Diffusion Models for Inverse Problems

ICMS Workshop on "Interfacing Bayesian statistics, machine learning, applied analysis, and blind and semi-blind imaging inverse problems"

Jiaming Song
NVIDIA Research

https://tsong.me

NVIDIA

Diffusion models as powerful image generators

A highly detailed digital painting of a portal in a mystic forest with many beautiful trees. A person is standing in front of the portal.

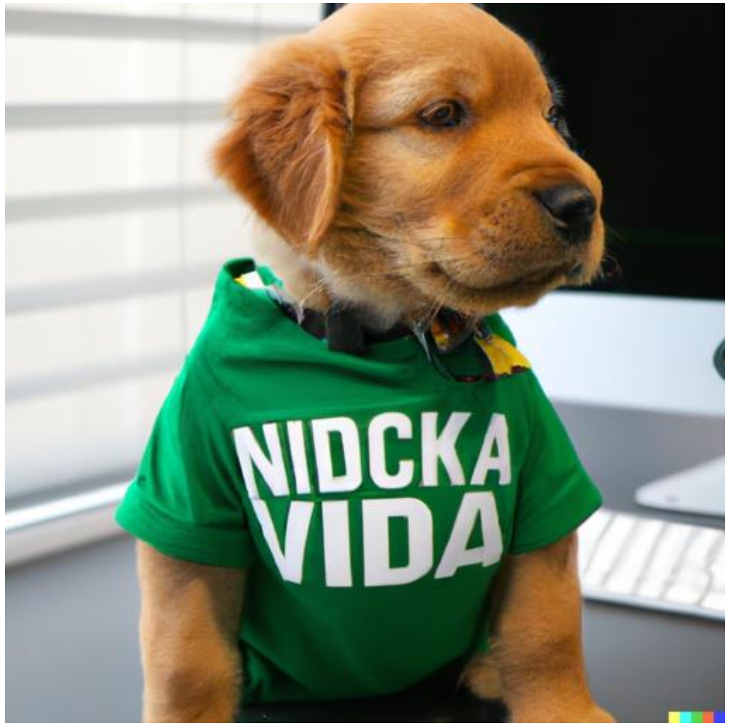
A highly detailed zoomed-in digital painting of a cat dressed as a witch wearing a wizard hat in a haunted house, artstation.

An image of a beautiful landscape of an ocean. There is a huge rock in the middle of the ocean. There is a mountain in the background. Sun is setting.

[Balaji et al., 2022] eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Experts (NVIDIA)

A photo of a golden retriever puppy wearing a green shirt.
The shirt has text that says, “NVIDIA rocks”.
Background office. 4k dslr.

Stable Diffusion



DALL·E 2

eDiff-I

[Balaji et al., 2022] eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Experts (NVIDIA)

[Balaji et al., 2022] eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Experts (NVIDIA)

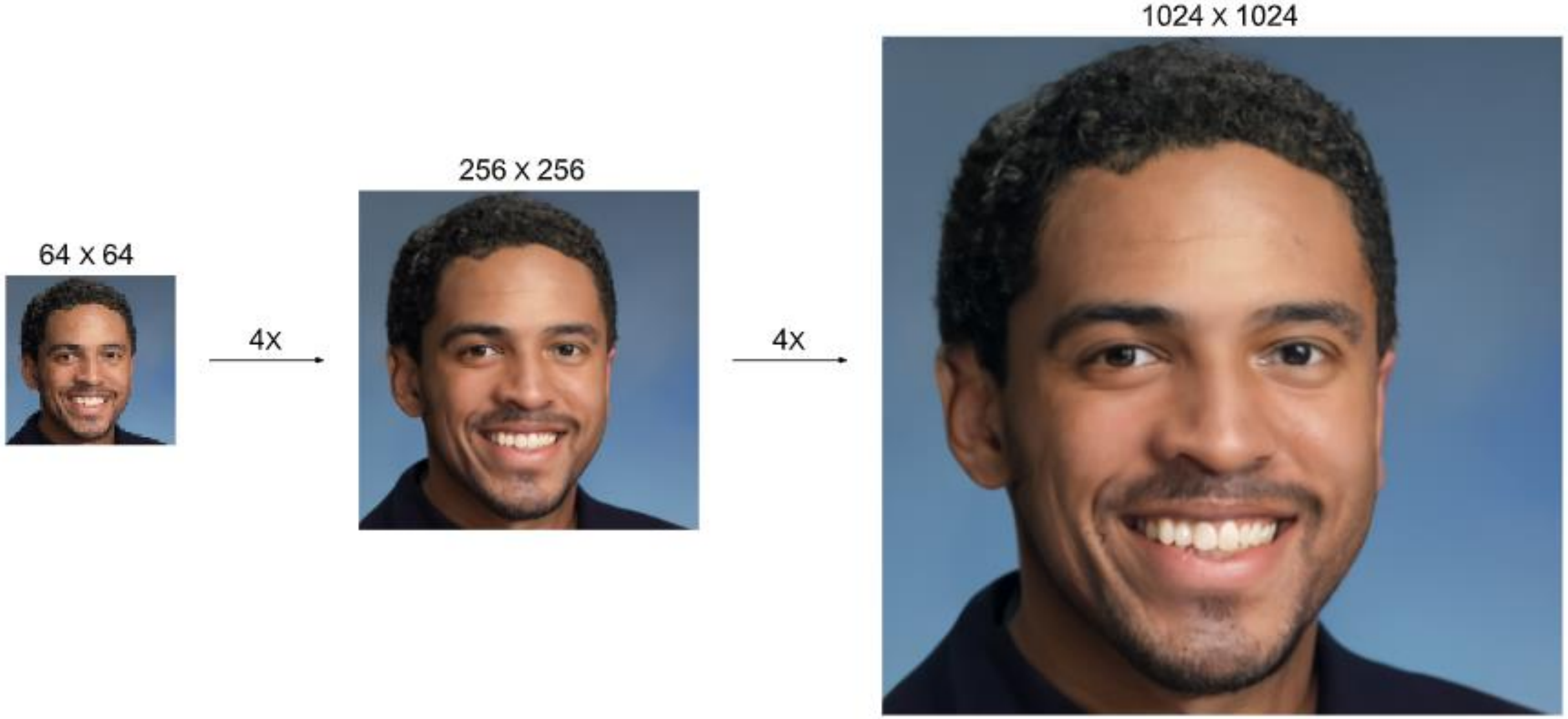
A digital painting of a half-frozen lake near mountains under a full moon and aurora. A boat is in the middle of the lake. Highly detailed.

[Balaji et al., 2022] eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Experts (NVIDIA)

“A {X} photo / painting of a penguin working as a fruit vendor in a tropical village

[Balaji et al., 2022] eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Experts (NVIDIA)

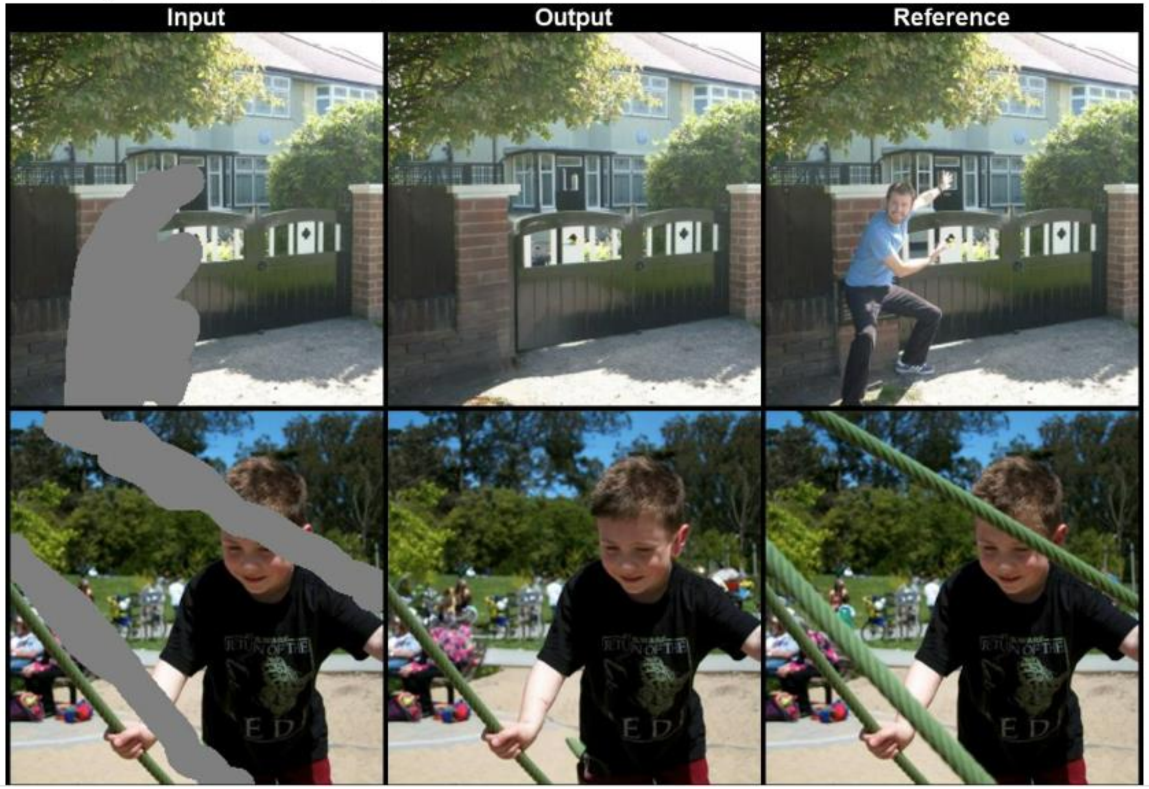
Conditional diffusion model for many image processing problems



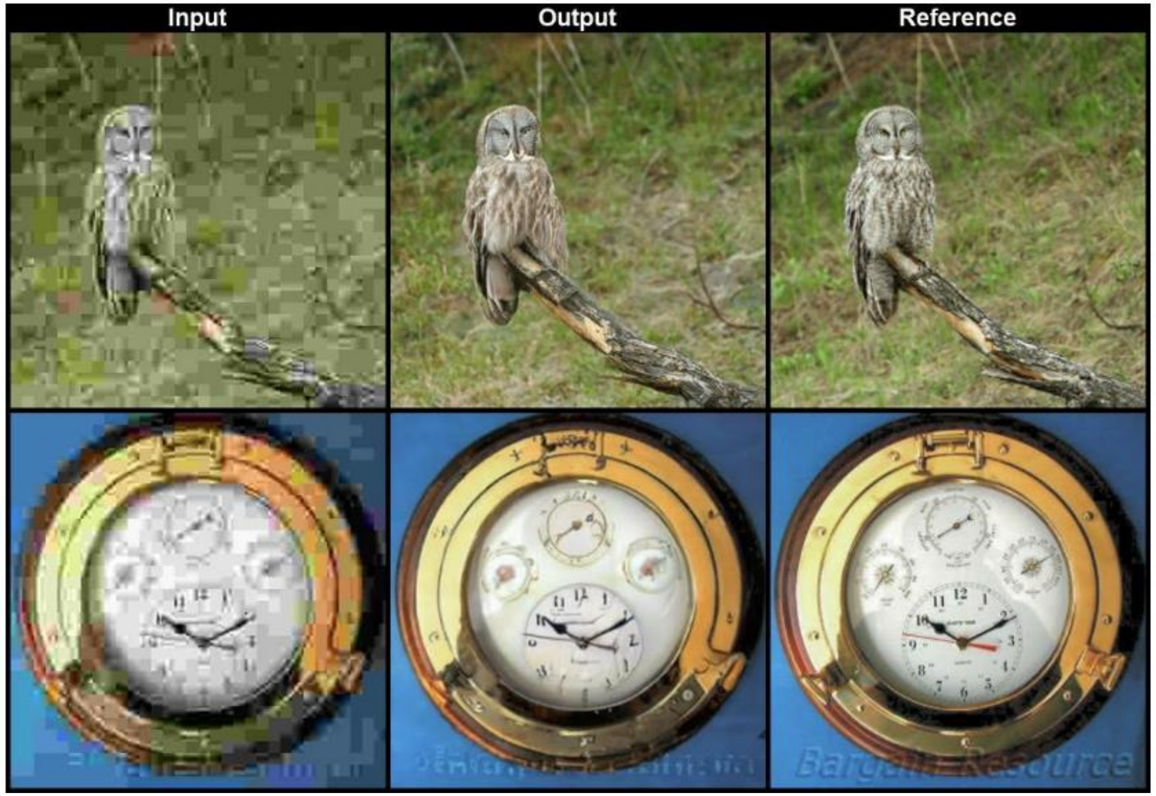
Super-resolution

https://iterative-refinement.github.io

Conditional diffusion model for many image processing problems



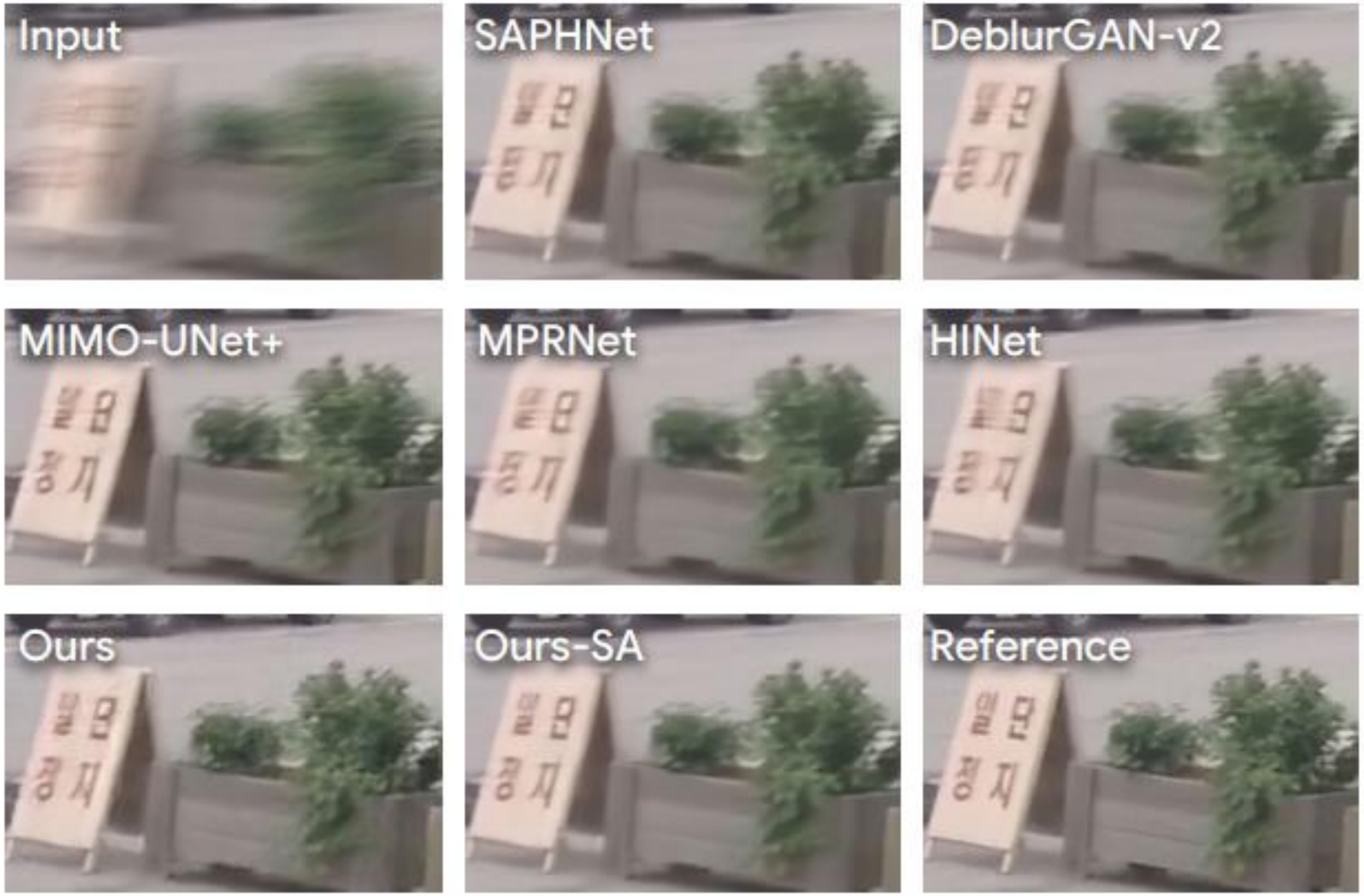
Inpainting



JPEG (QF = 5) Restoration

https://iterative-refinement.github.io/palette/

Conditional diffusion model for many image processing problems



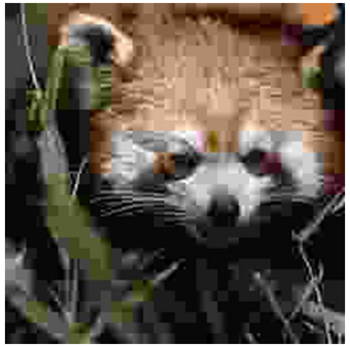
Blind deblurring

[CVPR 2022] Deblurring via Stochastic Refinement

Why not use a conditional diffusion model everywhere?

The base model was trained using 256 NVIDIA A100 GPUs, while the two super-resolution models were trained with 128 NVIDIA A100 GPUs each.

Training is expensive (Source: eDiff-I)



JPEG Restoration

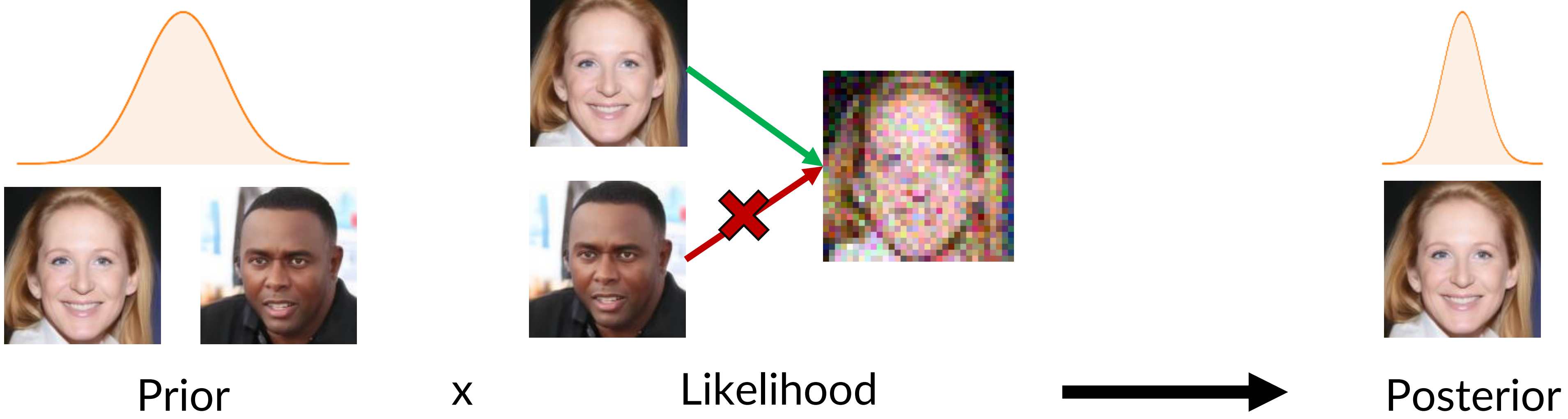
JPEG + Super-resolution

JPEG + Super-res + Inpainting

Many conditioning tasks

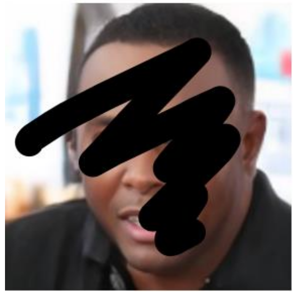
Super-resolution with Plug-and-Play

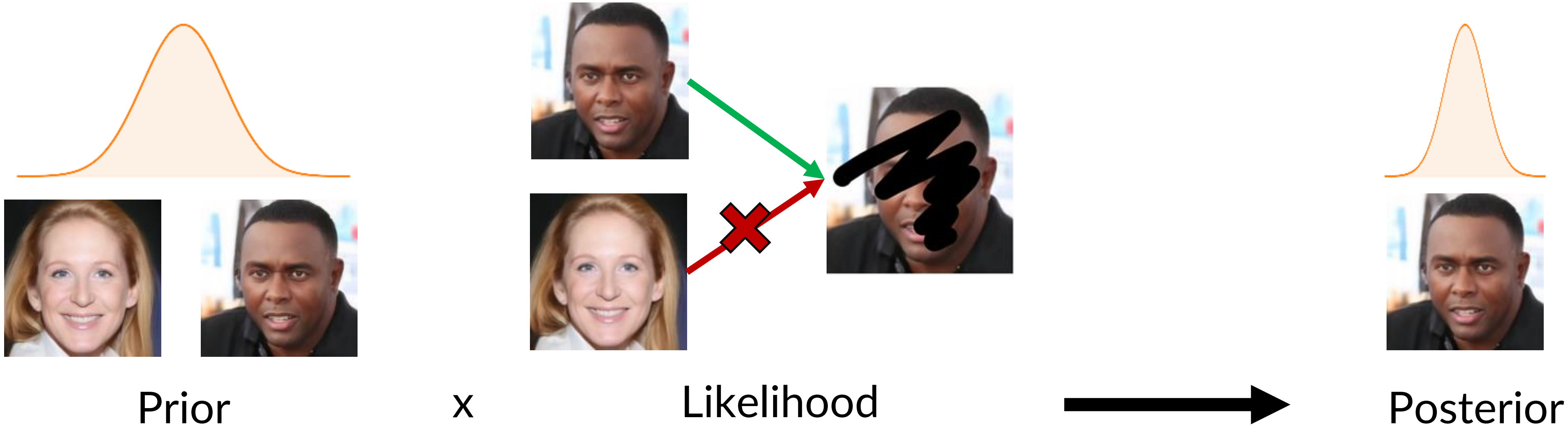
Goal: denoise and super-resolve an image



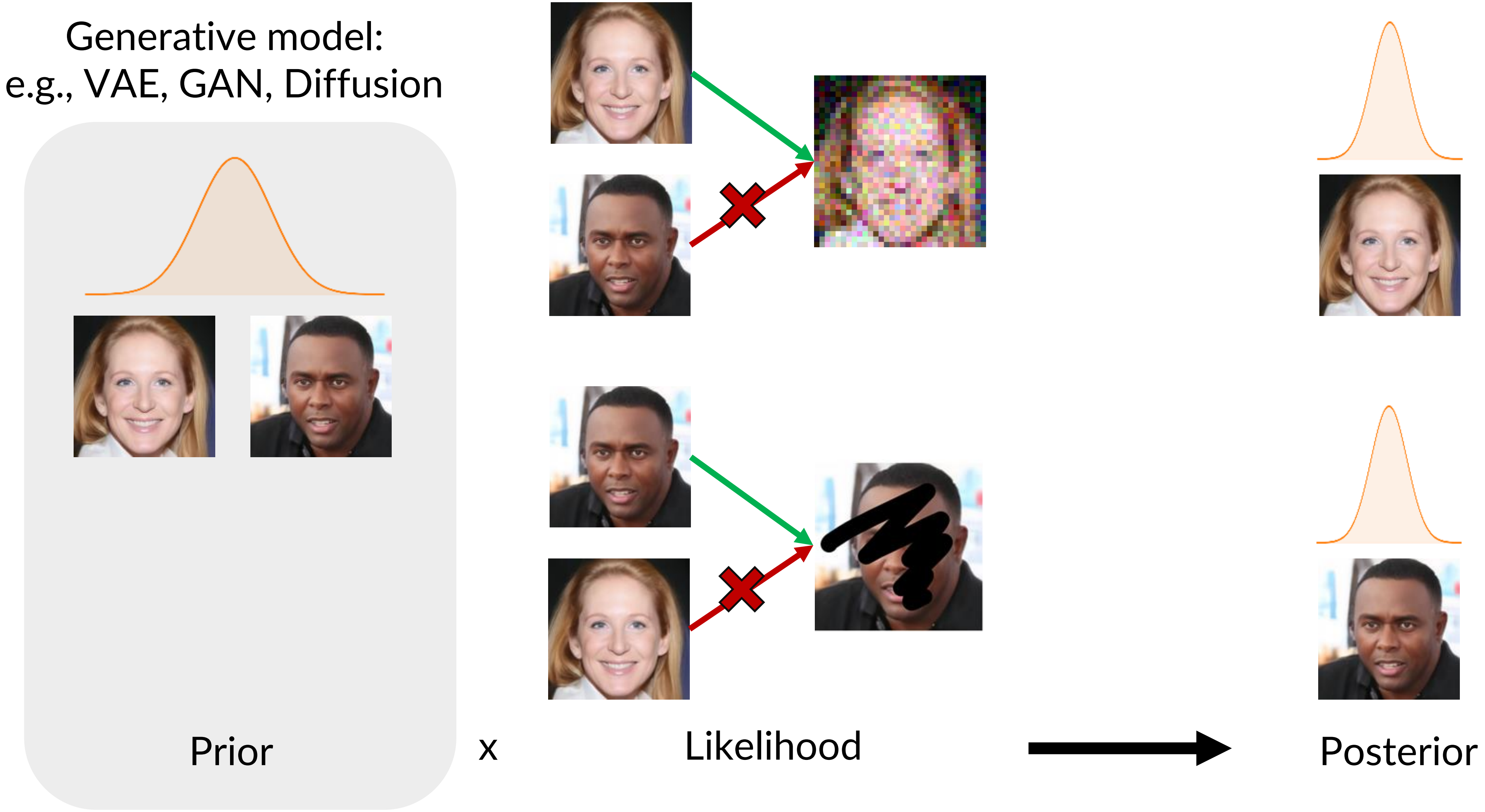
Inpainting with Plug-and-Play

Goal: recover the masked region of an image



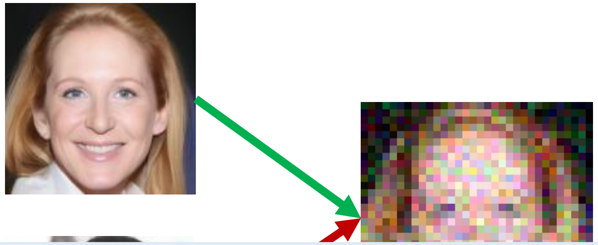


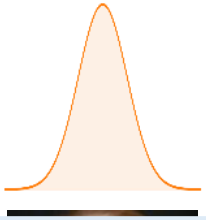
Inverse problems with Plug-and-Play



Inverse problems with Plug-and-Play

Generative model:
e.g., VAE, GAN, Diffusion





How can we use generic diffusion models for efficiently solving general inverse problems?

Prior x Likelihood → Posterior

Diffusion Models for Inverse Problems
Example 1. JPEG Restoration + Inpainting

Input

ΠGDM Output

Diffusion Models for Inverse Problems
Example 2. JPEG Restoration + Super-resolution

Input

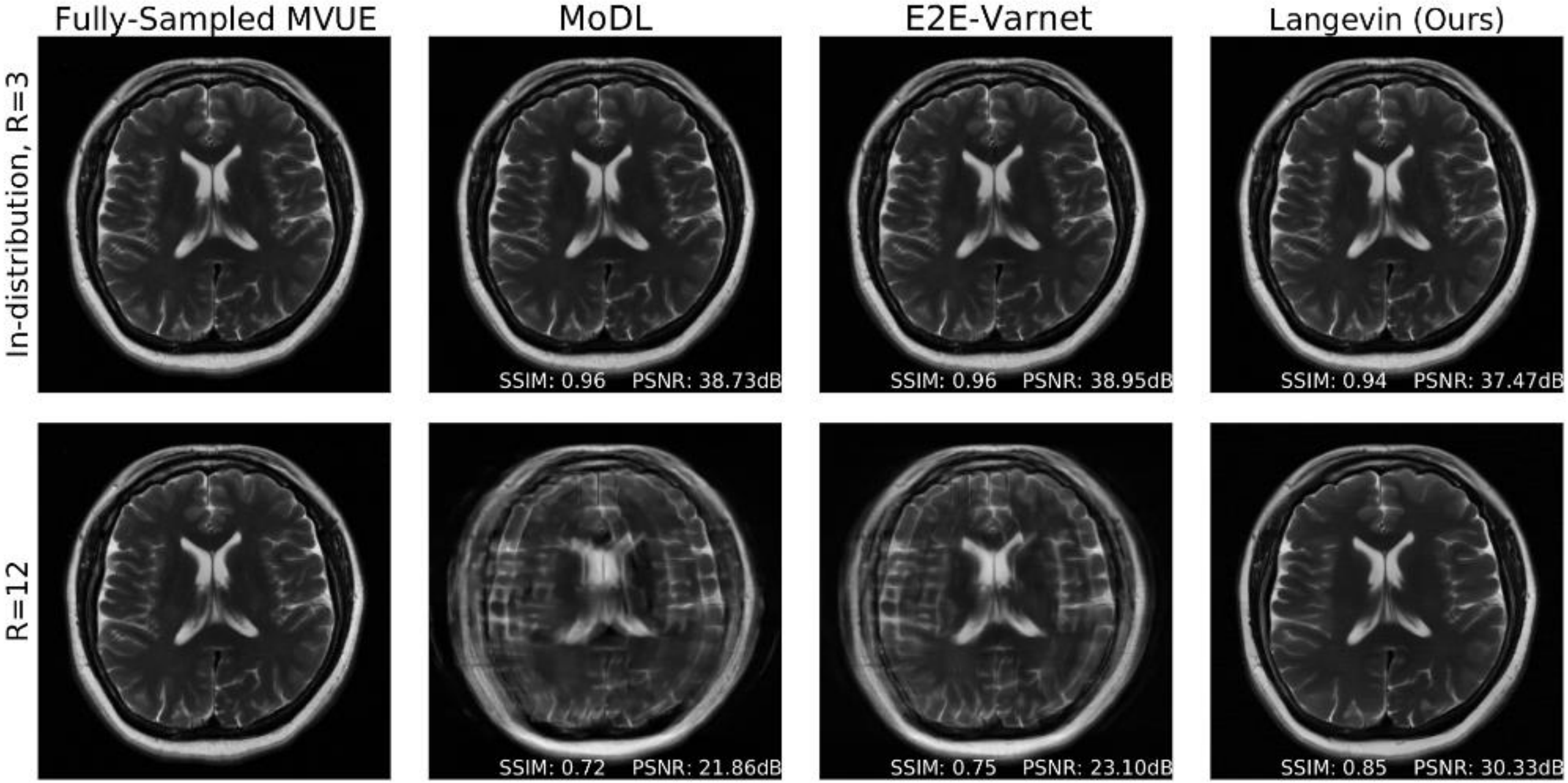
ΠGDM Output

Diffusion Models for Inverse Problems
Example 3. JPEG Restoration + Super-resolution + Inpainting

Input

ΠGDM Output

Diffusion Models for Inverse Problems
Example 4. Medical Imaging Problems



[NeurIPS 2021] Robust Compressed Sensing MRI with Deep Generative Priors

Roadmap

I. Overview of DDIM

II. Denoising Diffusion Restoration Models

Solving noisy, linear inverse problems on images, quickly.

III. PhysDiff: Guided Human Motion Diffusion Model

Enforce physical constraints in diffusion models.

IV. Pseudoinverse-Guided Diffusion Models

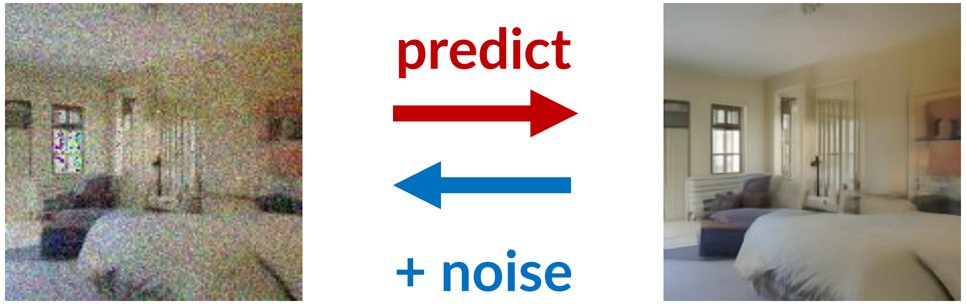
First to achieve SOTA performance comparable to domain-specific diffusion models.

Overview of (denoising) diffusion models

Learning with regression:

$$\|\boldsymbol{x}_0 - \underbrace{D(\boldsymbol{x}_t; \sigma_t)}_{\text{“predict } \boldsymbol{x}_0 \text{ from } \boldsymbol{x}_t\text{”}} \|_2^2$$

Noisy image **Clean image**



$\boldsymbol{x}_t = \boldsymbol{x}_0 + \sigma_t \epsilon$ $\boldsymbol{x}_0$

[Gaussian noise]

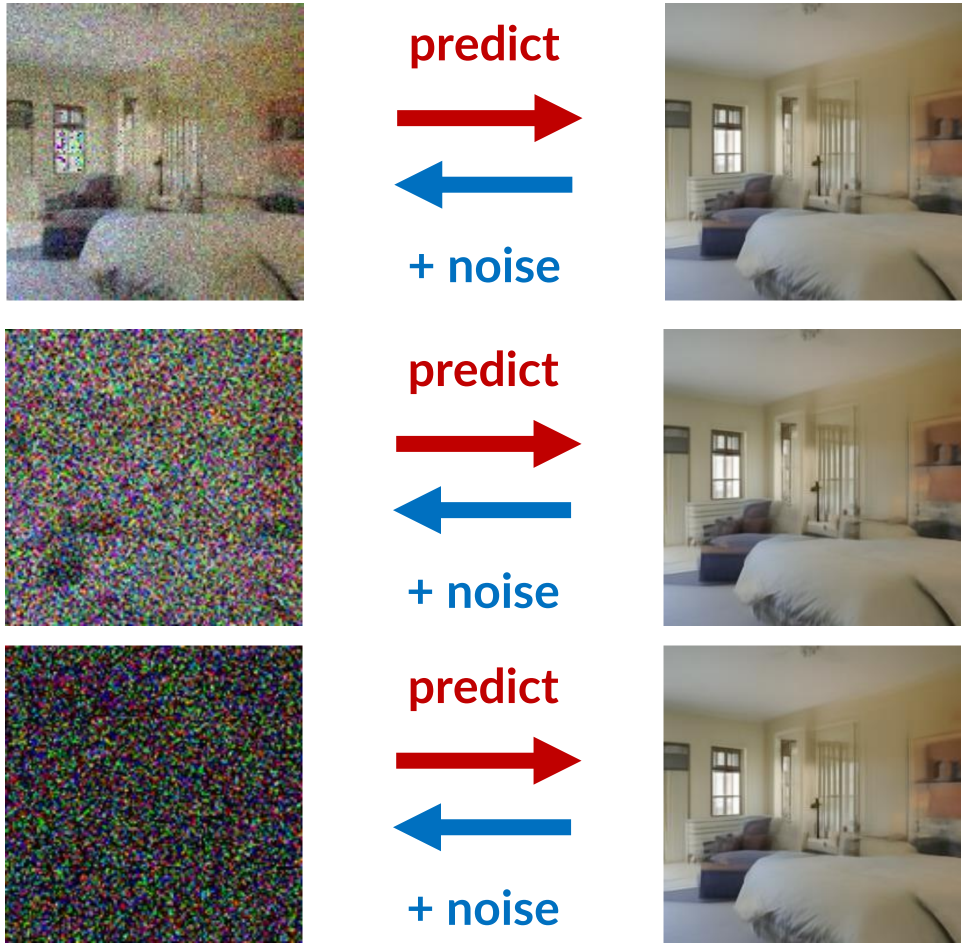
Overview of (denoising) diffusion models

Learning with regression: $\|\boldsymbol{x}_0 - \underbrace{D(\boldsymbol{x}_t;\sigma_t)}_{\text{“predict } \boldsymbol{x}_0 \text{ from } \boldsymbol{x}_t\text{”}}\|_2^2$

[Small noise]

[Mid noise]

[High noise]



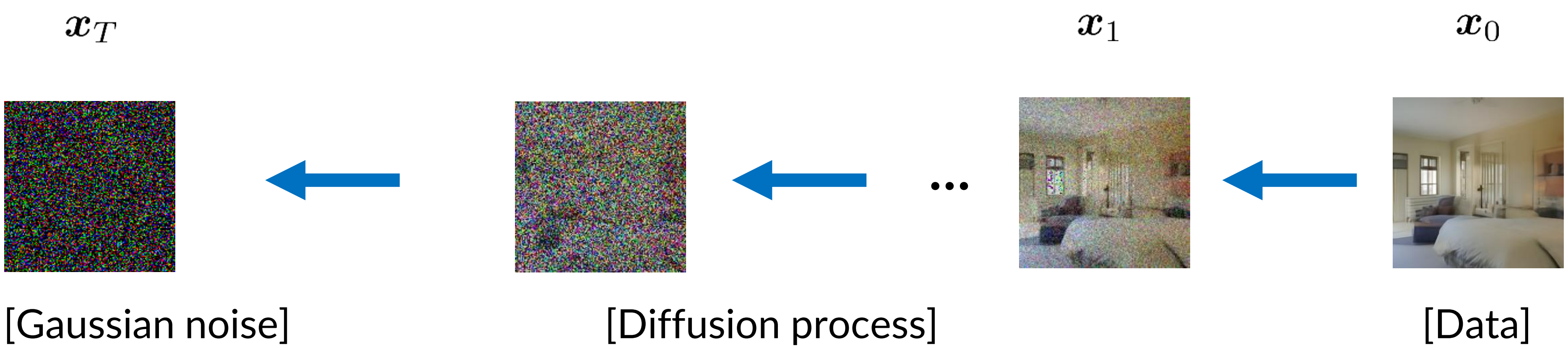
Overview of (denoising) diffusion models

Learning with regression: $\|\boldsymbol{x}_0 - \underbrace{D(\boldsymbol{x}_t; \sigma_t)}_{\text{"predict } \boldsymbol{x}_0 \text{ from } \boldsymbol{x}_t\text{"}}\|_2^2$

Forward diffusion process:

$\boldsymbol{x}_0 \to \boldsymbol{x}_1 \to \cdots \to \boldsymbol{x}_T$

[More and more noisy]



[Timestep tied to noise levels]

Overview of (denoising) diffusion models

Learning with regression: $\|\boldsymbol{x}_0 - \underbrace{D(\boldsymbol{x}_t; \sigma_t)}_{\text{"predict } \boldsymbol{x}_0 \text{ from } \boldsymbol{x}_t\text{"}} \|_2^2$

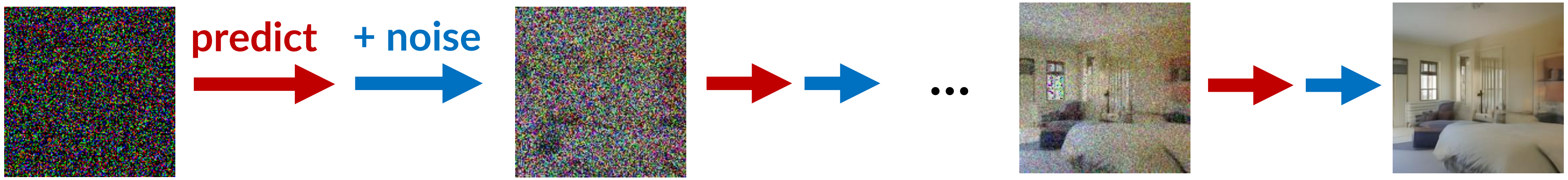
Forward diffusion process:

$\boldsymbol{x}_0 \to \boldsymbol{x}_1 \to \cdots \to \boldsymbol{x}_T$

[More and more noisy]

Reverse diffusion process:

$$\boldsymbol{x}_T \to \boldsymbol{x}_{T-1} \to \cdots \to \boldsymbol{x}_1 \to \boldsymbol{x}_0$$

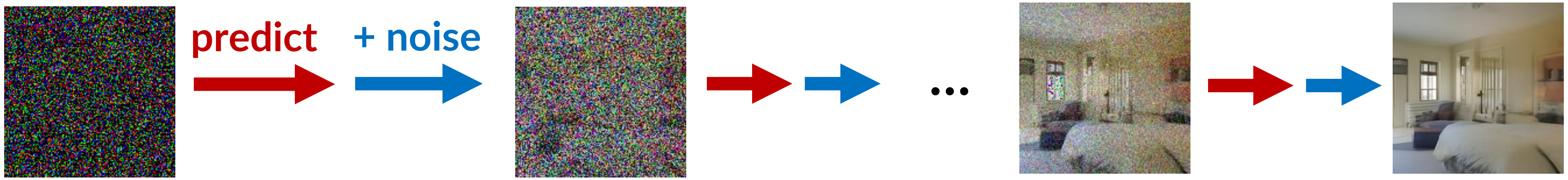


[Gaussian noise] [Gradual denoising process] [Sample]

Overview of (denoising) diffusion models

Reverse diffusion process:

$$\boldsymbol{x}_T \rightarrow \boldsymbol{x}_{T-1} \rightarrow \cdots \rightarrow \boldsymbol{x}_1 \rightarrow \boldsymbol{x}_0$$



[Gaussian noise] [Gradual denoising process] [Sample]

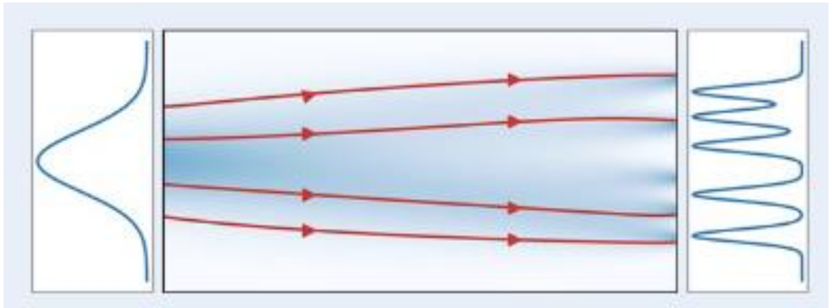
Connections to denoising score matching and score SDEs

“Denoise”

$$-\sigma_t \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \frac{\mathbf{x}_t - D(\mathbf{x}_t; \sigma_t)}{\sigma_t}$$

“Score”

$p_T(\mathbf{x}_T)$ $p_0(\mathbf{x}_0)$



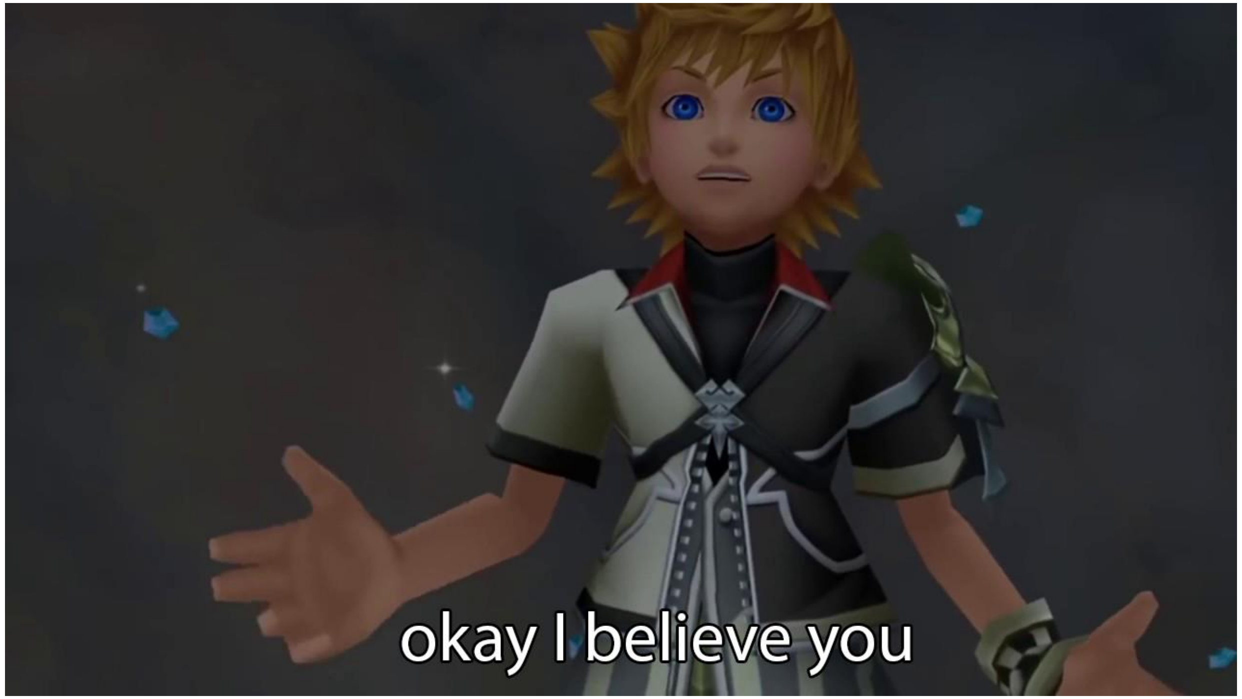
$$d\mathbf{x} = \underbrace{-\dot{\sigma}_t \sigma_t \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) dt}_{\text{Probabilistic ODE}} \underbrace{- \beta_t \sigma_t^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) dt + \sqrt{2\beta_t} \sigma_t d\omega_t}_{\text{Langevin process}}$$

Denoising Diffusion Implicit Models

DDIM: A first-order solver for the SDE

If a method works for general distributions, then it should work if dist. only has 1 datapoint.

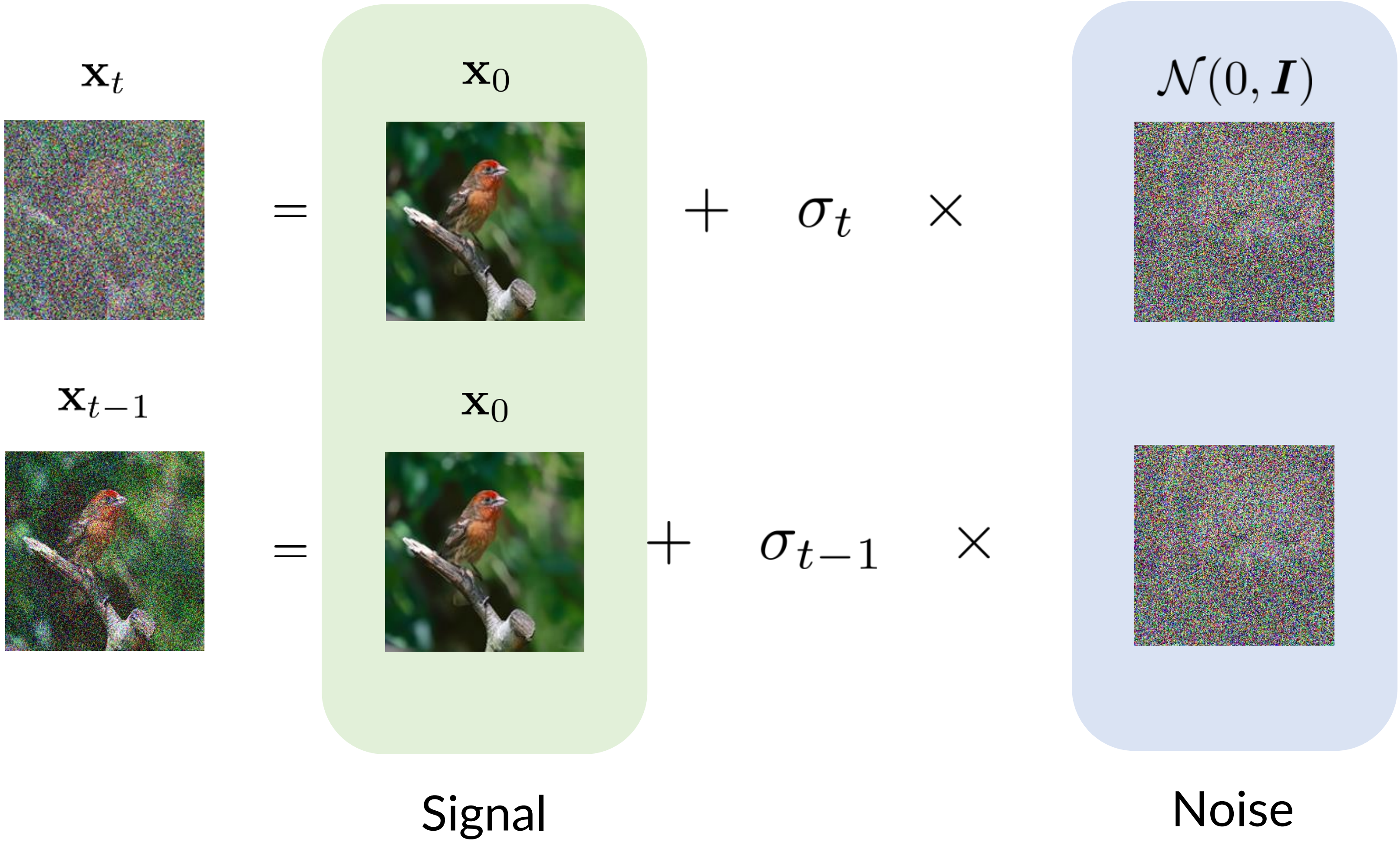
Going to explain the idea for just 1 datapoint, but it also works for general distributions
The general case is related to with Variational inference, Fokker-Planck Equations, Schrodinger bridge …



[ICLR 2021] Denoising Diffusion Implicit Models

Denoising Diffusion Implicit Models

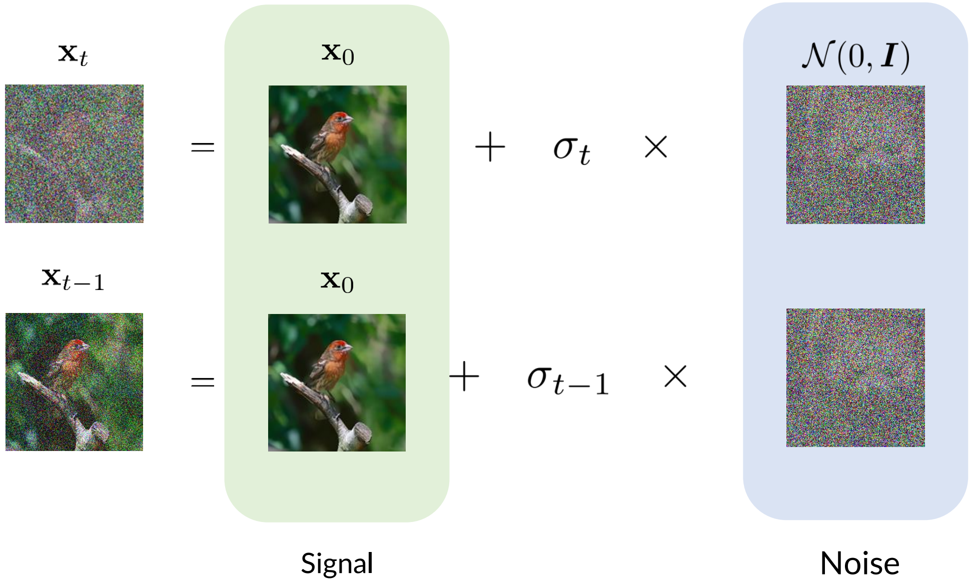
DDIM: A first-order solver for the SDE



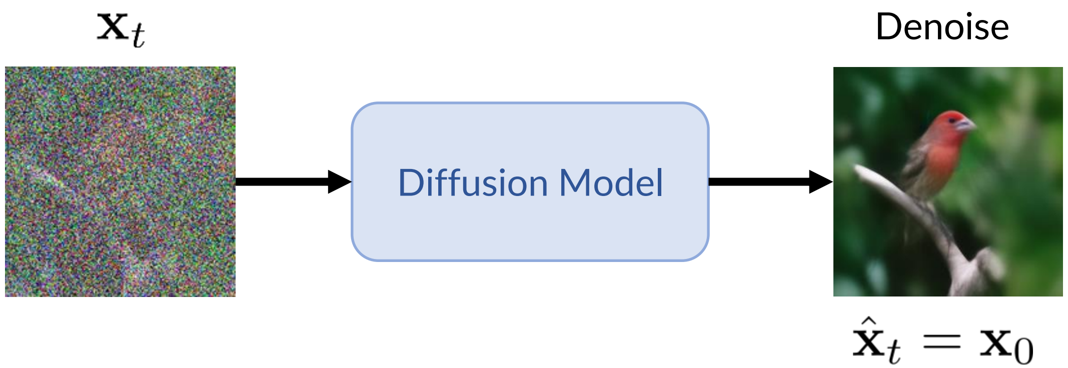
[ICLR 2021] Denoising Diffusion Implicit Models

Denoising Diffusion Implicit Models

DDIM: A first-order solver for the SDE



Decompose "signal" and "noise" linearly.



MMSE is always $\mathbf{x}_0$

Distribution of 1 datapoint.

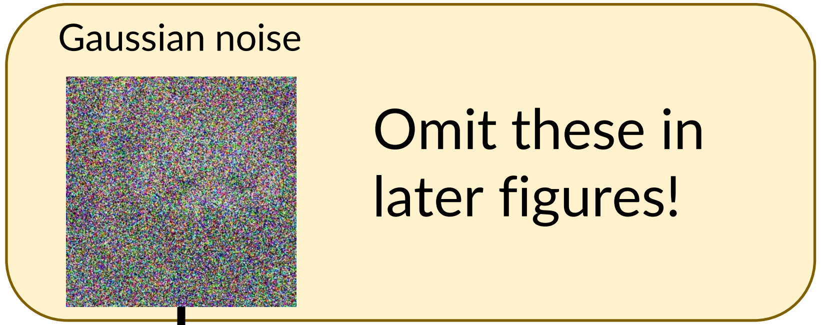
$$\mathcal{N}(0, a^2) + \mathcal{N}(0, b^2) = \mathcal{N}(0, a^2 + b^2)$$

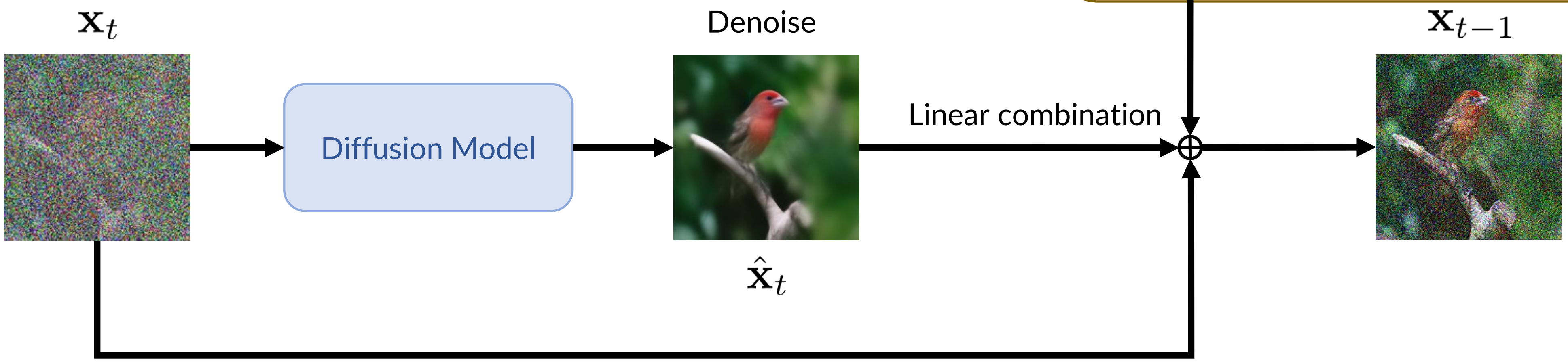
Summing iid. Gaussians gives a Gaussian.

[ICLR 2021] Denoising Diffusion Implicit Models

Denoising Diffusion Implicit Models

DDIM: A first-order solver for the SDE





Linearly combine input, denoise, standard Gaussian noise to get output.

$$A\mathbf{x}_t + B\hat{\mathbf{x}}_t + C\epsilon \rightarrow \mathbf{x}_{t-1}$$

Condition 1: noise coefficient $(A\sigma_t)^2 + C^2 = \sigma_{t-1}^2$

Condition 2: signal coefficient $A + B = 1$

There is 1 degree of freedom!
(amount of stochasticity in the process)

[ICLR 2021] Denoising Diffusion Implicit Models

Denoising Diffusion Implicit Models

DDIM: A first-order solver for the SDE

The ODE solver (C = 0) is quite efficient, often gives good results in 20 – 100 iterations!

A first-order exponential integrator in the ODE case.

DDIM (10, 20, 50, 100 iterations)

[ICLR 2021] Denoising Diffusion Implicit Models

Roadmap

I. Overview of DDIM

II. Denoising Diffusion Restoration Models

Solving noisy, linear inverse problems on images, quickly.

III. PhysDiff: Guided Human Motion Diffusion Model

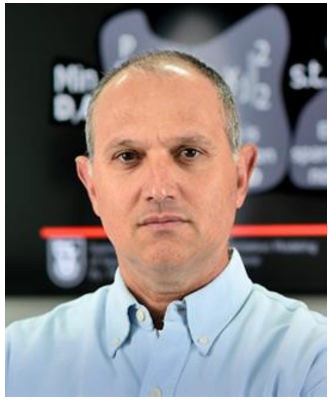
Enforce physical constraints in diffusion models.

IV. Pseudoinverse-Guided Diffusion Models

First to achieve SOTA performance comparable to domain-specific diffusion models.

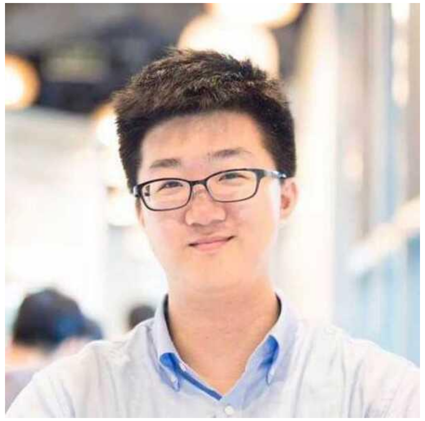
Denoising Diffusion Restoration Models

Bahjat Kawar



Michael Elad

Stefano Ermon



Jiaming Song

(Linear) Inverse Problems

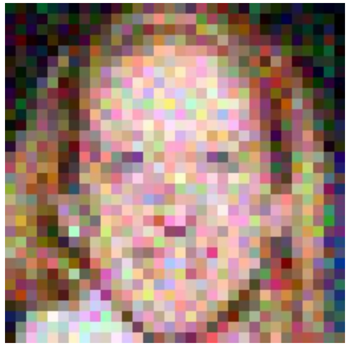
Given noisy observation y, recover x.

[Degradation]

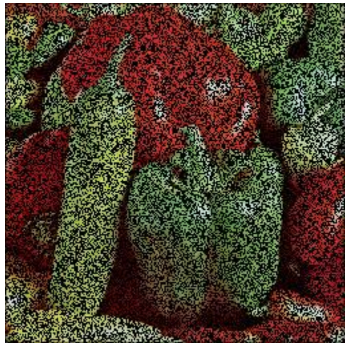
$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Noisy observation]

[Noise, Gaussian stddev = $\sigma_{\mathbf{y}}$]



Super-resolution: observed low resolution image.



Inpainting: observed masked image.

Deblurring: observed blurred image.

Denoising Diffusion Restoration Models

Super-resolution | Inpainting | Deblurring

Observations (Inputs) $\mathbf{y}$

Outputs from our method $\mathbf{x}_0$

[NeurIPS 2022] Denoising Diffusion Restoration Models

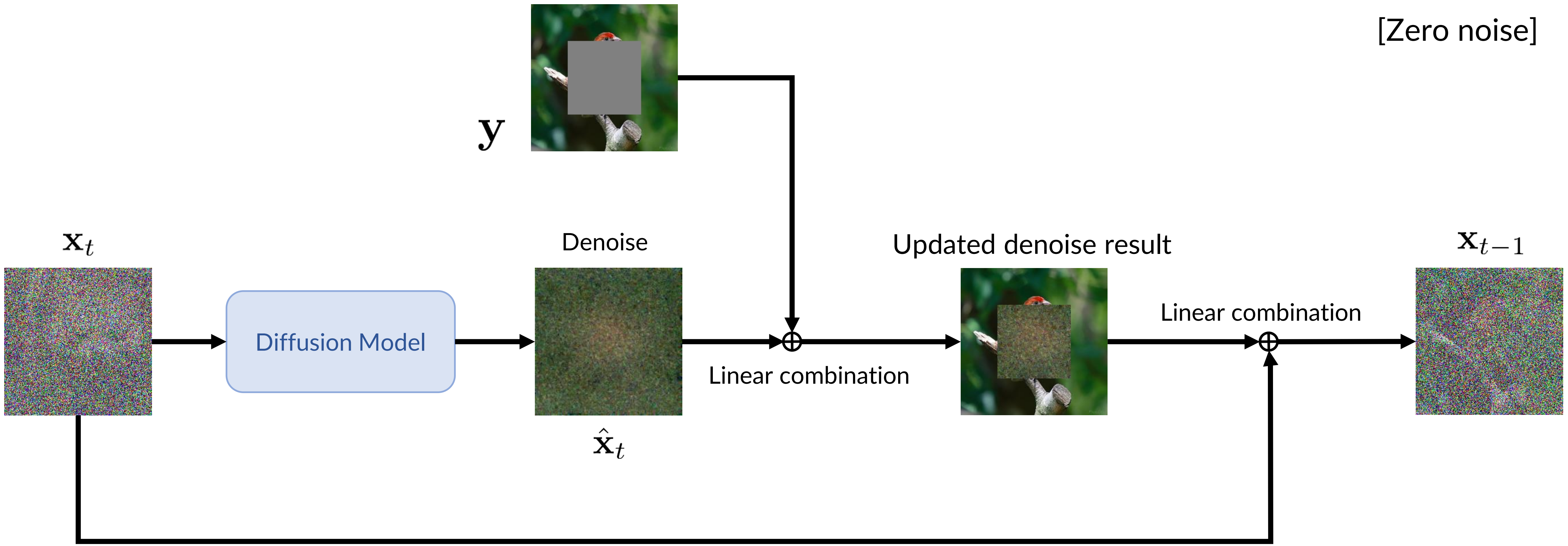
Denoising Diffusion Restoration Models

[H = Diagonal with 0 and 1's]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Zero noise]

Case 1: Noiseless inpainting



[NeurIPS 2022] Denoising Diffusion Restoration Models

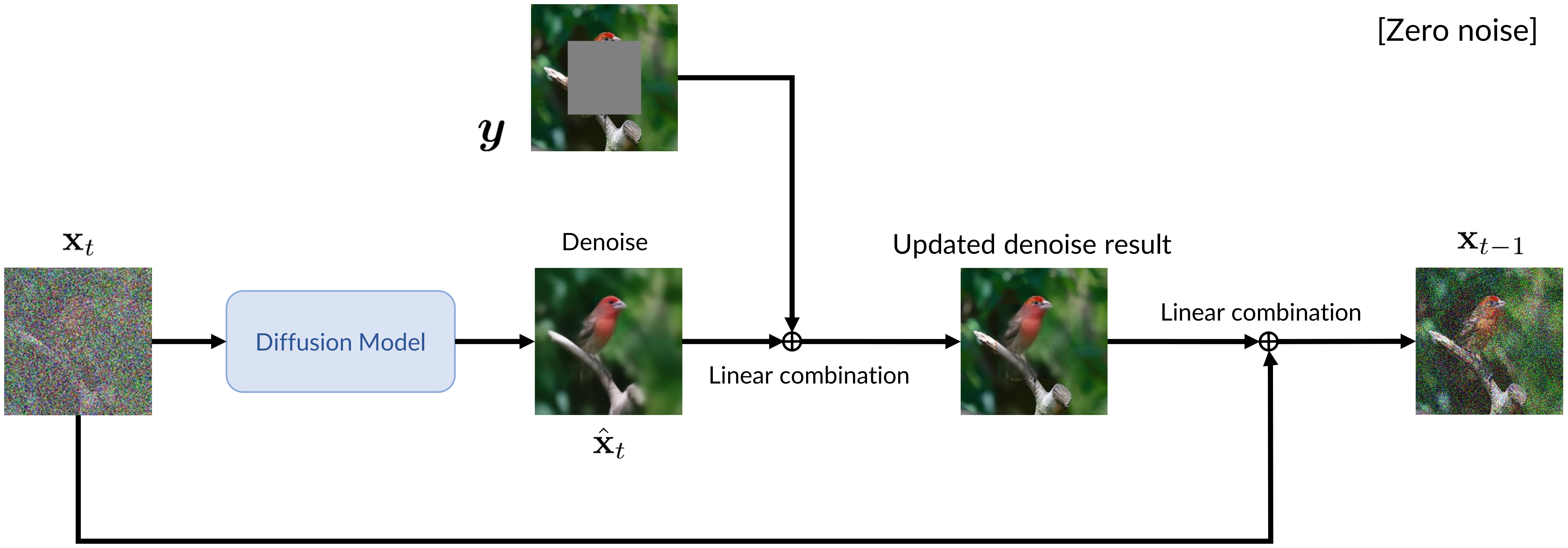
Denoising Diffusion Restoration Models

[H = Diagonal with 0 and 1's]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Zero noise]

Case 1: Noiseless inpainting



[ICLR 2021] Denoising Diffusion Implicit Models

[NeurIPS 2022] Denoising Diffusion Restoration Models

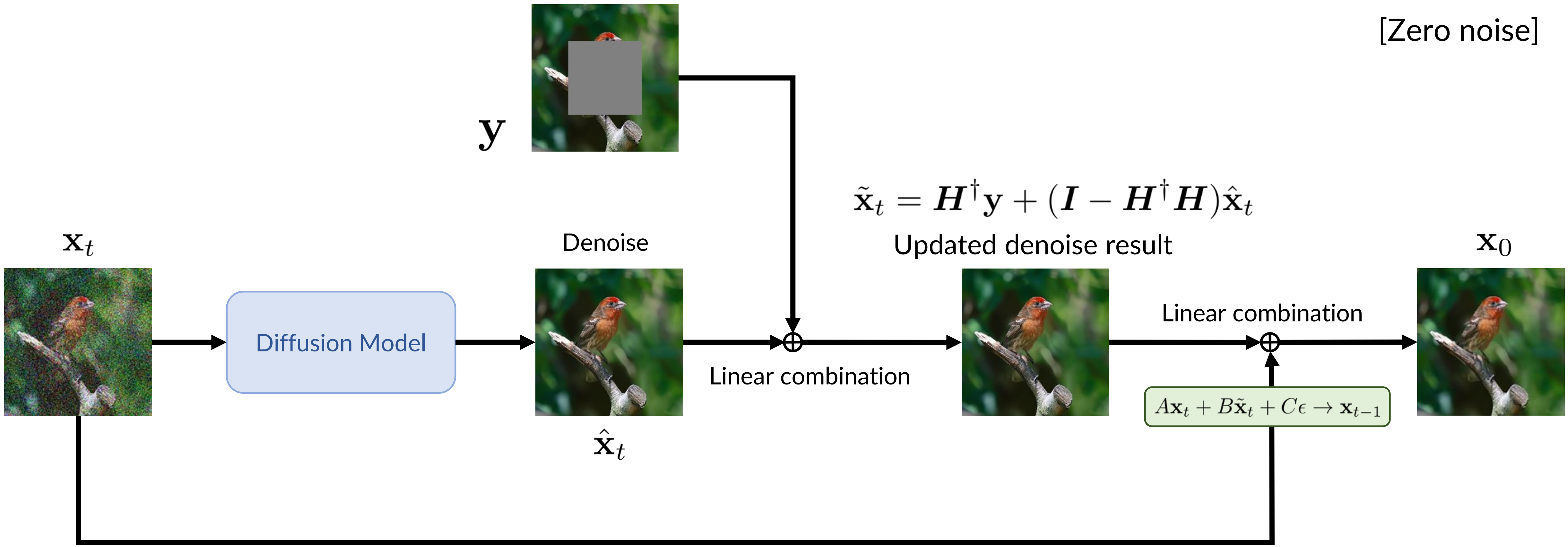
Denoising Diffusion Restoration Models

[H = Diagonal with 0 and 1's]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Zero noise]

Case 1: Noiseless inpainting



[ICLR 2021] Denoising Diffusion Implicit Models

[NeurIPS 2022] Denoising Diffusion Restoration Models

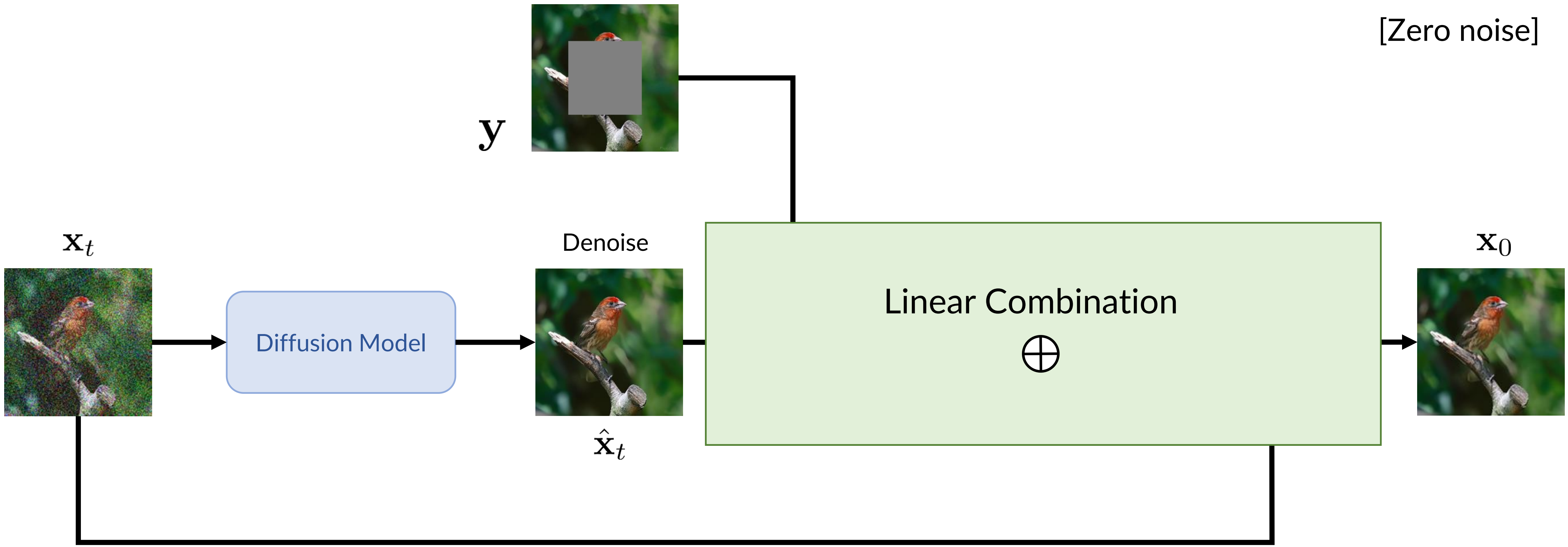
Denoising Diffusion Restoration Models

[H = Diagonal with 0 and 1's]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Zero noise]

Case 1: Noiseless inpainting



[ICLR 2021] Denoising Diffusion Implicit Models

[NeurIPS 2022] Denoising Diffusion Restoration Models

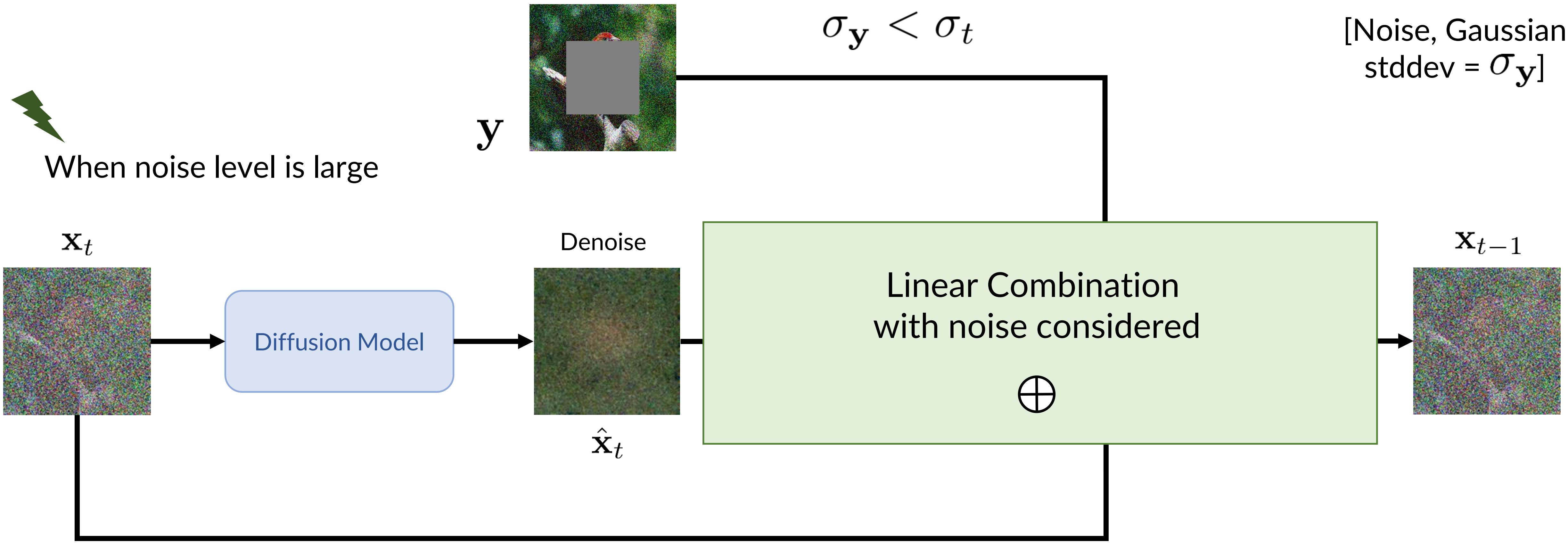
Denoising Diffusion Restoration Models

Case 2: **Noisy** inpainting

[H = Diagonal with 0 and 1's]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Noise, Gaussian stddev = $\sigma_{\mathbf{y}}$]



[NeurIPS 2022] Denoising Diffusion Restoration Models

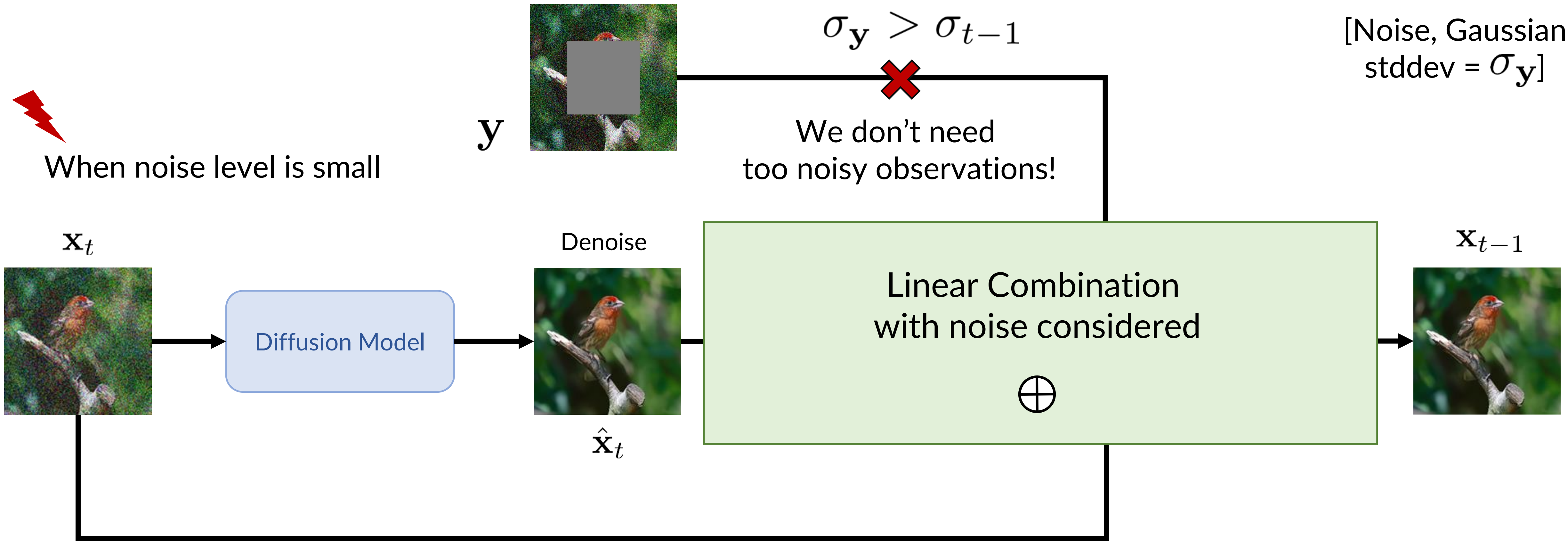
Denoising Diffusion Restoration Models

Case 2: **Noisy** inpainting

[H = Diagonal with 0 and 1's]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Noise, Gaussian stddev = $\sigma_{\mathbf{y}}$]



[NeurIPS 2022] Denoising Diffusion Restoration Models

Denoising Diffusion Restoration Models

[H = Diagonal with 0 and 1's]

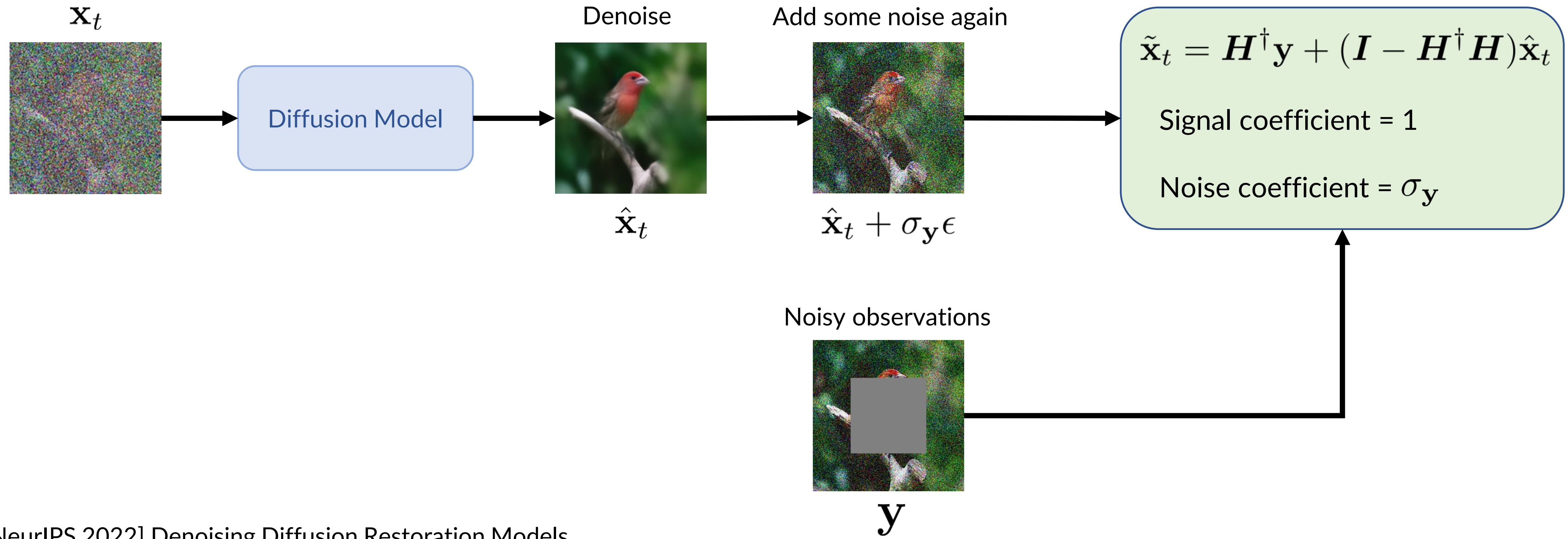
$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Noise, Gaussian stddev = $\sigma_{\mathbf{y}}$]

Case 2: **Noisy** inpainting

$\sigma_{\mathbf{y}} > \sigma_{t-1}$ Observation is already too noisy, just run DDIM.

$\sigma_{\mathbf{y}} \leq \sigma_{t-1}$ We can perform linear projection on "noisy denoised" samples.



[NeurIPS 2022] Denoising Diffusion Restoration Models

Denoising Diffusion Restoration Models

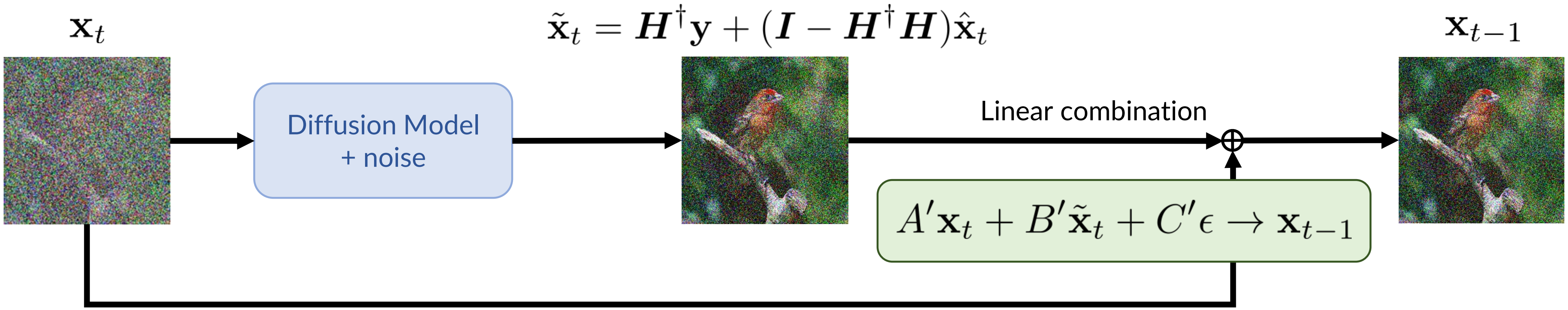
[H = Diagonal with 0 and 1's]

Case 2: **Noisy** inpainting

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

$\sigma_{\mathbf{y}} \leq \sigma_{t-1}$ We can perform linear projection on "noisy denoised" samples.

[Noise, Gaussian stddev = $\sigma_{\mathbf{y}}$]



Condition 1: noise coefficient $(A'\sigma_t)^2 + (B'\sigma_{\mathbf{y}})^2 + (C')^2 = \sigma_{t-1}^2$

Condition 2: signal coefficient $A' + B' = 1$

[NeurIPS 2022] Denoising Diffusion Restoration Models

Denoising Diffusion Restoration Models

[H = Diagonal with 0 and 1's]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Noise, Gaussian stddev = $\sigma_{\mathbf{y}}$]

Case 2: **Noisy** inpainting

$\sigma_{\mathbf{y}} > \sigma_{t-1}$ Observation is already too noisy, just run DDIM.

$$A\mathbf{x}_t + B\hat{\mathbf{x}}_t + C\epsilon \rightarrow \mathbf{x}_{t-1}$$

$\sigma_{\mathbf{y}} \leq \sigma_{t-1}$ We can perform linear projection on "noisy denoised" samples.

$$\tilde{\mathbf{x}}_t = \boldsymbol{H}^{\dagger}\mathbf{y} + (\boldsymbol{I} - \boldsymbol{H}^{\dagger}\boldsymbol{H})\hat{\mathbf{x}}_t$$

$$A'\mathbf{x}_t + B'\tilde{\mathbf{x}}_t + C'\epsilon \rightarrow \mathbf{x}_{t-1}$$

1 + 1 = 2 degrees of freedom! In the paper, these are η and η_b, respectively.

[NeurIPS 2022] Denoising Diffusion Restoration Models

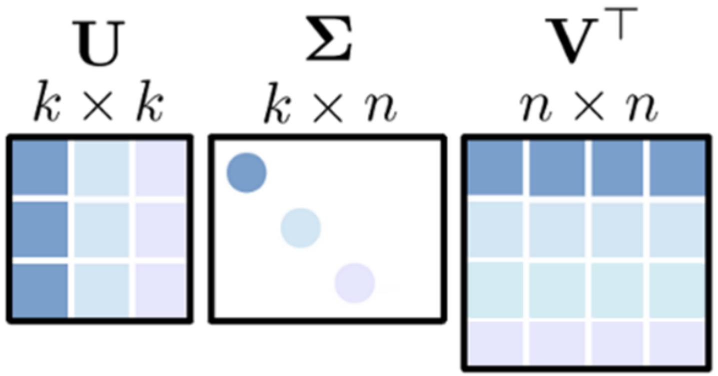
Denoising Diffusion Restoration Models

Most general case: **any linear inverse problem**

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

$$H = U\Sigma V^\top$$

H is “diagonal” with respect to its spectral space



$$\boldsymbol{U}^\top \mathbf{y} = \boldsymbol{\Sigma}(\boldsymbol{V}^\top \mathbf{x}_0) + \boldsymbol{U}^\top \mathbf{z}$$

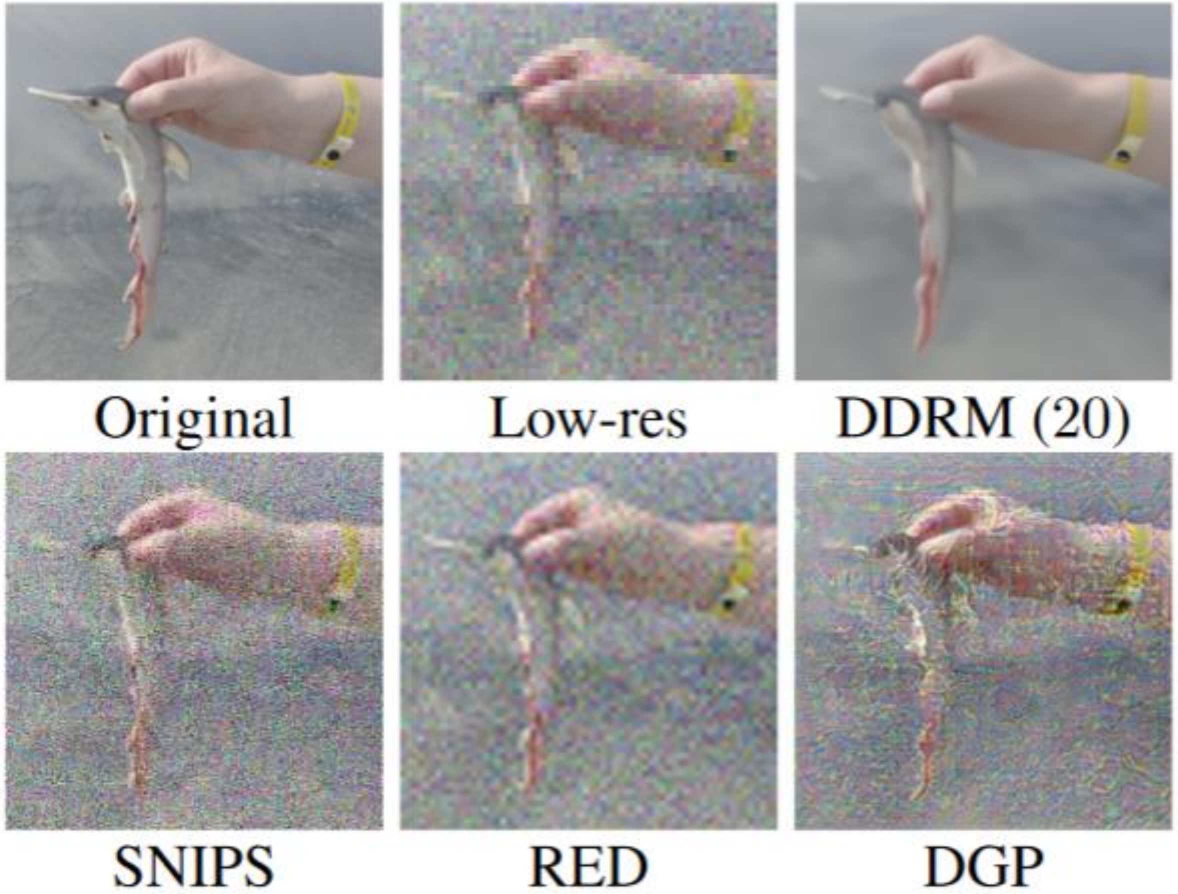
DDRM: run “denoising and inpainting”, but in spectral space

(handle noisy cases $\sigma_{\mathbf{y}} > \sigma_{t-1}$ for each dimension)

Results: compare against other DL-based methods

Input

Ours



[NeurIPS 2022] Denoising Diffusion Restoration Models

Results: compare against other DL-based methods

4× super-res (noiseless)	PSNR ↑	KID ↓	NFEs ↓
DGP	23.06	21.22	1500
RED	26.08	53.55	100
SNIPS	17.58	35.17	1000
DDRM	**26.55**	**7.22**	**20**

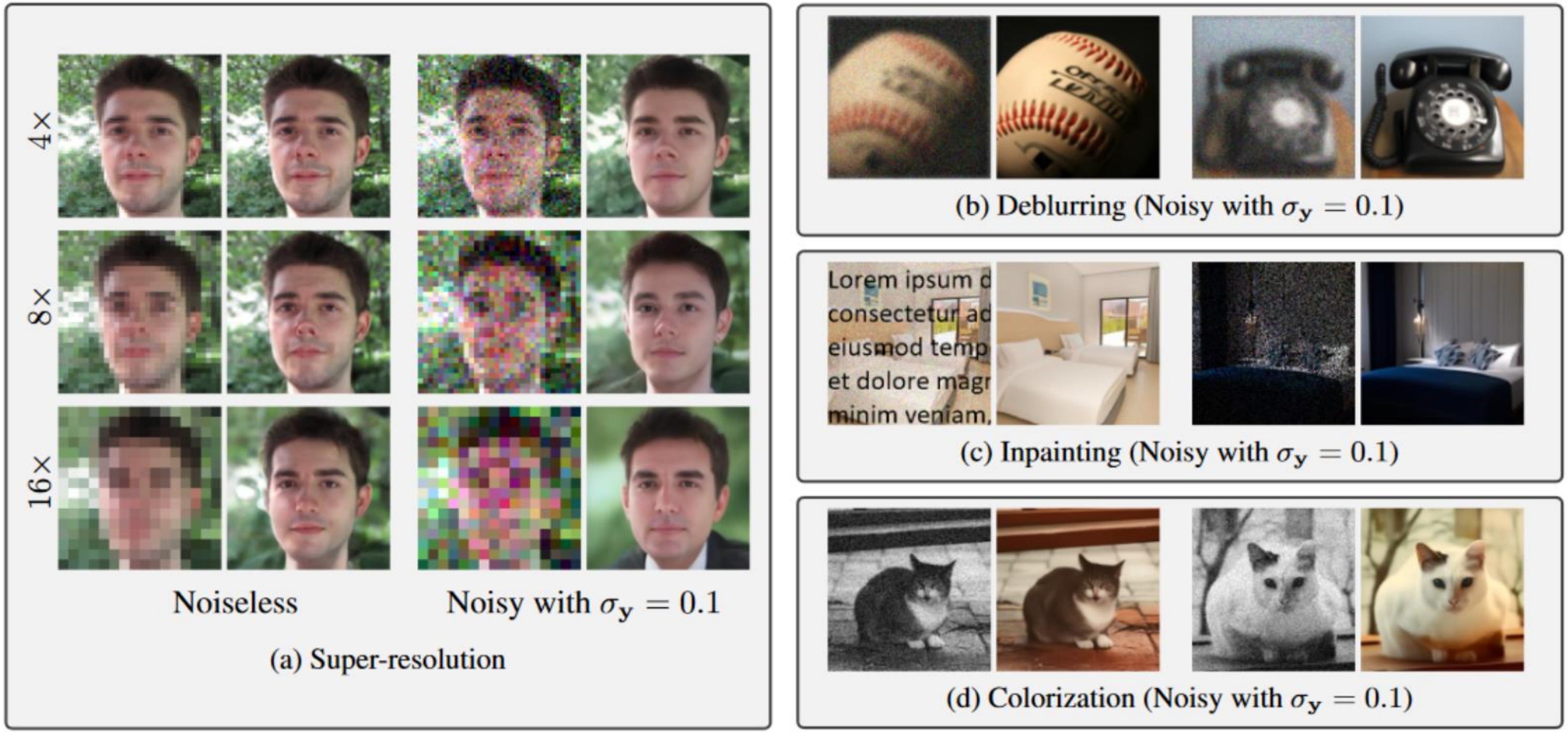
Ours → DDRM

Deblurring (noisy)	PSNR ↑	KID ↓	NFEs ↓
DGP	21.20	34.02	1500
RED	14.69	121.82	500
SNIPS	16.37	77.96	1000
DDRM	**25.45**	**15.24**	**20**

DDRM performs well within 20 Neural Function Evaluations (NFEs)!

[NeurIPS 2022] Denoising Diffusion Restoration Models

Qualitative Results



(a) Super-resolution

(b) Deblurring (Noisy with $\sigma_{\mathbf{y}} = 0.1$)

(c) Inpainting (Noisy with $\sigma_{\mathbf{y}} = 0.1$)

(d) Colorization (Noisy with $\sigma_{\mathbf{y}} = 0.1$)

[NeurIPS 2022] Denoising Diffusion Restoration Models

Applicable to other domains as well!

Astronomy

Strong-Lensing Source Reconstruction with Denoising Diffusion Restoration Models

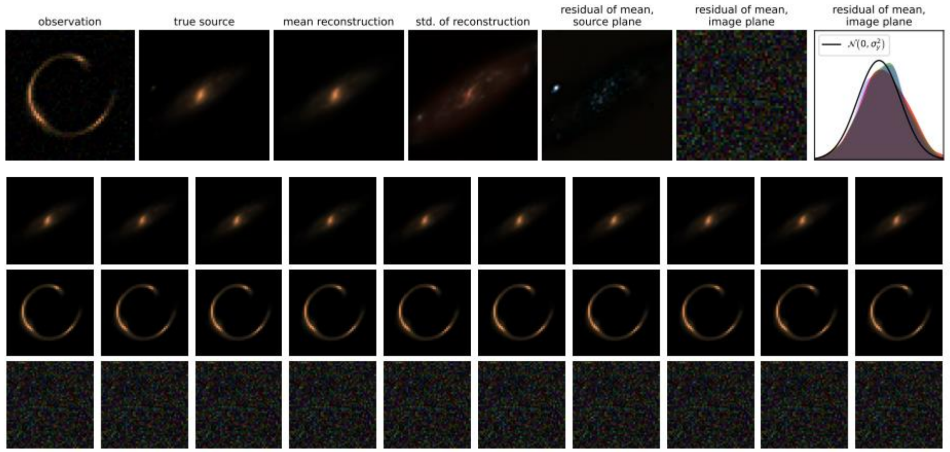


Figure 1: Top: from left to right, the mock observation, $\mathbf{y}$ (with a medium noise level), the true source, $\mathbf{x}$ (an unconstrained sample from AstroDDPM), the mean and standard deviation of 100 posterior samples from DDRM, $\mathbf{x}_{0,i} \sim p_\Theta(\mathbf{x}_0 \mid \mathbf{y})$, and the residual of the mean with respect to the true source and with respect to the observation in the image plane; finally, a histogram of the latter compared to a Gaussian. Bottom: each column is a random posterior sample (top row), which is then lensed to produce the respective noiseless image $\mathbf{H}\mathbf{x}_{0,i}$ (middle row). Shown (bottom row) are also the residuals between $\mathbf{H}\mathbf{x}_{0,i}$ and the observation. In residual plots, negative values in one channel are shown as positive values in the other two (red ↔ cyan, green ↔ magenta, blue ↔ yellow), considering complementary colors as "negative".

Speech

A VERSATILE DIFFUSION-BASED GENERATIVE REFINER FOR SPEECH ENHANCEMENT

Ryosuke Sawata *Naoki Murata* *Yuhta Takida* *Toshimitsu Uesaka*
Takashi Shibuya *Shusuke Takahashi* *Yuki Mitsufuji*

Sony Group Corporation, Tokyo, Japan

UNSUPERVISED VOCAL DEREVERBERATION WITH DIFFUSION-BASED GENERATIVE MODELS

Koichi Saito *Naoki Murata* *Toshimitsu Uesaka* *Chieh-Hsin Lai*
Yuhta Takida *Takao Fukui* *Yuki Mitsufuji*

Sony Group Corporation, Tokyo, Japan

Roadmap

I. Overview of DDIM

II. Denoising Diffusion Restoration Models

Solving noisy, linear inverse problems on images, quickly.

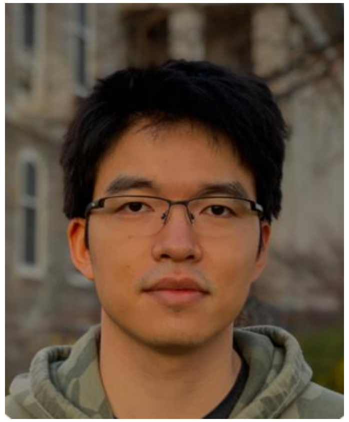
III. PhysDiff: Guided Human Motion Diffusion Model

Enforce physical constraints in diffusion models.

IV. Pseudoinverse-Guided Diffusion Models

First to achieve SOTA performance comparable to domain-specific diffusion models.

PhysDiff: Guided Human Motion Diffusion Model



Ye Yuan

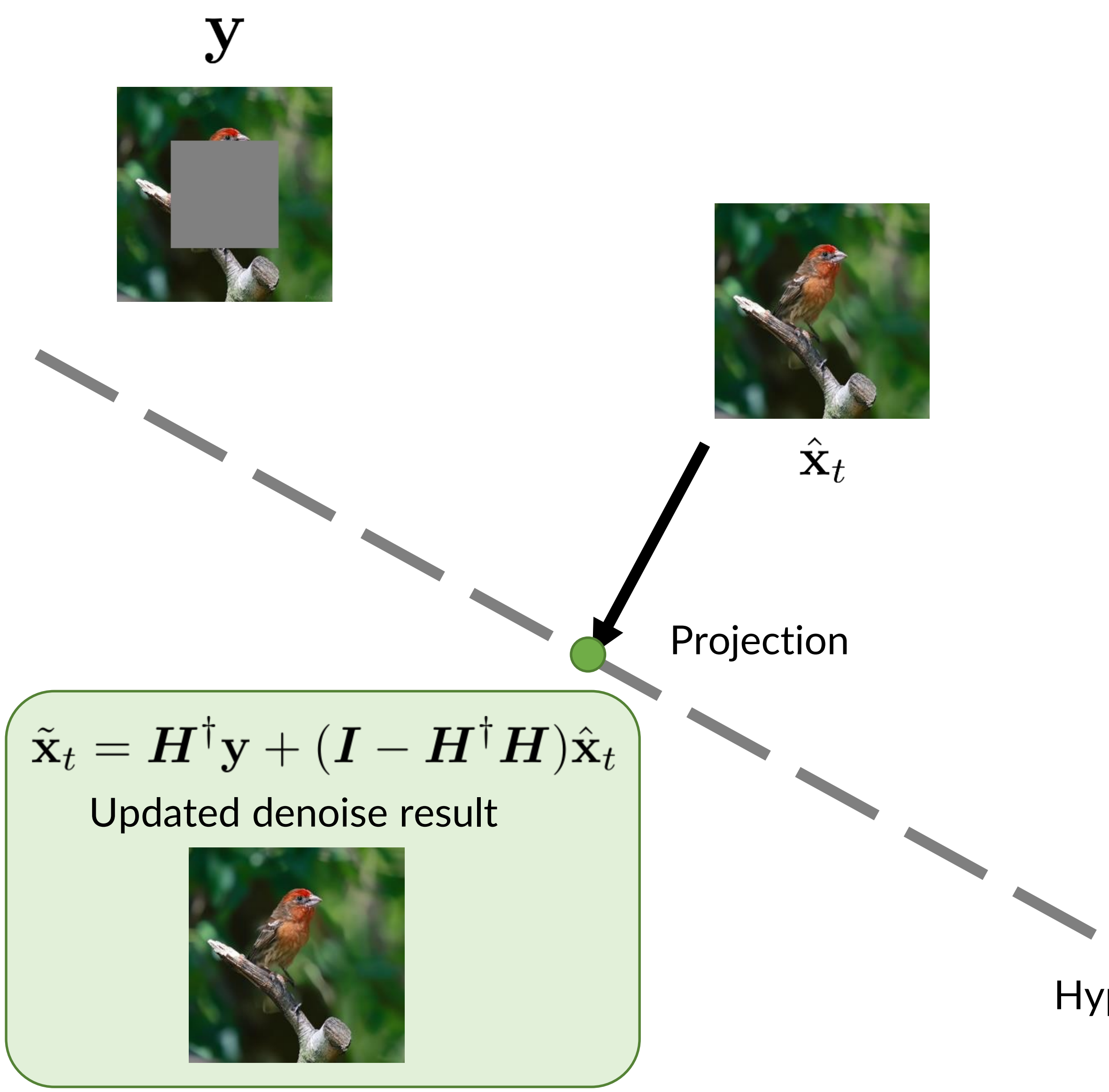
Jiaming Song

Umar Iqbal

Arash Vahdat

Jan Kautz

A projection step within DDRM…

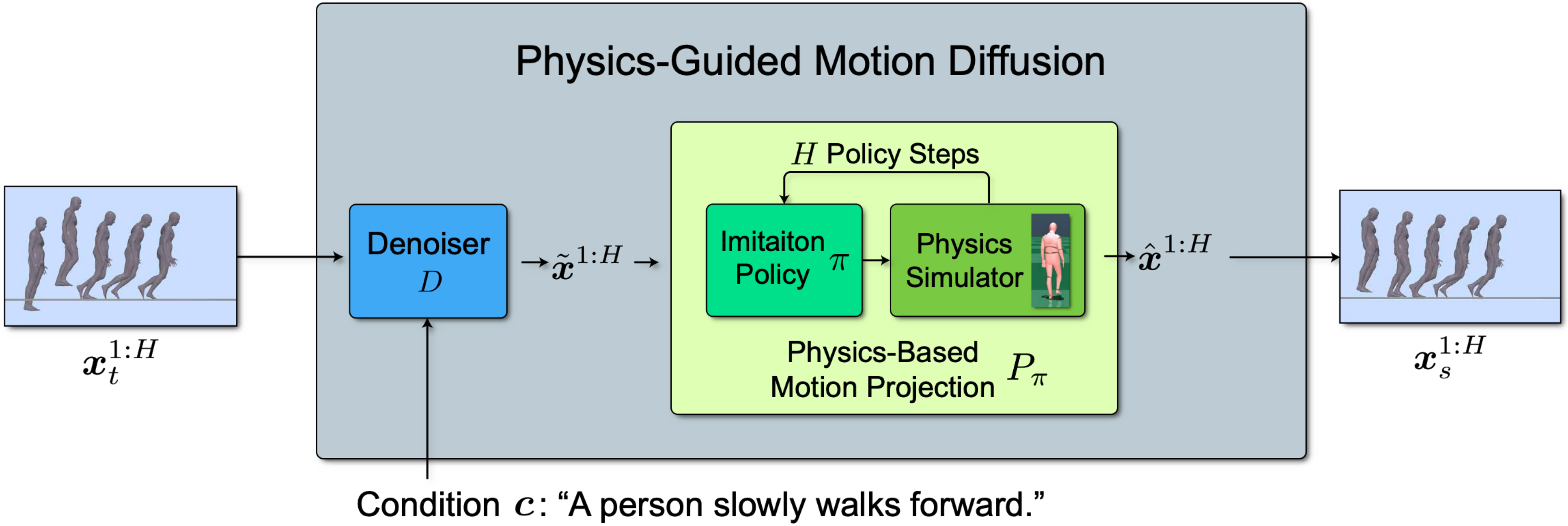


Hyperplane satisfying

$$\mathbf{y} = \boldsymbol{H}\mathbf{x}$$

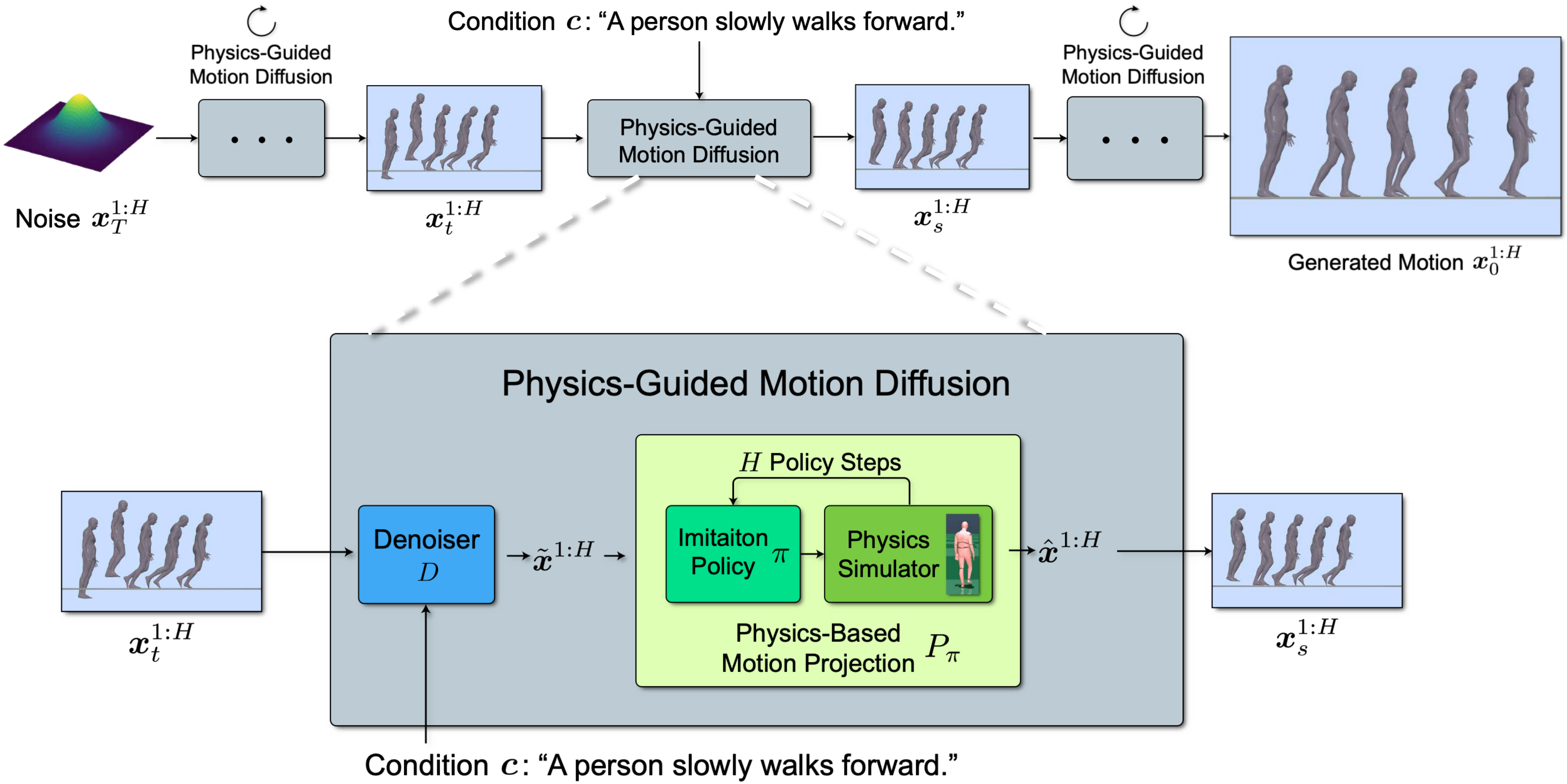
Can we extend the projection idea to other non-linear problems?

Idea behind PhysDiff



Project non-physically-plausible motions to physically-plausible ones!

Overview of PhysDiff



Roadmap

I. Overview of DDIM

II. Denoising Diffusion Restoration Models

Solving noisy, linear inverse problems on images, quickly.

III. PhysDiff: Guided Human Motion Diffusion Model

Enforce physical constraints in diffusion models.

IV. Pseudoinverse-Guided Diffusion Models

First to achieve SOTA performance comparable to domain-specific diffusion models.

Pseudoinverse-Guided Diffusion Models for Inverse Problems

Jiaming Song

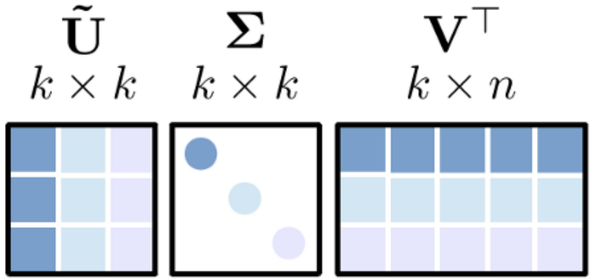
Arash Vahdat

Morteza Mardani

Jan Kautz

Limitations of DDRM

1. Only supports linear measurements.



Linear Combination
with noise considered

$\oplus$

2. Works poorly for very sparse measurements.

“Ground truth”

Input

Output

Problem: update only affects a few pixels!

[NeurIPS 2022] Denoising Diffusion Restoration Models

Challenges in plug-and-play style inverse problems

Solution: backprop through diffusion model, so update affects all pixels!

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

Conditional score | Prior diffusion model | This is not known!

Graphical model is a Markov chain:

Data

$$\mathbf{y} \leftarrow \mathbf{x}_0 \rightarrow \mathbf{x}_t$$

Observation | Add Gaussian noise

$$p_t(\mathbf{y}|\mathbf{x}_t) = \int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t)p(\mathbf{y}|\mathbf{x}_0)d\mathbf{x}_0$$ is “intractable” even if we have $p(\mathbf{y}|\mathbf{x}_0)$

Guidance methods for inverse problems

Solution: backprop through diffusion model, so update affects all pixels!

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

"Score" — Prior diffusion model — This is not known!

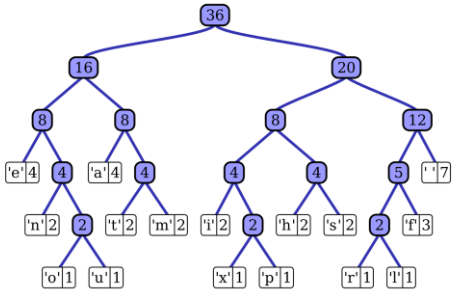
We approximate this with *Pseudoinverse Guidance*

[NeurIPS 2021] Diffusion Models Beat GANs on Image Synthesis

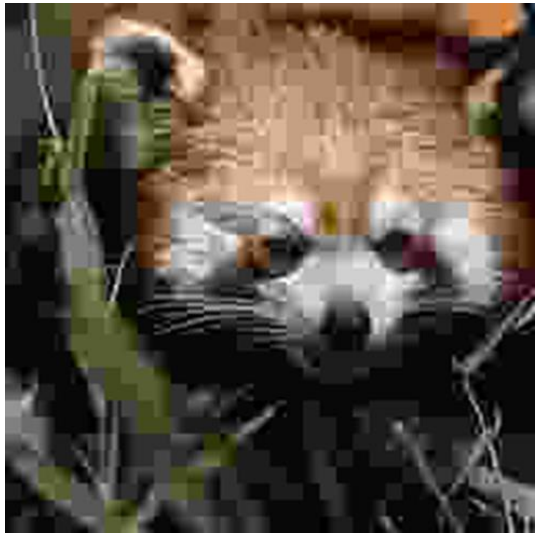
Pseudoinverse Guidance

Given input (e.g., JPEG encoding), how to recover with diffusion models?

JPEG is not differentiable!



JPEG Huffman coding

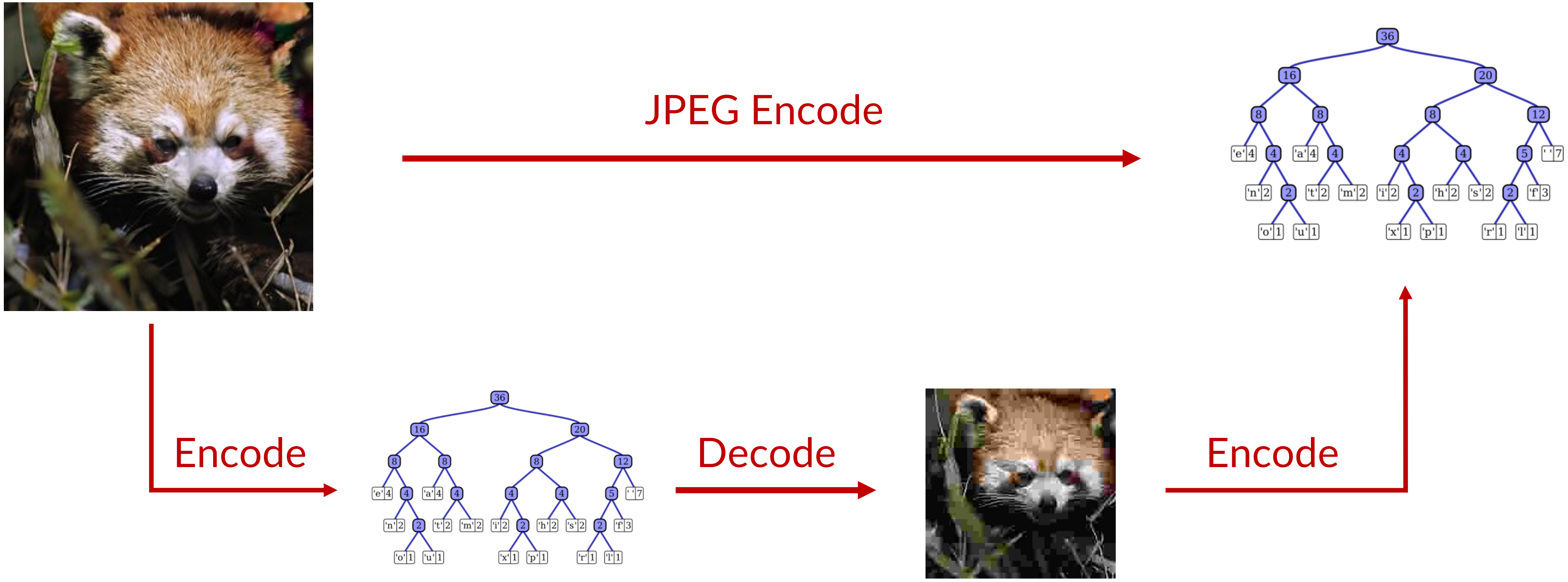


JPEG Decode

ΠGDM Output

Pseudoinverse Guidance

We use a property of pseudoinverse of matrices: $$\boldsymbol{H}\boldsymbol{H}^{\dagger}\boldsymbol{H} = \boldsymbol{H}$$

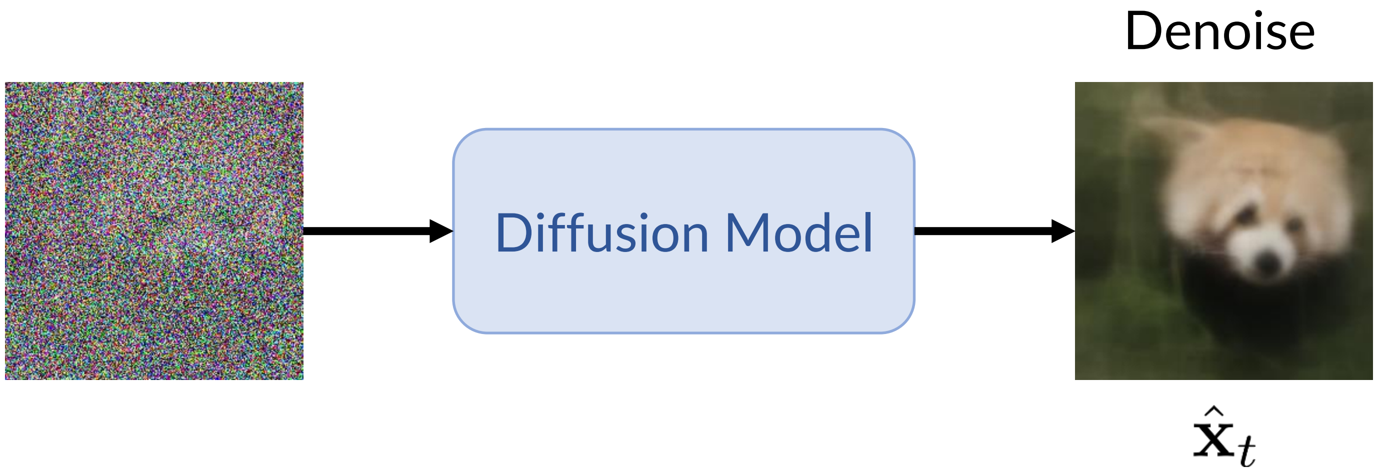


JPEG decode is "pseudoinverse" of JPEG encode!

[NeurIPS 2022 Workshop] JPEG Artifact Correction using Denoising Diffusion Restoration Models

Pseudoinverse Guidance: Step by step

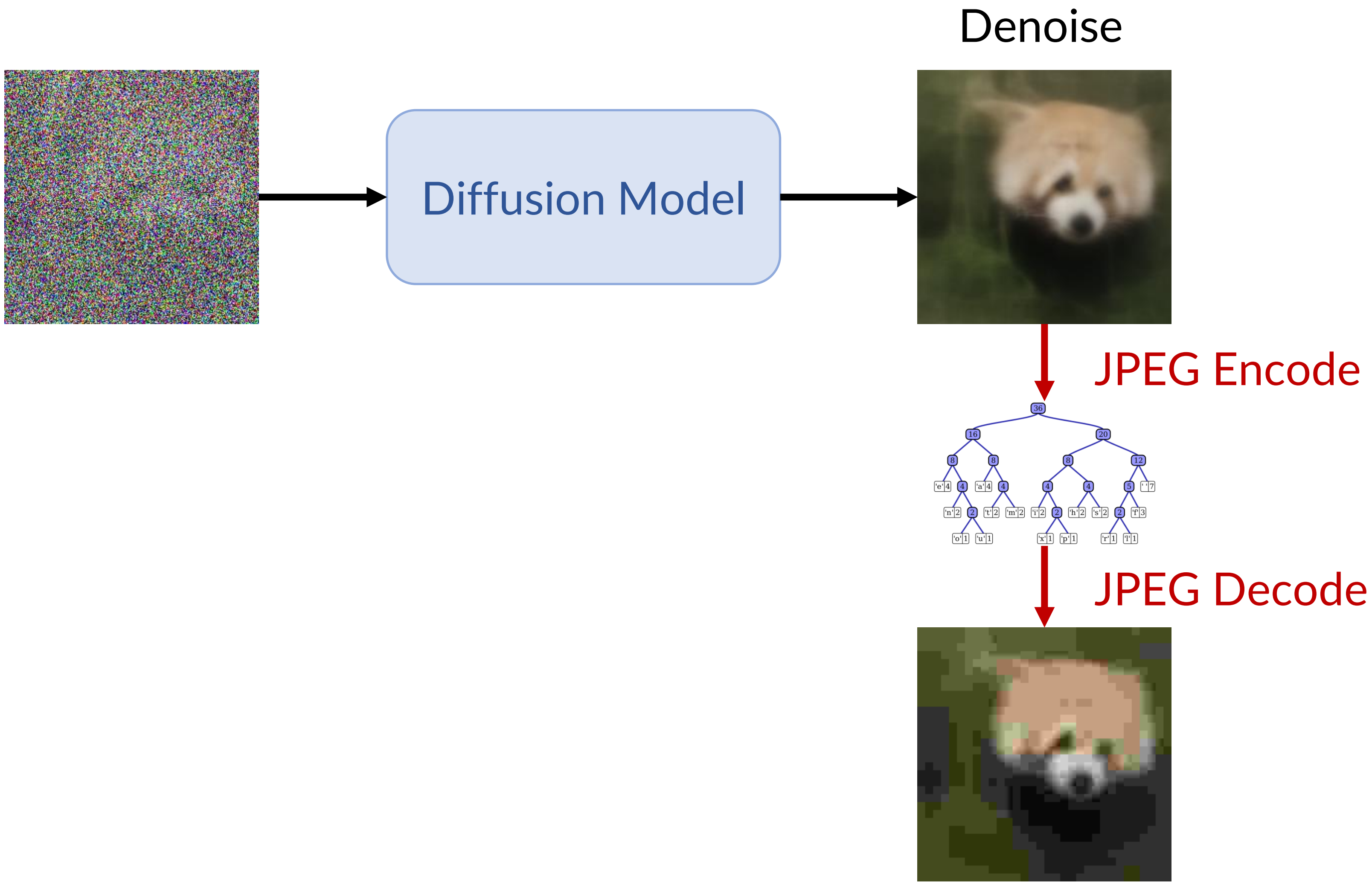
Step 1: Diffusion model makes a prediction that is "denoised".



The diffusion model is generic and not problem-dependent!

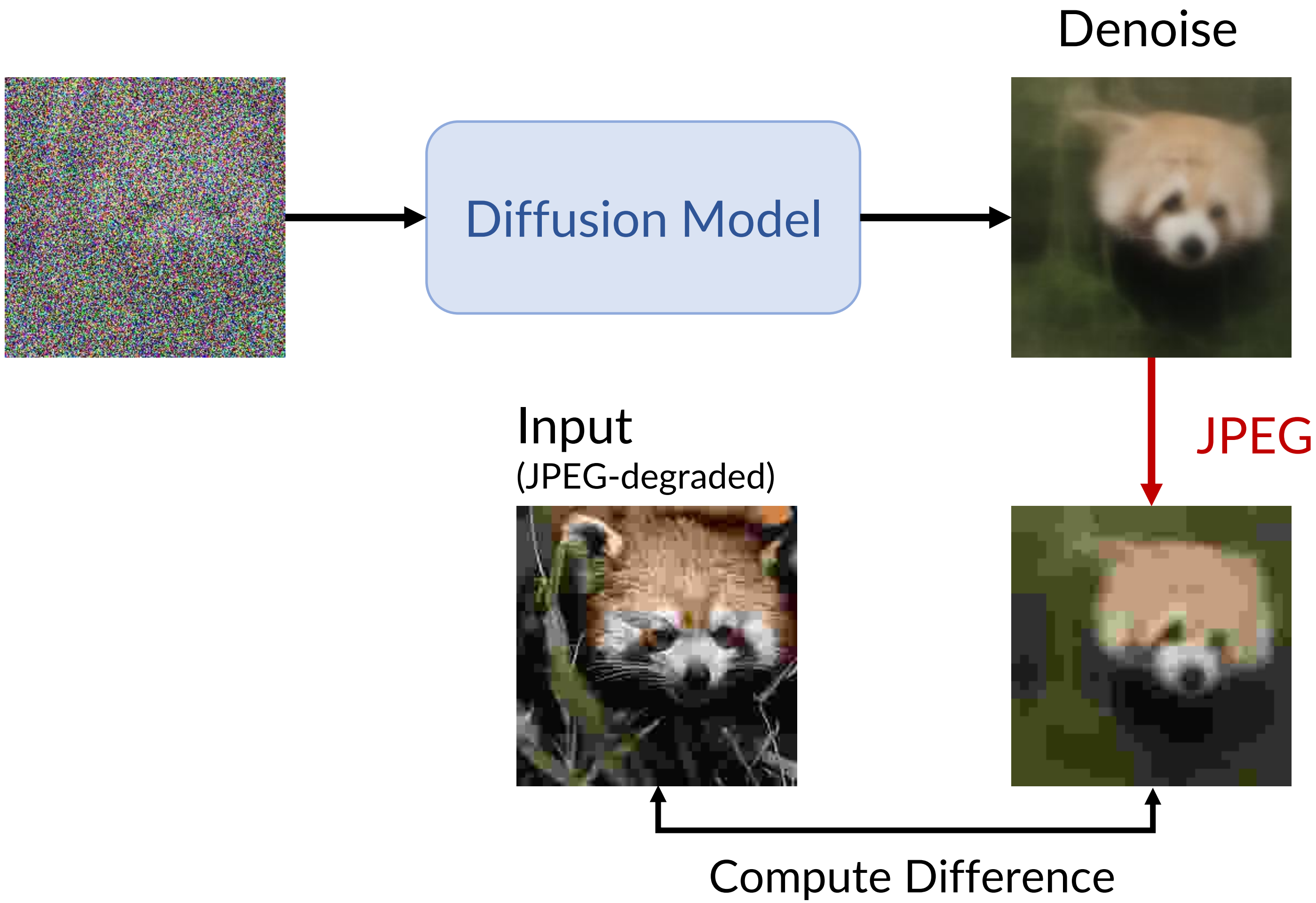
Pseudoinverse Guidance: Step by step

Step 2: Degradation & its “pseudoinverse” are applied to denoised prediction



Pseudoinverse Guidance: Step by step

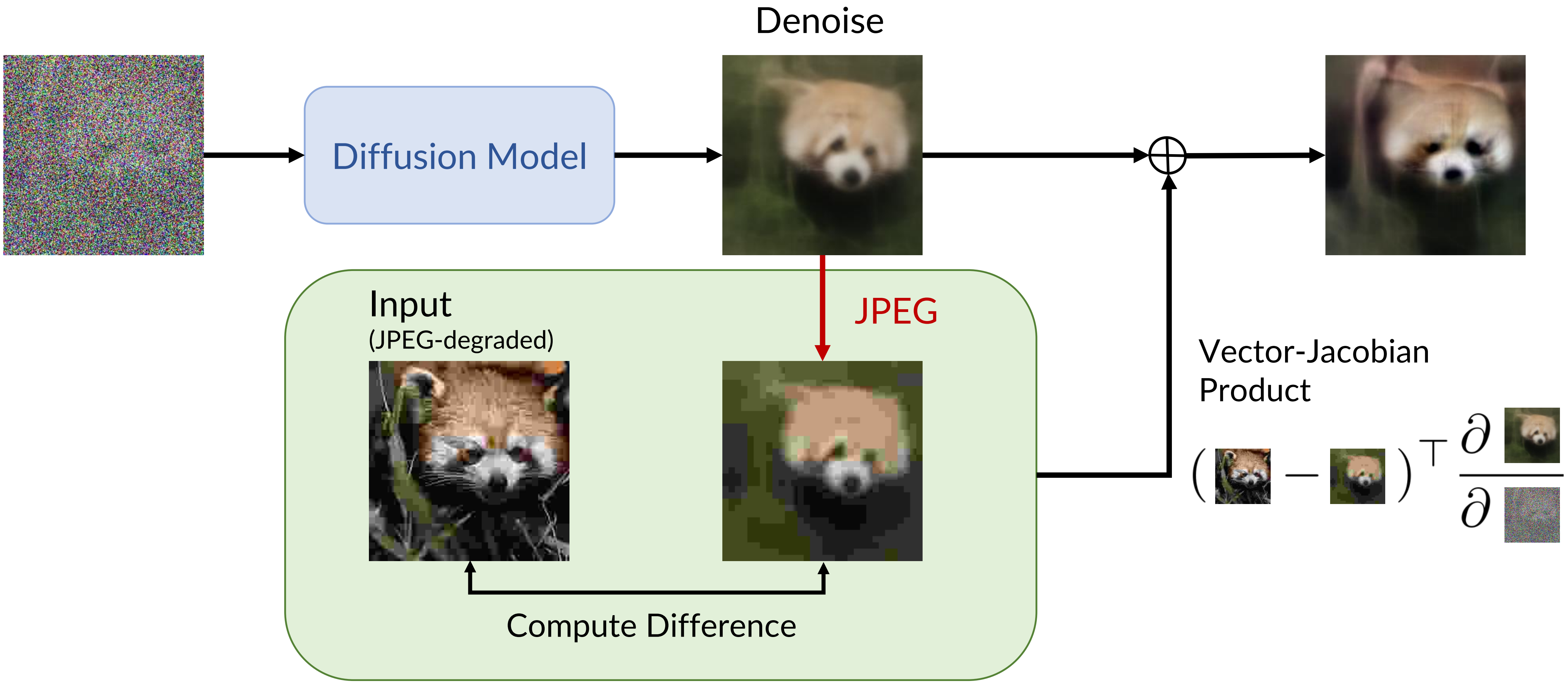
Step 3: Compute difference between the given input.



$$\boldsymbol{H}^{\dagger}\mathbf{y} - \boldsymbol{H}^{\dagger}\boldsymbol{H}\hat{\mathbf{x}}_t$$

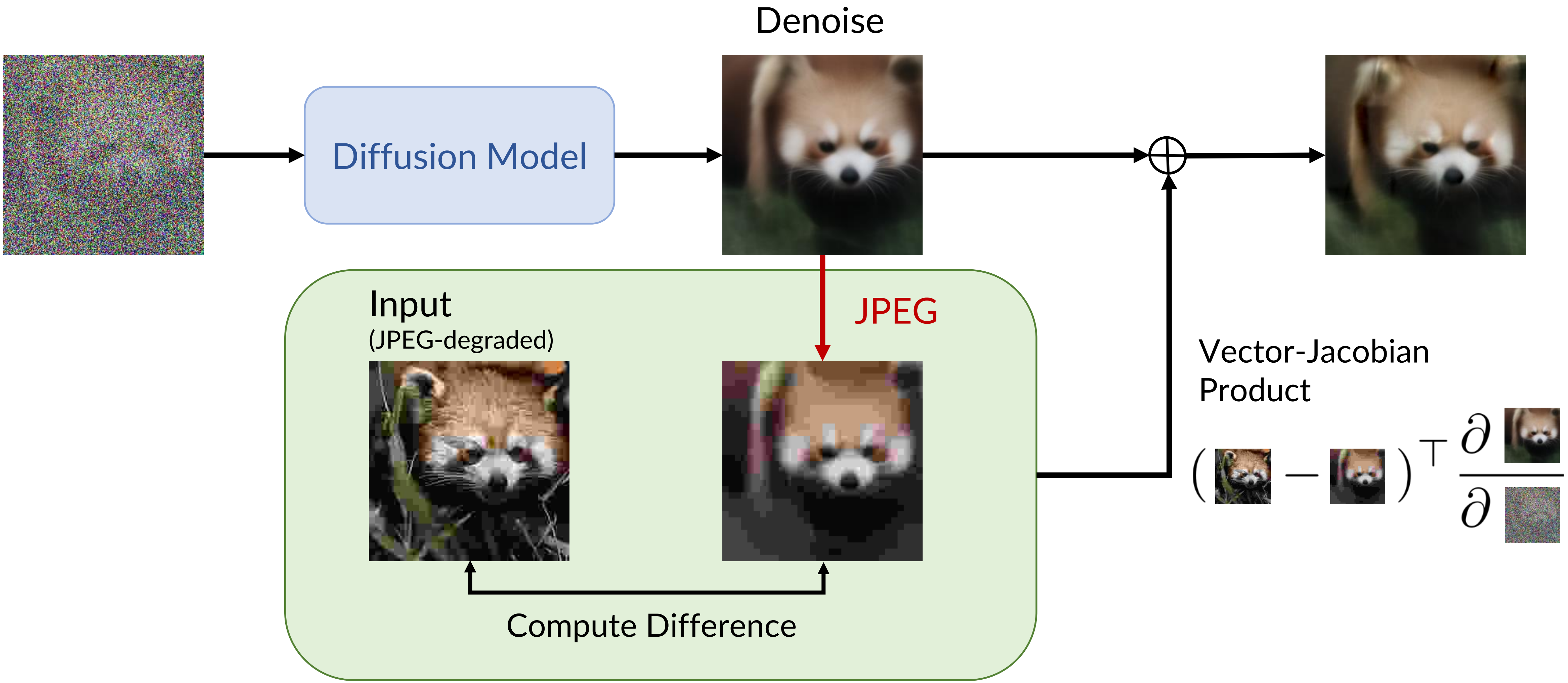
Pseudoinverse Guidance: Step by step

Step 4: Leverage the difference to guide the prediction closer to input.



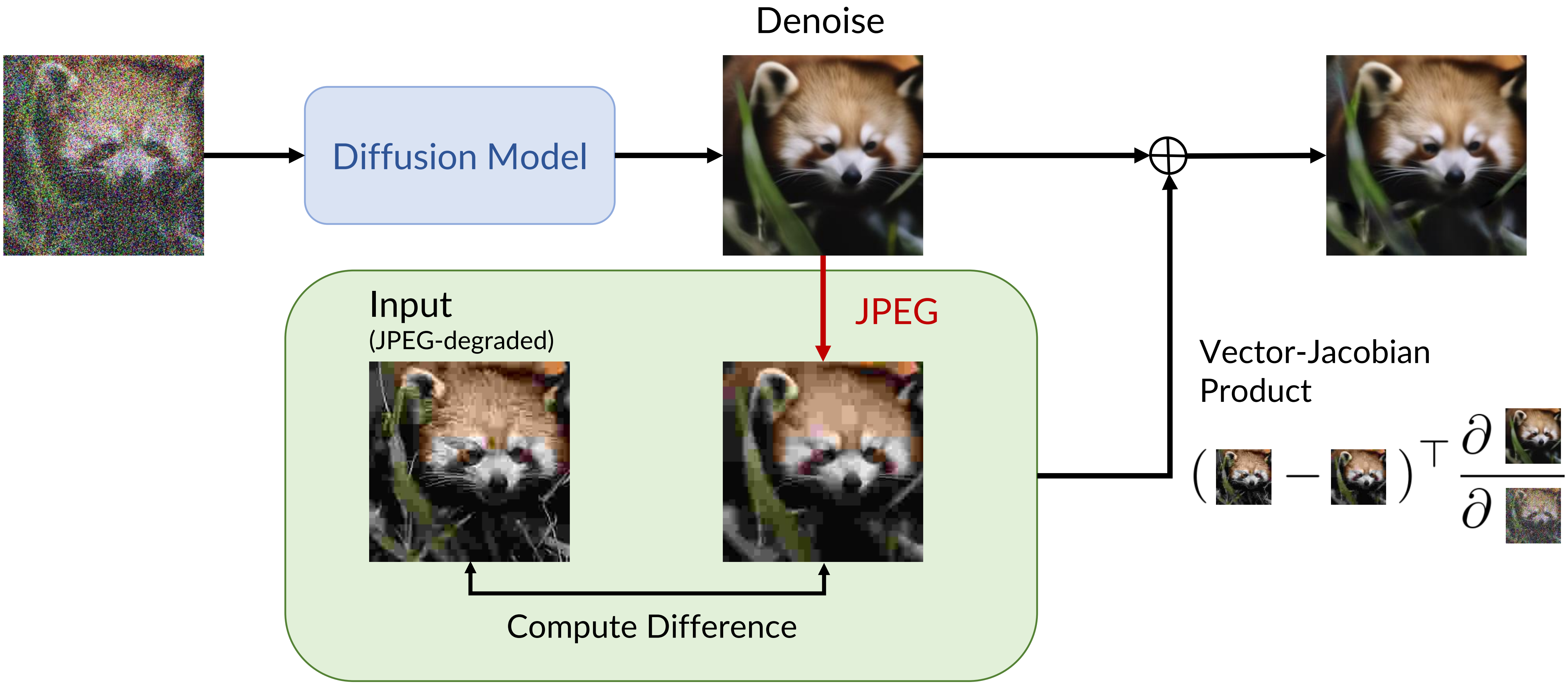
Pseudoinverse Guidance: Step by step

Step 5: Repeat for lower noise levels (high noise).



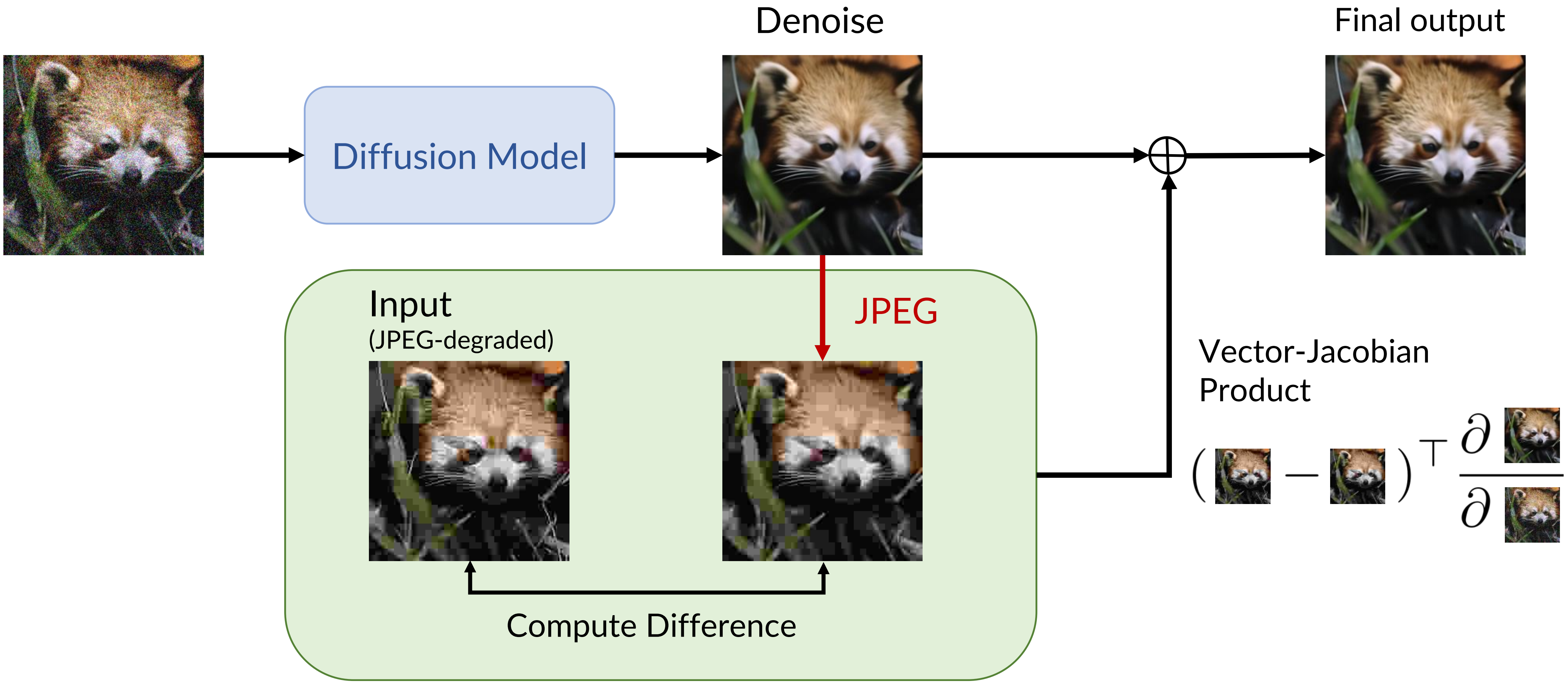
Pseudoinverse Guidance: Step by step

Step 5: Repeat for lower noise levels (mid noise).



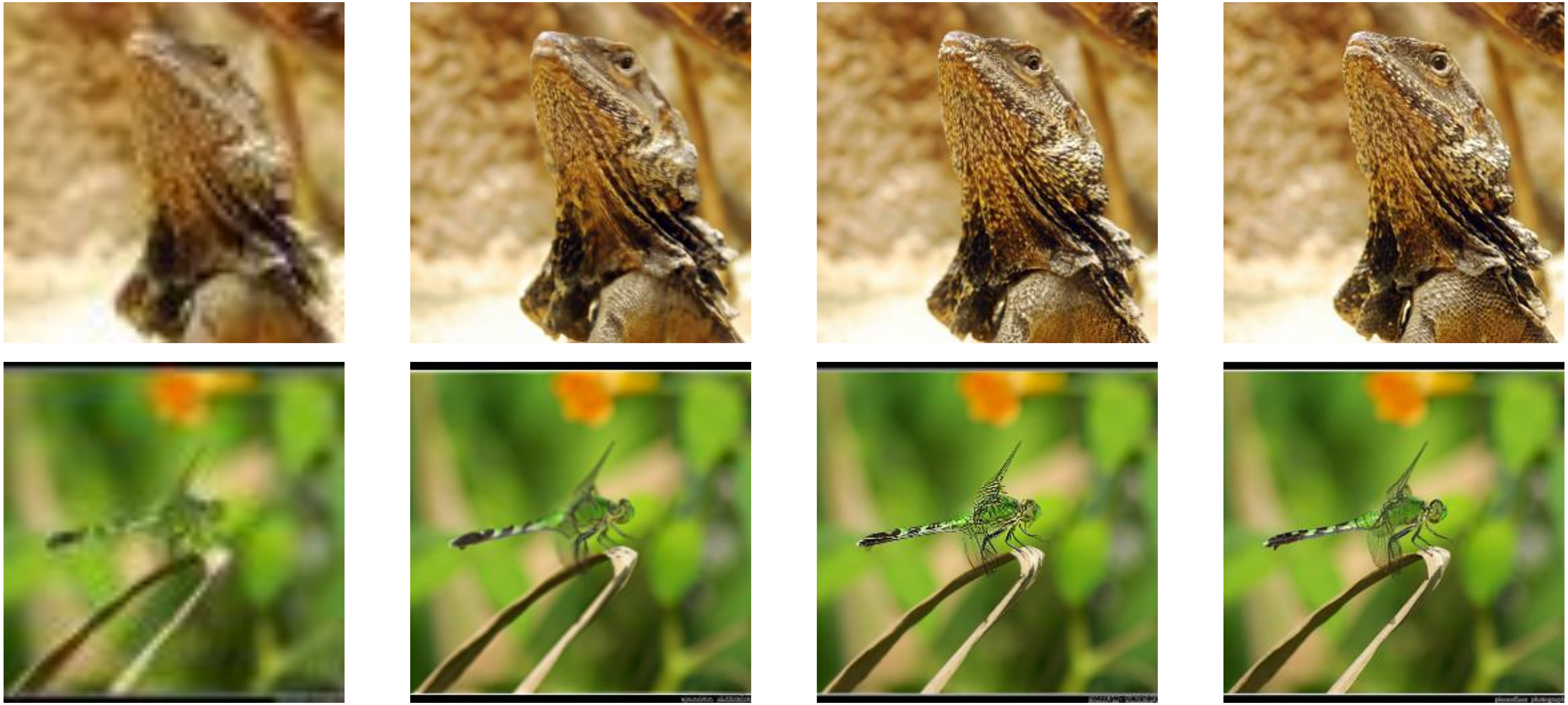
Pseudoinverse Guidance: Step by step

Step 5: Repeat for lower noise levels (low noise).



ΠGDM in practice (super-resolution)

Using a generic diffusion model, ΠGDM is competitive against specialized models!



Low-res Input | ADM-U Output | ΠGDM Output | Reference

ΠGDM in practice (super-resolution)

Using a generic diffusion model, ΠGDM is competitive against specialized models!

Low-res Input | ADM-U Output | ΠGDM Output | Reference

ΠGDM in practice (JPEG restoration)

Using a generic diffusion model, ΠGDM is competitive against specialized models!

JPEG Input | Palette Output | ΠGDM Output | Reference

ΠGDM in practice (Inpainting)

Using a generic diffusion model, ΠGDM is competitive against specialized models!

Masked Input | ΠGDM Output 1 | ΠGDM Output 2 | ΠGDM Output 3

Pseudoinverse Guidance

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

"Score" — Prior diffusion model — This is not known!

Idea: find good approximations to $p_t(\mathbf{y}|\mathbf{x}_t)$

$$p_t(\mathbf{y}|\mathbf{x}_t) = \int_{\mathbf{x}_0} p(\mathbf{x}_0|\mathbf{x}_t) p(\mathbf{y}|\mathbf{x}_0) d\mathbf{x}_0$$

Approximate with Gaussian

$$p_t(\mathbf{x}_0|\mathbf{x}_t) \approx \mathcal{N}(\hat{\mathbf{x}}_t, r_t^2 \boldsymbol{I})$$

$$\hat{\mathbf{x}}_t = D(\mathbf{x}_t; \sigma_t)$$

Mean = denoised result
Standard deviation = hyperparameter

Known from linear relationship

[Degradation]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Noisy observation] [Noise]

Pseudoinverse Guidance

$$p_t(\mathbf{y}|\mathbf{x}_t) \approx \mathcal{N}(\boldsymbol{H}\hat{\mathbf{x}}_t, r_t^2 \boldsymbol{H}\boldsymbol{H}^\top + \sigma_\mathbf{y}^2 \boldsymbol{I})$$ is approximately Gaussian!

Case 1: Noise is positive

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx \Big(\underbrace{(\mathbf{y} - \boldsymbol{H}\hat{\mathbf{x}}_t)^\top \left(r_t^2 \boldsymbol{H}\boldsymbol{H}^\top + \sigma_\mathbf{y}^2 \boldsymbol{I}\right)^{-1} \boldsymbol{H}}_{\text{vector}} \underbrace{\frac{\partial \hat{\mathbf{x}}_t}{\partial \mathbf{x}_t}}_{\text{Jacobian}} \Big)^\top.$$

Backprop through diffusion model

Case 2: Noise is zero

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx r_t^{-2} \left((\boldsymbol{H}^\dagger \mathbf{y} - \boldsymbol{H}^\dagger \boldsymbol{H}\hat{\mathbf{x}}_t)^\top \frac{\partial \hat{\mathbf{x}}_t}{\partial \mathbf{x}_t} \right)^\top$$

$\boldsymbol{H}^\dagger = \boldsymbol{H}^\top (\boldsymbol{H}\boldsymbol{H}^\top)^{-1}$ is matrix pseudoinverse!

- Vector Jacobian Product (vJp) can be computed by backprop
- Vector does not have to be differentiable

Pseudoinverse Guidance

Case 2: Noise is zero

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx r_t^{-2}\left((\boldsymbol{H}^\dagger \mathbf{y} - \boldsymbol{H}^\dagger \boldsymbol{H} \hat{\mathbf{x}}_t)^\top \frac{\partial \hat{\mathbf{x}}_t}{\partial \mathbf{x}_t}\right)^\top$$

$\boldsymbol{H}^\dagger = \boldsymbol{H}^\top(\boldsymbol{H}\boldsymbol{H}^\top)^{-1}$ is matrix pseudoinverse!

Pseudoinverse guidance for case 2:

1. Compute vector $\boldsymbol{H}^\dagger \mathbf{y} - \boldsymbol{H}^\dagger \boldsymbol{H} \hat{\mathbf{x}}_t$
2. Compute vector-Jacobian product with backprop.

Pseudoinverse Guidance vs. Reconstruction Guidance

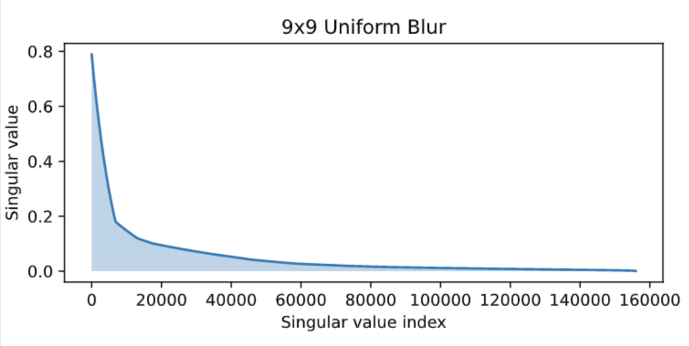
Reconstruction guidance [Ho et al., 2022 (Video Diffusion Models)]:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx r_t^{-2}\|\mathbf{y} - \boldsymbol{H}\hat{\mathbf{x}}_t\|_2^2 = r_t^{-2}\left((\boldsymbol{H}^\top \mathbf{y} - \boldsymbol{H}^\top \boldsymbol{H}\hat{\mathbf{x}}_t)^\top \frac{\partial \hat{\mathbf{x}}_t}{\partial \mathbf{x}_t}\right)^\top$$

Pseudoinverse guidance changes transpose to pseudoinverse!

Singular values of $\boldsymbol{H}^\dagger \boldsymbol{H}$ are 0 or 1

Works well for poorly-conditioned matrices!



Pseudoinverse Guidance: Quantitative Results

Super-resolution

Filter	Method	FID ↓	CA ↑
Pool	ADM (*cc*, Dhariwal & Nichol (2021))	**3.1**	**73.4%**
	DDRM (Kawar et al., 2022a)	14.8	64.6%
	ΠGDM (*Ours*)	3.8	72.3%
	DDRM (*cc*, Kawar et al. (2022a))	14.1	65.2%
	ΠGDM (*cc*, *Ours*)	3.6	72.2%
Bicubic	SR3 (Saharia et al., 2021)	5.2	68.3%
	ADM (*cc*, Dhariwal & Nichol (2021))	14.8	66.7%
	DDRM (Kawar et al., 2022a)	21.3	63.2%
	ΠGDM (*Ours*)	3.6	72.1%
	DDRM (*cc*, Kawar et al. (2022a))	19.6	65.3%
	ΠGDM (*cc*, Ours)	**3.2**	**75.1%**

Inpainting

Mask	Method	FID-10k ↓	CA ↑
Center	DeepFillv2 (Yu et al., 2019)	18.0	64.3%
	Palette (Saharia et al., 2022a)	**6.6**	69.3%
	DDRM (Kawar et al., 2022a)	24.4	62.1%
	ΠGDM (*Ours*)	7.3	**72.6%**
	ΠGDM (noisy, *Ours*)	9.5	72.2%
Freeform	DeepFillv2 (Yu et al., 2019)	9.4	68.8%
	Palette (Saharia et al., 2022a)	**5.2**	72.3%
	DDRM (Kawar et al., 2022a)	8.6	71.9%
	ΠGDM (*Ours*)	5.3	**75.3%**
	ΠGDM (noisy, *Ours*)	7.3	74.5%

Comparable with Palette and ADM-U,
state-of-the-art diffusion models specifically trained on the tasks.

Pseudoinverse Guidance: Quantitative Results

JPEG Restoration

QF	Method	FID-10k ↓	CA ↑
5	Regression (Saharia et al., 2022a)	29.0	52.8%
	Palette (Saharia et al., 2022a)	**8.3**	**64.2%**
	ΠGDM (*Ours*)	8.6	64.1%
10	Regression (Saharia et al., 2022a)	18.0	63.5%
	Palette (Saharia et al., 2022a)	**5.4**	70.7%
	ΠGDM (Ours)	6.0	**71.0%**
20	Regression (Saharia et al., 2022a)	11.5	69.7%
	Palette (Saharia et al., 2022a)	**4.3**	73.5%
	ΠGDM (Ours)	4.7	**74.4%**

Comparable with Palette and ADM-U,
state-of-the-art diffusion models specifically trained on the tasks.

Combining multiple operators

$$h(\mathbf{x}) = h_1 \circ h_2 \ldots \circ h_k(\mathbf{x})$$

down sampling -> JPEG encode -> masking

$$h^\dagger(\mathbf{x}) \approx h_k^\dagger \circ \ldots \circ h_2^\dagger \circ h_1^\dagger(\mathbf{x})$$

unmasking -> up sampling -> JPEG decode

Input

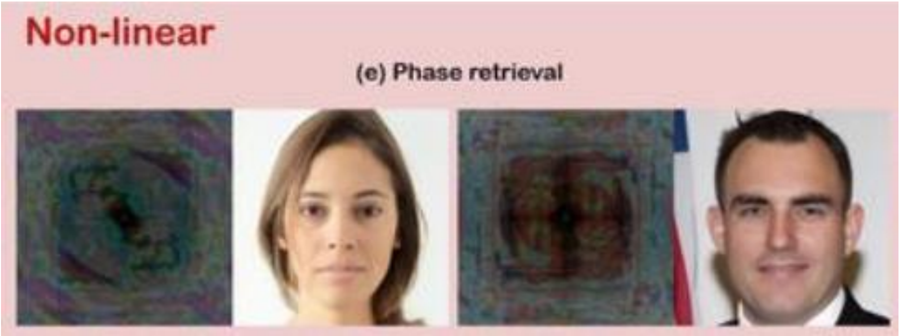
ΠGDM Output

Prospects and challenges

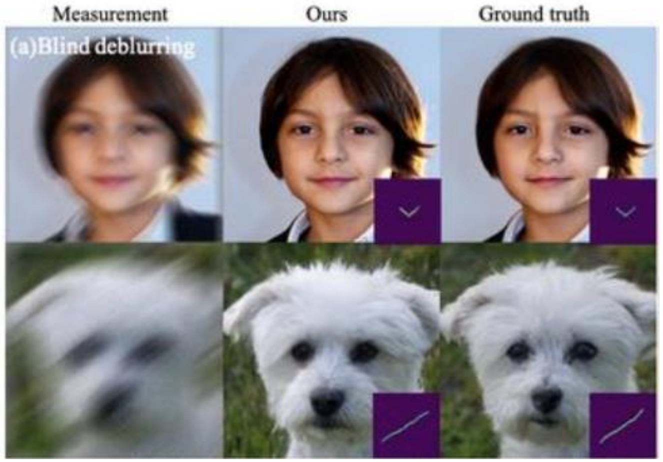
Efficiency: ΠGDM is slower & memory inefficient, due to backpropagation.

Generality: ΠGDM is not suitable to problems without a "pseudoinverse".

Blindness: ΠGDM is limited to "non-blind" inverse problems.



Non-linear problems
Chung *et al*., https://arxiv.org/abs/2209.14687



Blind inverse problems
Chung *et al*., https://arxiv.org/abs/2211.10656

Some efforts on these directions, yet not fast / robust enough!

Summary

Diffusion models can act as efficient priors for inverse problems.

[NeurIPS 2022] Diffusion Denoising Restoration Models
- https://github.com/bahjat-kawar/ddrm
- https://ddrm-ml.github.io/

PhysDiff: Physics-Guided Human Motion Diffusion Model
- https://nvlabs.github.io/PhysDiff/

Pseudoinverse-Guided Diffusion Models for Inverse Problems
- Accepted to ICLR 2023
- Draft: https://openreview.net/forum?id=9_gsMA8MRKQ

Thanks!

https://tsong.me