is a reversed seed

  - Gaussian regularization

Foundation model: Stable Diffusion 3
Gaussian deblur noise free on AFHQ-Cat dataset

	PSNR↑	SSIM↑	LPIPS↓	CLIPIQA↑
DIP	27.5854	0.7179	0.3898	0.2396
D-Flow (DS)	**28.1389**	**0.7628**	**0.2783**	**0.5871**
D-Flow (FD)	25.0120	0.7084	0.5335	0.3607
D-Flow (FD-S)	25.1453	0.6829	0.5213	0.3228
FlowDPS (DS)	22.1191	0.5603	0.3850	0.5417
FlowDPS (FD)	22.1404	0.5930	0.5412	0.2906
FlowDPS (FD-S)	22.0538	0.5920	0.5408	0.2913

Our contributions

Enhance the foundation FM priors in:

- Simple-distortion settings
- Few-shot scientific settings

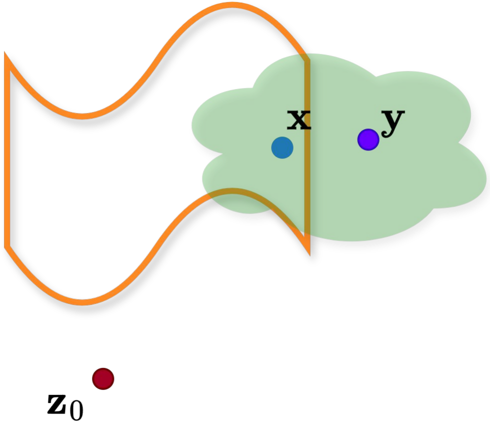
Simple-distortion settings

x

Gaussian Blur

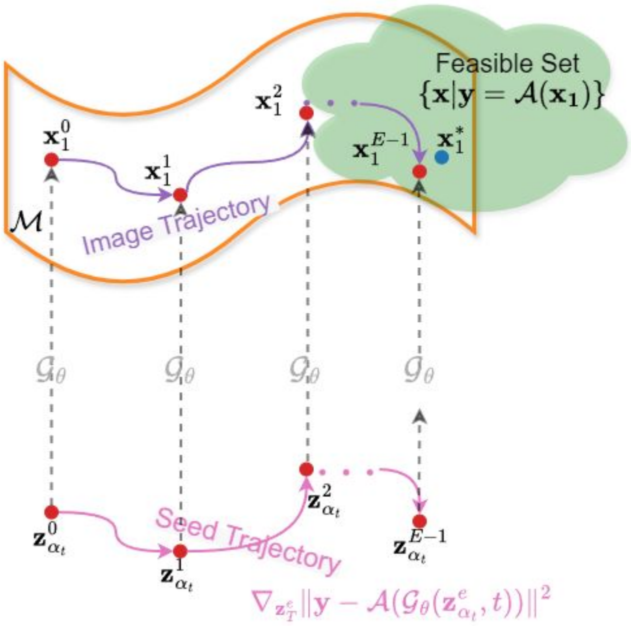
Down Sampling

Motion Blur

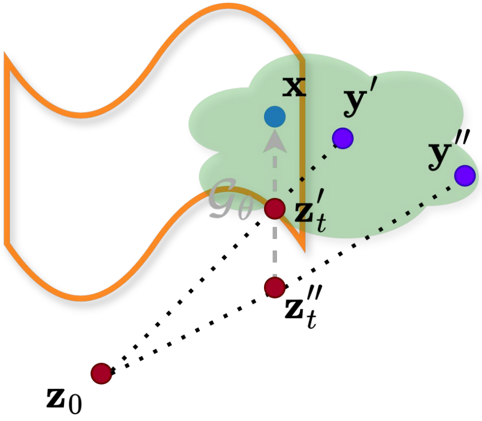


$$x = y + \varepsilon$$

A simple warm-start idea



plug-in approach



$$\boldsymbol{z}_t = \alpha_t \boldsymbol{x} + \beta_t \boldsymbol{z} \quad \text{where } \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}),$$

$$\boldsymbol{z}_t \approx \alpha_t \boldsymbol{y} + \beta_t \boldsymbol{z} \quad \text{where } \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

$$\boxed{\min_{\boldsymbol{z}, t \in [0,1]} \ \ell(\boldsymbol{y}, \mathcal{A} \circ \mathcal{G}_{\boldsymbol{\theta}}(\alpha_t \boldsymbol{y} + \beta_t \boldsymbol{z}, t))}$$

warm-start

- **Sharp Gaussian regularization**

Boosted priors >= untrained priors

- Gaussian Blur and Motion Blur on DIV2K 512 × 512 with additive Gaussian noise
- **Bold**: best, <u>under</u>: second best; -P: prompt, -W: warm-start only

method	Gaussian Blur						Motion Blur					
	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	CLIPIQA ↑	MUSIQ ↑	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	CLIPIQA ↑	MUSIQ ↑
DIP	25.23	0.70	<u>0.43</u>	0.18	**0.38**	32.54	24.75	0.68	0.45	0.20	0.35	32.59
FlowChef-P	20.41	0.49	0.62	0.34	0.23	16.68	21.27	0.54	0.57	0.32	0.34	19.76
FlowChef	20.41	0.49	0.62	0.34	0.23	16.68	21.28	0.54	0.57	0.32	0.34	19.82
FlowDPS-P	20.23	0.46	0.58	0.29	0.32	35.90	21.07	0.51	0.54	0.26	0.39	39.56
FlowDPS	20.22	0.45	0.61	0.30	0.20	30.51	21.05	0.50	0.58	0.27	0.26	34.21
D-Flow	23.64	0.64	0.52	0.17	<u>0.37</u>	**53.03**	25.21	0.70	0.47	0.17	**0.42**	**53.78**
FMPlug-W	<u>26.05</u>	<u>0.72</u>	<u>0.43</u>	**0.16**	0.29	36.66	<u>26.83</u>	<u>0.74</u>	<u>0.40</u>	<u>0.14</u>	0.36	46.95
FMPlug	**26.26**	**0.73**	**0.41**	**0.16**	0.28	<u>38.14</u>	**27.38**	**0.78**	**0.36**	**0.12**	**0.42**	<u>51.71</u>

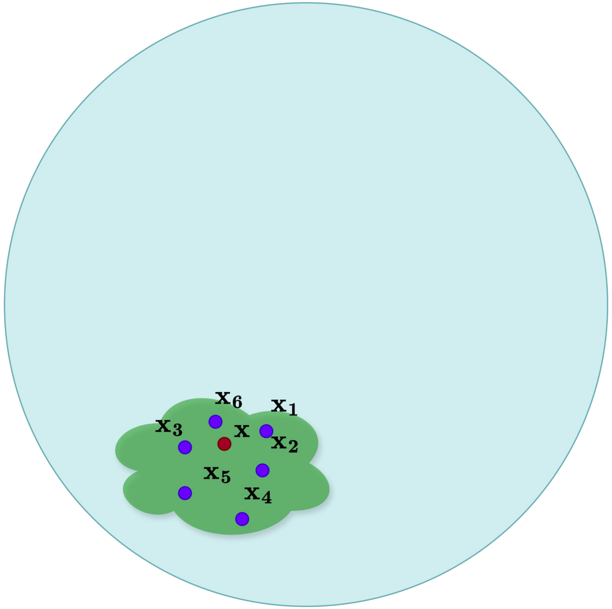
~ Domain-specific priors

- Gaussian Deblur and Super Resolution 4X on AFHQ-Cat 256 × 256 with additive Gaussian noise
- **Bold**: best, <u>under</u>: second best

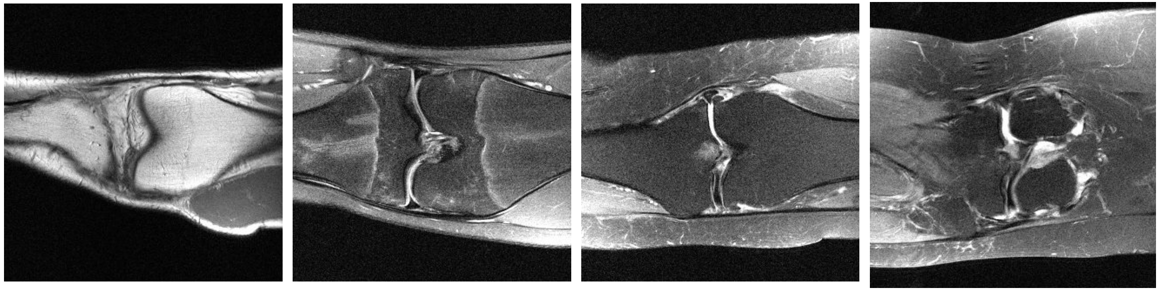
	Super Resolution $4\times$						**Gaussian Blur**					
	LPIPS↓	PSNR↑	SSIM↑	DIST↓	CLIPIQA↑	MUSIQ↑	LPIPS↓	PSNR↑	SSIM↑	DIST↓	CLIPIQA↑	MUSIQ↑
DIP	0.36	28.17	0.76	0.21	0.25	28.12	0.36	27.92	0.75	0.23	0.26	23.94
OT-ODE (DS)	**0.19**	26.43	0.74	0.90	<u>0.59</u>	**64.63**	**0.19**	27.67	0.75	0.89	<u>0.62</u>	**63.82**
OT-ODE (FD)	-	-	-	-	-	-	-	-	-	-	-	-
PnP-Flow (DS)	0.24	27.45	**0.80**	0.82	0.52	51.95	0.31	<u>28.70</u>	**0.79**	0.77	**0.66**	40.26
PnP-Flow (FD	-	-	-	-	-	-	-	-	-	-	-	-
FlowDPS (DS)	0.24	<u>28.56</u>	0.79	<u>0.14</u>	0.57	55.63	0.38	22.27	0.56	**0.20**	0.52	52.42
FlowDPS (FD	0.37	24.45	0.74	0.27	**0.63**	27.96	0.55	22.11	0.59	0.38	0.28	15.10
D-Flow (DS)	0.27	25.81	0.69	0.82	0.52	<u>57.74</u>	<u>0.20</u>	28.41	0.77	0.87	0.61	<u>59.29</u>
D-Flow (FD)	0.53	24.64	0.67	0.31	0.31	45.27	0.56	24.42	0.62	<u>0.21</u>	0.30	49.12
FMPlug (DS)	<u>0.22</u>	27.52	0.79	**0.12**	<u>0.61</u>	61.21	0.36	27.44	0.75	0.77	0.24	31.19
FMPlug (FD)	0.33	**28.85**	**0.80**	0.22	0.31	28.77	0.35	**29.00**	**0.79**	0.23	0.24	30.58

Few-shot scientific settings

- Scientific imaging:
  - Usually with narrow image domain
  - A few available high-quality samples only (**cannot support domain-specific generative models**)



- Example: Knee MRI

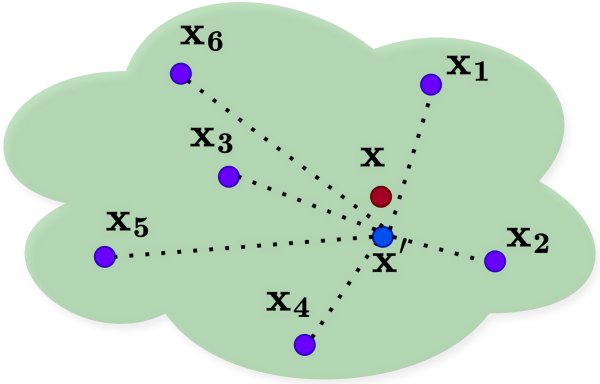


Few-shot scientific settings

- Assume a few instances, some of which are close to the target

$$\{\boldsymbol{x}_k\}_{\mathrm{k=1,2,...,K}}$$

- Consider sparse combinations of them for warm-start



$$\min_{\boldsymbol{z}\in\mathbb{S}_\varepsilon^{d-1}(\mathbf{0},\sqrt{d}),t\in[0,1],\boldsymbol{w}} \ \ell(\boldsymbol{y},\mathcal{A}\circ\mathcal{G}_{\boldsymbol{\theta}}(\alpha_t(\textstyle\sum_{k=1}^{K} w_k\boldsymbol{x}_k)+\beta_t\boldsymbol{z},t)) \quad \text{s.t. } \boldsymbol{w}\in\Delta^K$$

Preliminary results (<= instances)

	LIS		MRI (4×)	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DIP	28.72	0.96	18.35	0.39
D-Flow	17.15	0.66	8.94	0.15
FMPlug	**31.83**	**0.97**	**23.26**	**0.48**
Red-diff	36.55	0.98	28.71	0.62

With domain-specific model

LIS: linear inverse scattering
MRI: compressive-sensing MRI

Take-home messages

- Naively applying recent **foundation FM (i.e., generative) models for IPs** can be **very suboptimal , even poorer than untrained priors**

- We can significantly boost the performance by **careful initialization** and proper regularization, with **instance guidance**

More details

[Submitted on 20 Nov 2025]

Saving Foundation Flow-Matching Priors for Inverse Problems

Yuxiang Wan, Ryan Devera, Wenjie Zhang, Ju Sun

Foundation flow-matching (FM) models promise a universal prior for solving inverse problems (IPs), yet today they trail behind domain-specific or even untrained priors. How can we unlock their potential? We introduce FMPlug, a plug-in framework that redefines how foundation FMs are used in IPs. FMPlug combines an instance-guided, time-dependent warm-start strategy with a sharp Gaussianity regularization, adding problem-specific guidance while preserving the Gaussian structures. This leads to a significant performance boost across image restoration and scientific IPs. Our results point to a path for making foundation FM models practical, reusable priors for IP solving.

Related

DMPlug: A Plug-in Method for Solving Inverse Problems with Diffusion Models

Hengkang Wang, Xu Zhang, Taihui Li, Yuxiang Wan, Tiancong Chen, Ju Sun

Pretrained diffusion models (DMs) have recently been popularly used in solving inverse problems (IPs). The existing methods mostly interleave iterative steps in the reverse diffusion process and iterative steps to bring the iterates closer to satisfying the measurement constraint. However, such interleaving methods struggle to produce final results that look like natural objects of interest (i.e., manifold feasibility) and fit the measurement (i.e., measurement feasibility), especially for nonlinear IPs. Moreover, their capabilities to deal with noisy IPs with unknown types and levels of measurement noise are unknown. In this paper, we advocate viewing the reverse process in DMs as a function and propose a novel plug-in method for solving IPs using pretrained DMs, dubbed DMPlug. DMPlug addresses the issues of manifold feasibility and measurement feasibility in a principled manner, and also shows great potential for being robust to unknown types and levels of noise. Through extensive experiments across various IP tasks, including two linear and three nonlinear IPs, we demonstrate that DMPlug consistently outperforms state-of-the-art methods, often by large margins especially for nonlinear IPs. The code is available at this https URL.