Generative Plug-and-Play (GPnP): Posterior Sampling for Inverse Problems[†]

(The Saga Continues…)

Charles A. Bouman and *Gregery T. Buzzard,* Purdue University

2023 Allerton Conference on Communications, Control, & Communications
September 28, 2023

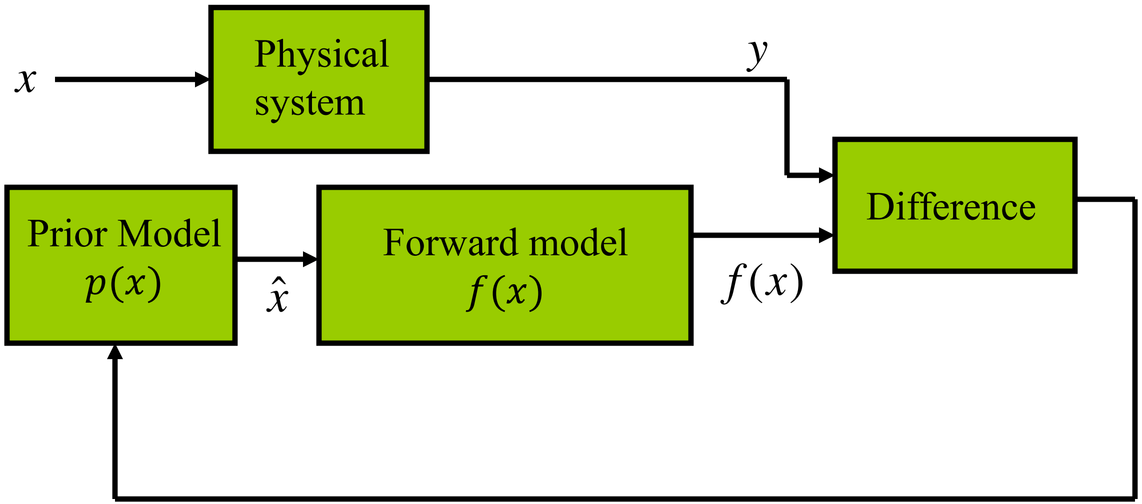
†Thank you to Showalter Foundation, NSF, ORNL, LANL, GE Healthcare, AFRL, Eli Lilly, and DHS

PnP Original Recipe*

- Motivation
- Variable Splitting and proximal maps
- The ADMM Algorithm
- PnP-ADMM

*Singanallur V. Venkatakrishanan, Charles A. Bouman, and Brendt Wohlberg, "Plug-and-Play Priors for Model Based Reconstruction," *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Austin, Texas, USA, December 3-5, 2013.

Model-Based Iterative Reconstruction (MBIR)



$$\hat{x} = \arg\min_{x}\{-\log p(y|x) - \log p(x)\}$$
$$= \arg\min_{x}\left\{\frac{1}{2}\|y - Ax\|_{\Lambda}^{2} - \log p(x)\right\}$$

Fresh Look at MBIR (circa 2013)

- Forward model: $u_1(x) = -\log p(y|x)$
- Prior model: $u_0(x) = -\log p(x)$

- MAP or regularized inverse

$$\hat{x} = \arg\min_x \{u_1(x) + u_0(x)\}$$

Can we minimize these two terms separately?

→ **Proximal maps**

Proximal Maps

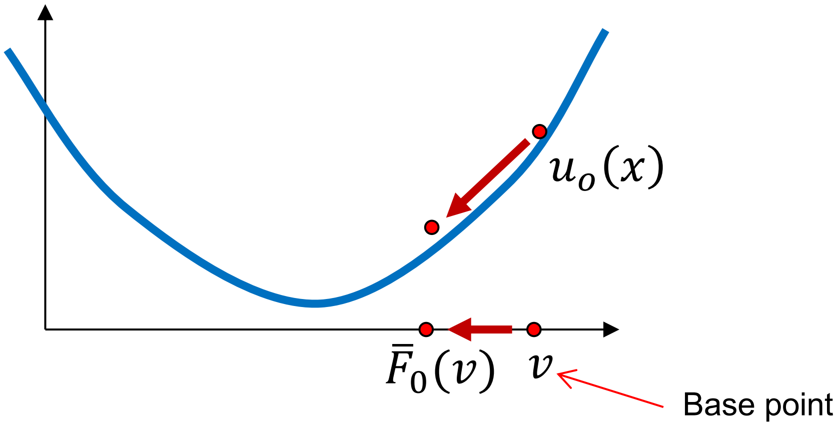
- Proximal map of f with base point x:

$$\bar{F}_0(v) = \arg\min_x \left\{ u_0(x) + \frac{1}{2\gamma^2}\|x - v\|^2 \right\}$$

Base point

Minimize a function

Quadratic "spring" penalty



Prior Proximal Map is a Denoiser

$$\bar{F}_0(v) = \arg\min_x \left\{ u_0(x) + \frac{1}{2\gamma^2} \|x - v\|^2 \right\}$$

- **Denoiser:** When $u_0(x) = -\log p(x)$, the proximal map is a denoiser

$$\bar{F}_0(v) = \arg\min_x \left\{ \underbrace{\frac{1}{2\gamma^2} \|v - x\|^2}_{\text{-Log likelihood for AWGN with variance } \gamma^2} - \log p(x) \right\}$$

-Log likelihood for
AWGN with variance γ^2

$$= \text{Denoise}(v; \gamma)$$

MAP denoiser for AWGN

ADMM for MBIR Reconstruction

Initialize $v, u = 0$
Repeat {

$x \leftarrow \bar{F}_1(v - u)$ // Project onto sensor manifold

$v \leftarrow \bar{F}_0(x + u)$ // Projection onto prior manifold

$u \leftarrow u + (x - v)$ // Augmented Lagrangian update

}

- ADMM:
  - Iteratively reproject on sensor/prior manifolds
  - Minimizes $u(x) = u_1(x) + u_0(x)$

PnP for MBIR Reconstruction

Initialize $v, u = 0$

Repeat {

$x \leftarrow \bar{F}_1(v - u)$ // Project onto sensor manifold

$v \leftarrow \text{Denoise}(x + u)$ // Denoise

$u \leftarrow u + (x - v)$ // Augmented Lagrangian update

}

- Big Idea:
  - Replace F_0 with any denoiser!
  - Does it still converge? Does it minimize anything?

PnP circa 2013

Forward model: sparse subsampling

$$u_1(x) = \sum_{s \in \{\text{sampled}\}} \frac{1}{2} \|x_s - y_s\|^2$$


Ground Truth

Prior model: denoising algorithm

K-SVD

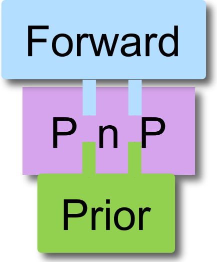

RMSE : 14.11

BM3D


RMSE : 12.56

Subsamples


Noise std. dev : 5% of max signal



PLOW


RMSE : 14.54

TV


RMSE : 15.50

q-GGMRF


RMSE : 15.72

So what's the problem?

- PnP only generates a single "best" result
- Can PnP be modified to generate samples from the posterior distribution?

$$X \sim p_{x|y}(x|y) = \frac{1}{Z} p(y|x)p(x)$$

Generative PnP (GPnP):

- Proximal generators
- Markov chains
- Intuition behind GPnP

Posterior Distribution

- The posterior distribution is given by

$$p(x|y) = \frac{1}{Z}\exp\{-u_1(x) - u_0(x)\}$$

where

$$u_1(x) = -\log p(y|x)$$

$$u_0(x) = -\log p(x)$$

- Strategy:
  - Create Markov chain
  - Proximal generators: create sequential random samples
  - Modular implementation

Proximal Generators

- Proximal Map

$$\bar{F}_0(x) = \arg\min_v \left\{ u_0(v) + \frac{1}{2\gamma^2}\|v - x\|^2 \right\}$$

- Proximal distribution

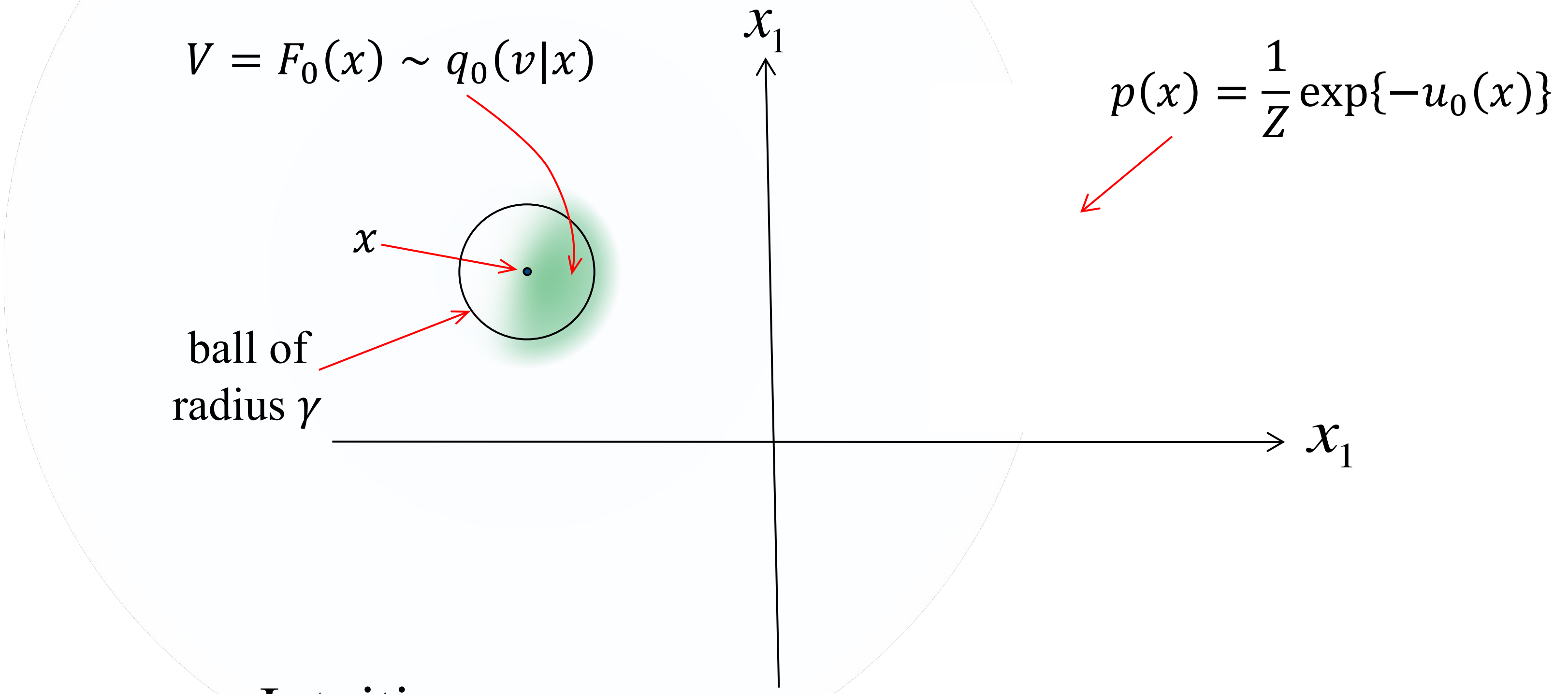
$$q_0(v|x) = \frac{1}{Z}\exp\left\{ -u_0(v) - \frac{1}{2\gamma^2}\|v - x\|^2 \right\}$$

- Proximal Generator

$$V = F_0(x) \sim q_0(v|x)$$

Generates a sample from the proximal distribution

Interpretation of Proximal Generator



- Intuition:
  - Locally samples from the prior distribution
  - Expected change approximates score

Generative PnP

Initialize $X = \text{Random}(0, I) + {}^{1}/_{2}$

Repeat {

$X \leftarrow F_0(X)$ // Prior Model Proximal Generator

$X \leftarrow F_1(X)$ // Forward Model Proximal Generator

}

Return(x)

- Observations/questions:
  - This is a Markov chain
  - Does it converge to a stationary distribution?
  - If so, then what is the stationary distribution?

GPnP Theorem

Theorem: Consider $X_n = F_1\left(F_0(X_{n-1})\right)$, then

- X_n is a reversible Markov chain
- X_n has a stationary distribution given by

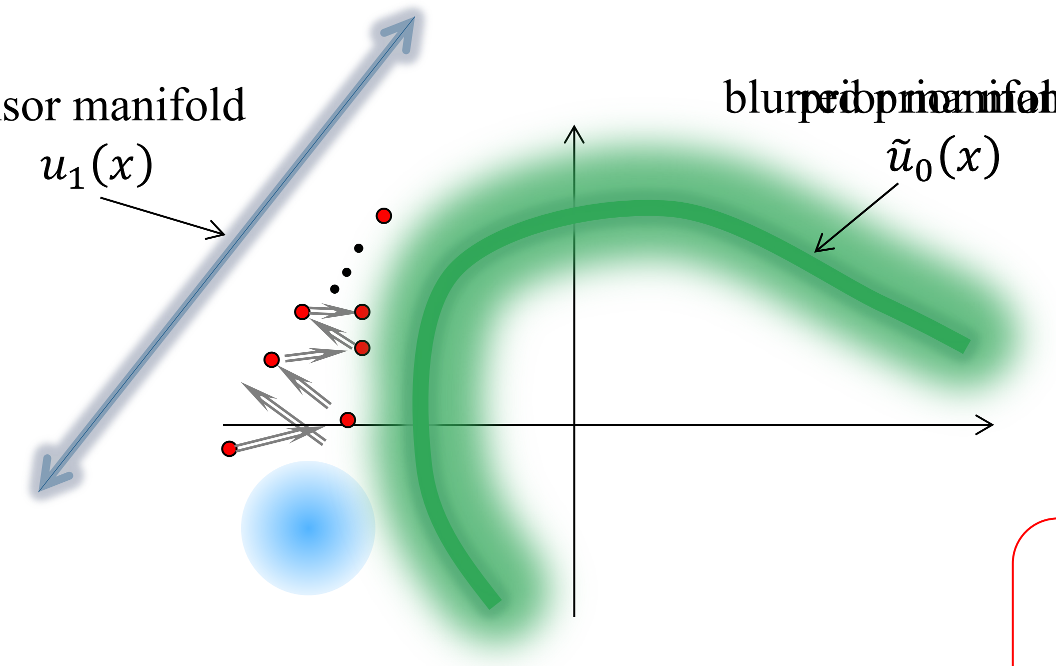
$$\tilde{p}(x|y) = \frac{1}{Z} \exp\{-u_1(x) - \tilde{u}_0(x;\gamma^2)\}$$

  – where $\tilde{u}_0(x;\gamma^2)$ is $u_0(x)$ blurred with a Gaussian noise of variance γ^2.

▪ Bottom line:

  – Repeated sequential application of F_0 and F_1 converges to “desired” distribution.
  – But GPnP introduces AWGN with variance γ^2 to the prior distribution!

Generative Plug-and-Play Intuition



Repeat {

$X \leftarrow F_0(X)$

$X \leftarrow F_1(X)$

}

Implementing Proximal Generators:

- Generic implementation
- Prior model proximal generator
- GPnP Psuedo-code

How to implement the Proximal Generator?

- For γ small, just add white noise!

$$F(x) \approx \bar{F}(x) + \gamma W$$

Proximal generator

Ordinary proximal map

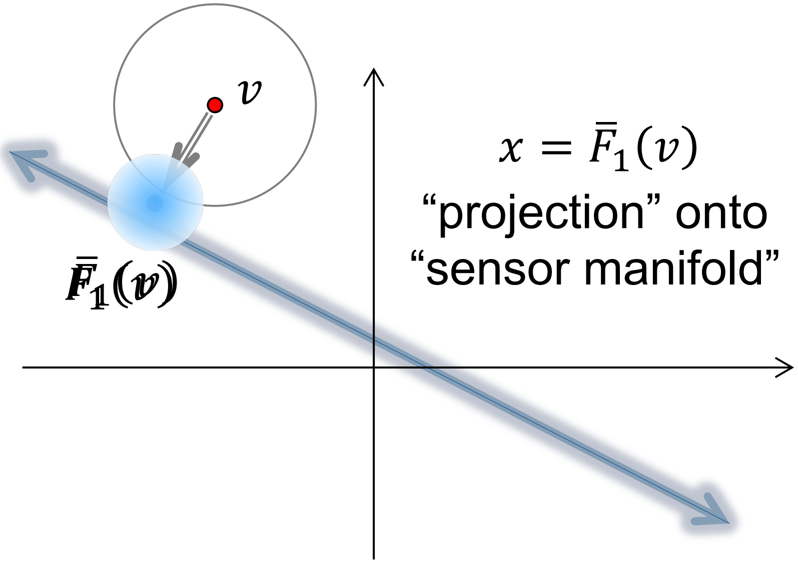
Proximal map parameter

white Gaussian noise

Forward Model Proximal Generator

- For small γ,

$$F_1(v) = \bar{F}_1(v) + \gamma W$$



Prior Proximal Generator (First Order Approx.)

- First order approximation

$$F_0(v) = \bar{F}_0(v) + \gamma W$$

$$\approx \text{Denoise}(v, \gamma) + \gamma W$$

MAP denoiser for AWGN

  – But we can get a better approximation…

Denoising Score Matching (Vincent 2011)*

- Amazing result:
  - The AWGN denoiser provides an exact MMSE estimate of the score

$$-\nabla\tilde{u}_0(x;\sigma^2) = \frac{1}{\sigma^2}[\text{Denoise}(x;\sigma) - x]$$

  - Exactly true for any σ

MMSE denoiser for AWGN

- But….
  - $\tilde{u}_0(x;\sigma^2)$ is the energy function for the "noisy" prior
  - So we have the exact solution, but for a noisy prior

*P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Computation*, 2011.

Prior Proximal Generator (Second Order Approx.)

- Define

noise variance

$$\gamma^2 = \sigma^2 \beta$$

proximal map parameter

Step size $\beta \rightarrow 0$

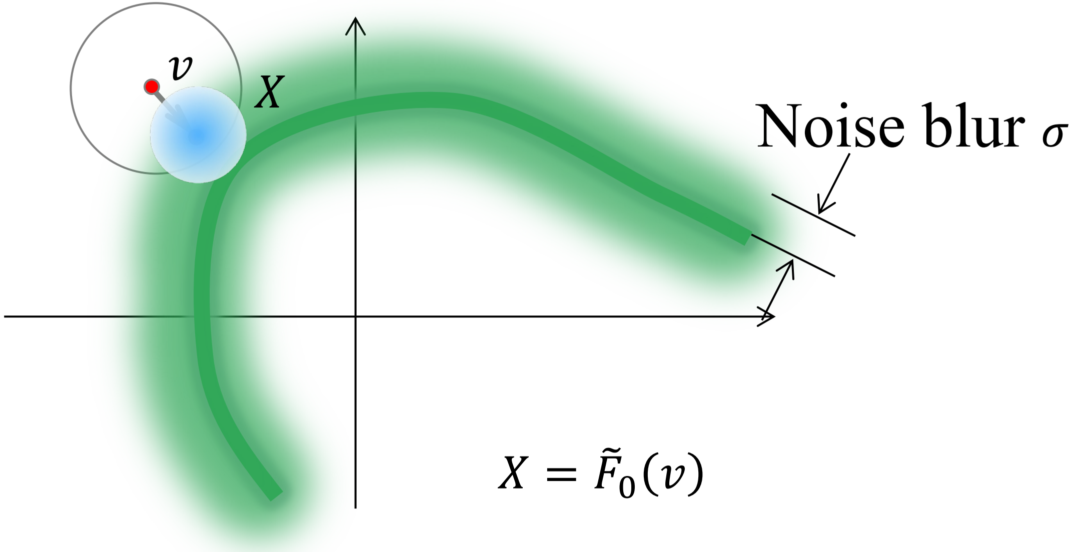
- Better approximation using score matching is:

$$\tilde{F}_0(x; \beta, \sigma) \approx (1 - \beta)x + \beta \text{Denoise}(x; \sigma) + \sqrt{\beta}\sigma W$$

- Remember:
  - $\beta = \frac{1}{4}$ works well in all cases we tried.
  - $\tilde{F}_0$ is based on "noisy" prior, but noise decreases as $\sigma \rightarrow 0$

Prior Model Proximal Generator

$$\tilde{F}_0(x;\beta,\sigma) \approx (1-\beta)x + \beta\text{Denoise}(x;\sigma) + \sqrt{\beta}\sigma W$$



- Prior blurred by σ
- Step size scaled by β

GPnP Basic Algorithm

$\beta = {}^1/_4;\ \sigma_{\max} = 2;$

Initialize $X = \text{Random}(0, I) + {}^1/_2$

Repeat {

$X \leftarrow (1-\beta)X + \beta\text{Denoise}(X;\sigma) + \sqrt{\beta}\sigma\text{RandN}(0,\text{I})$

$X \leftarrow \bar{F}_1(X) + \sqrt{\beta}\sigma\text{RandN}(0,\text{I})$

$\sigma \leftarrow \text{Reduce}(\sigma)$

}

Return(x)

- Prior is blurred by $(1+\beta)\sigma^2$
- But with time $\sigma \to 0$

GPnP Basic Algorithm with Regularization

$\beta = {}^{1}/_{4};\ \sigma_{\max} = 2;\ \alpha = 1.3;$

Initialize $X = \text{Random}(0, I) + {}^{1}/_{2}$

Repeat {

$X \leftarrow (1-\beta)X + \beta\text{Denoise}(X; \alpha\sigma) + \sqrt{\beta}\sigma\text{RandN}(0, \text{I})$

$X \leftarrow \bar{F}_1(X) + \sqrt{\beta}\sigma\text{RandN}(0, \text{I})$

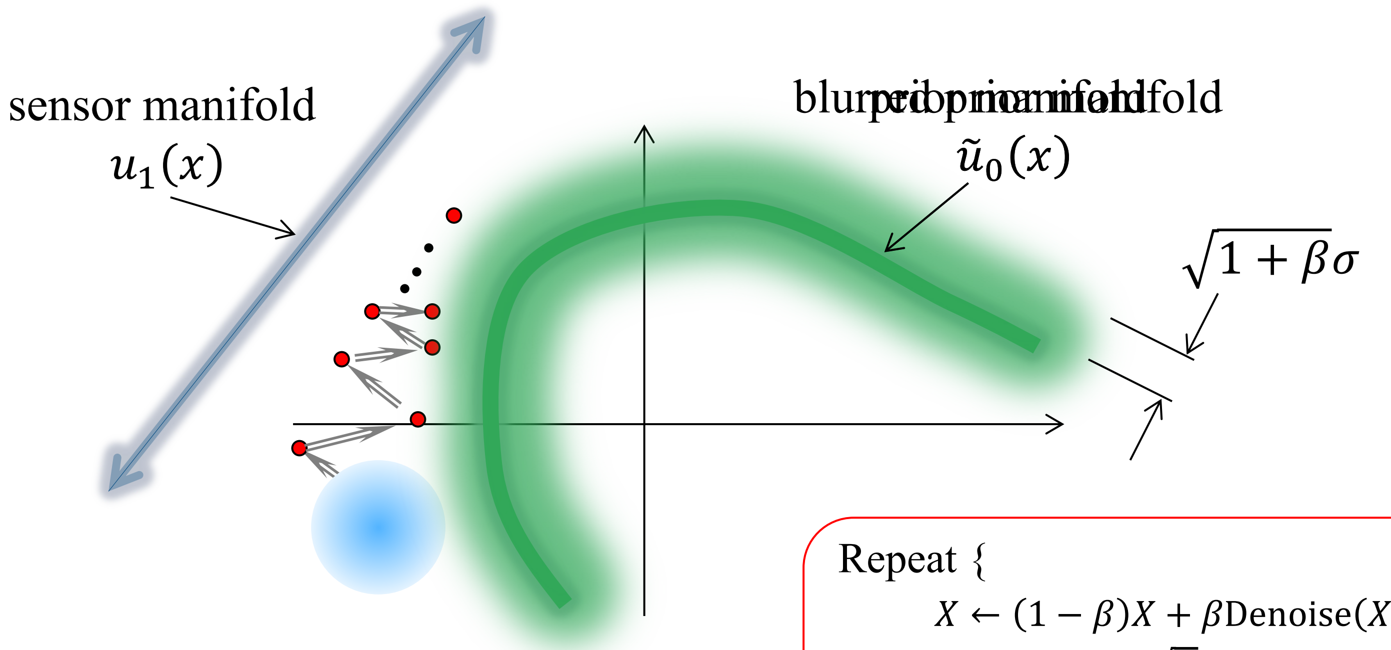
$\sigma \leftarrow \text{Reduce}(\sigma)$

}

Return(x)

- Denoise$(X; \sigma)$ - MMSE denoiser trained for AWGN with variance σ^2.
- Increasing α increases regularization
- Prior is blurred by $(1+\beta)\sigma^2$, but with time $\sigma \to 0$

GPnP Interations



Repeat {

$$X \leftarrow (1-\beta)X + \beta \text{Denoise}(X;\alpha\sigma) + \sqrt{\beta}\sigma \text{W}$$

$$X \leftarrow \bar{F}_1(X) + \sqrt{\beta}\sigma \text{W}$$

$$\sigma \leftarrow \text{Reduce}(\sigma)$$

}

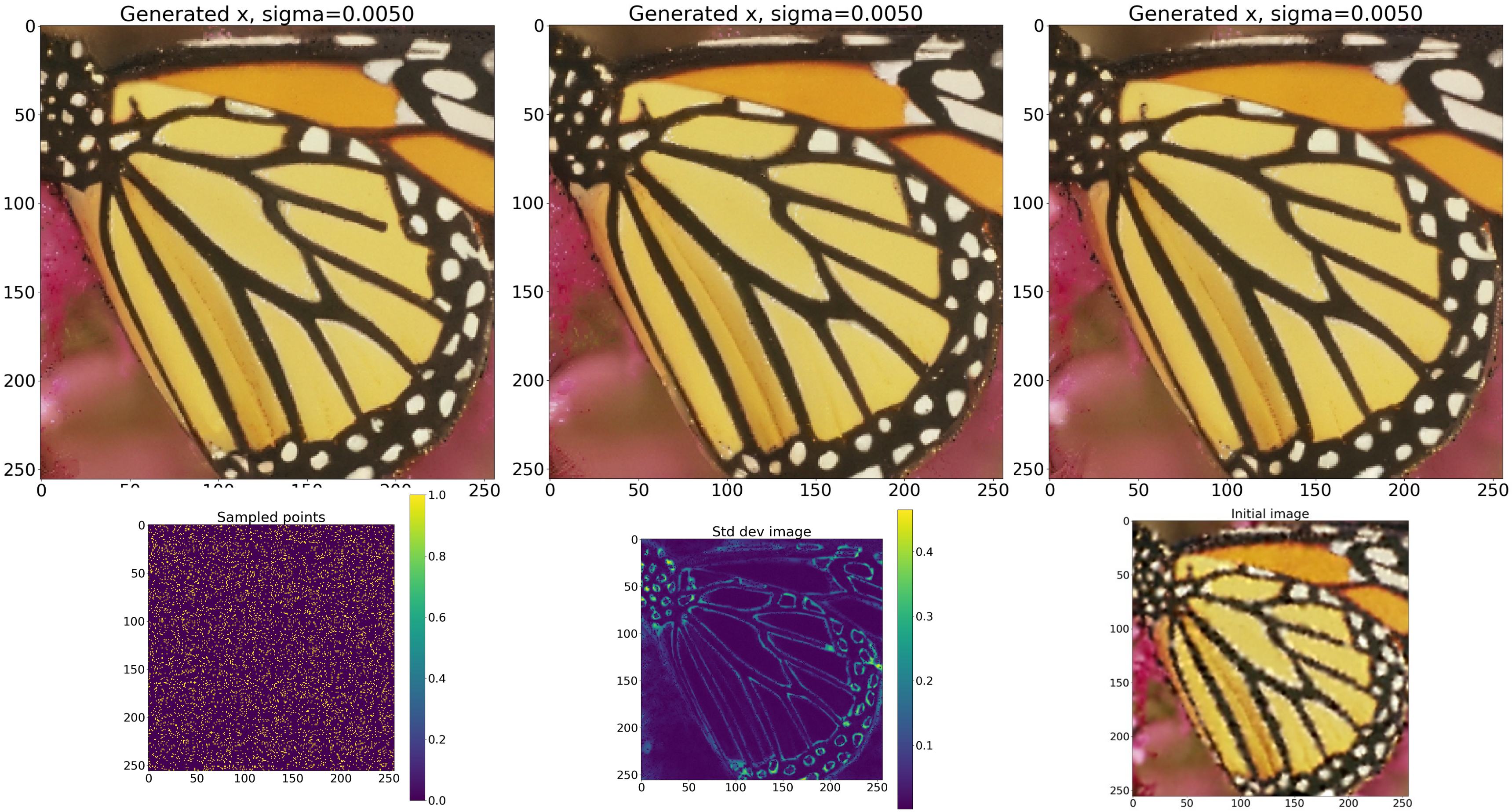
Experiments

- Experiment:
  - Prior proximal generator: BM3D, DRUNet*, DDPM denoiser trained on CelebAHQ-256**
  - Forward model: interpolation with sparse sampling of 10%, 5%, 2% and missing rectangle.
- Parameters
  - $N = 100$; $\sigma_{\max} = 0.5$ or 2.0; $\sigma_{\min} = 0.005$; $\beta = {}^{1}/_{4}$; $\alpha = 1.3$;
  - Same parameters work for different problems (interpolation, tomography, …) and different denoisers (BM3D, DRUNet, …).

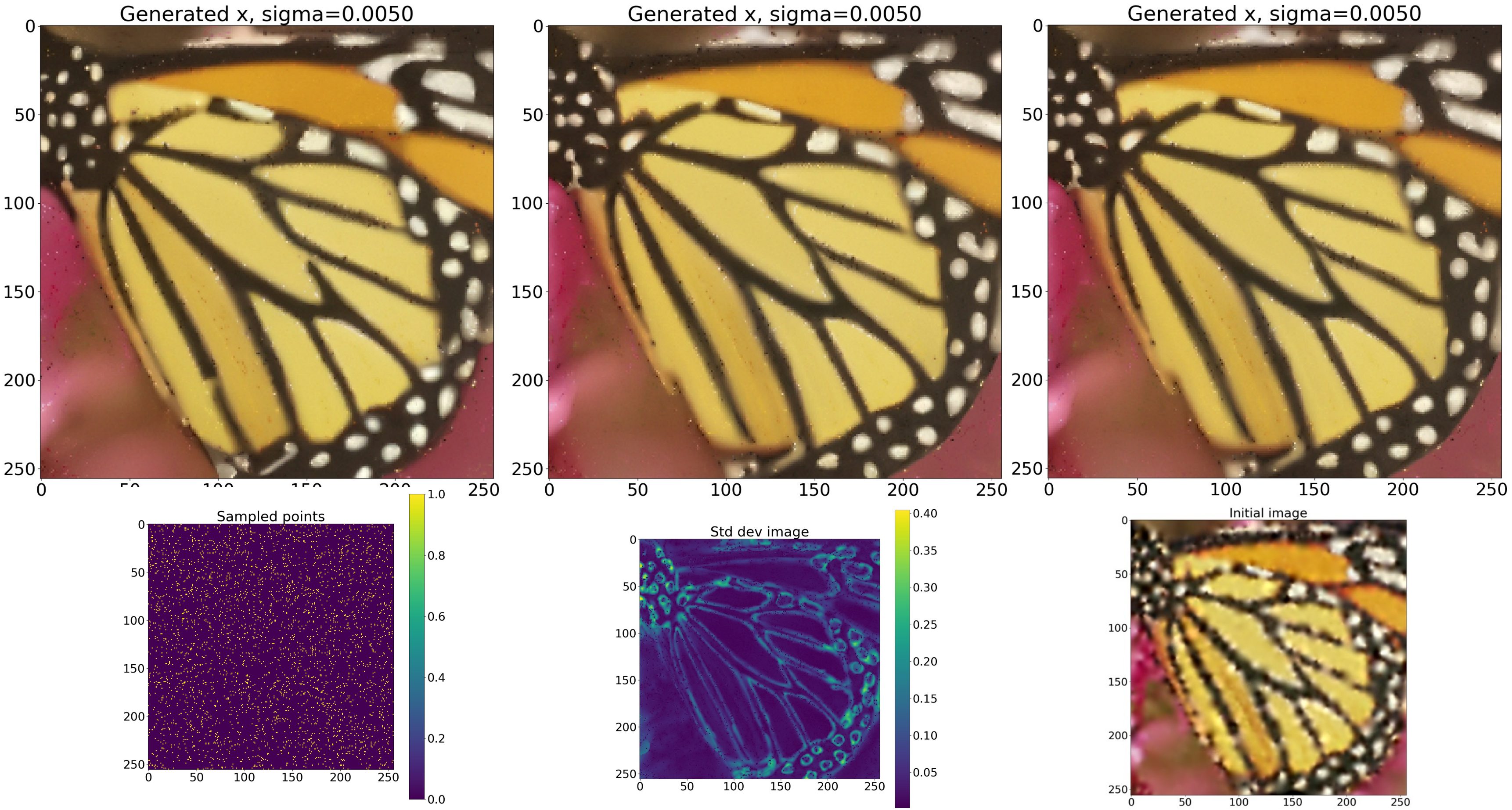
*Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte, "Plug-and-Play Image Restoration With Deep Denoiser Prior," PAMI 2022.

**Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte, "Plug-and-Play Image Restoration With Deep Denoiser Prior," PAMI 2022.

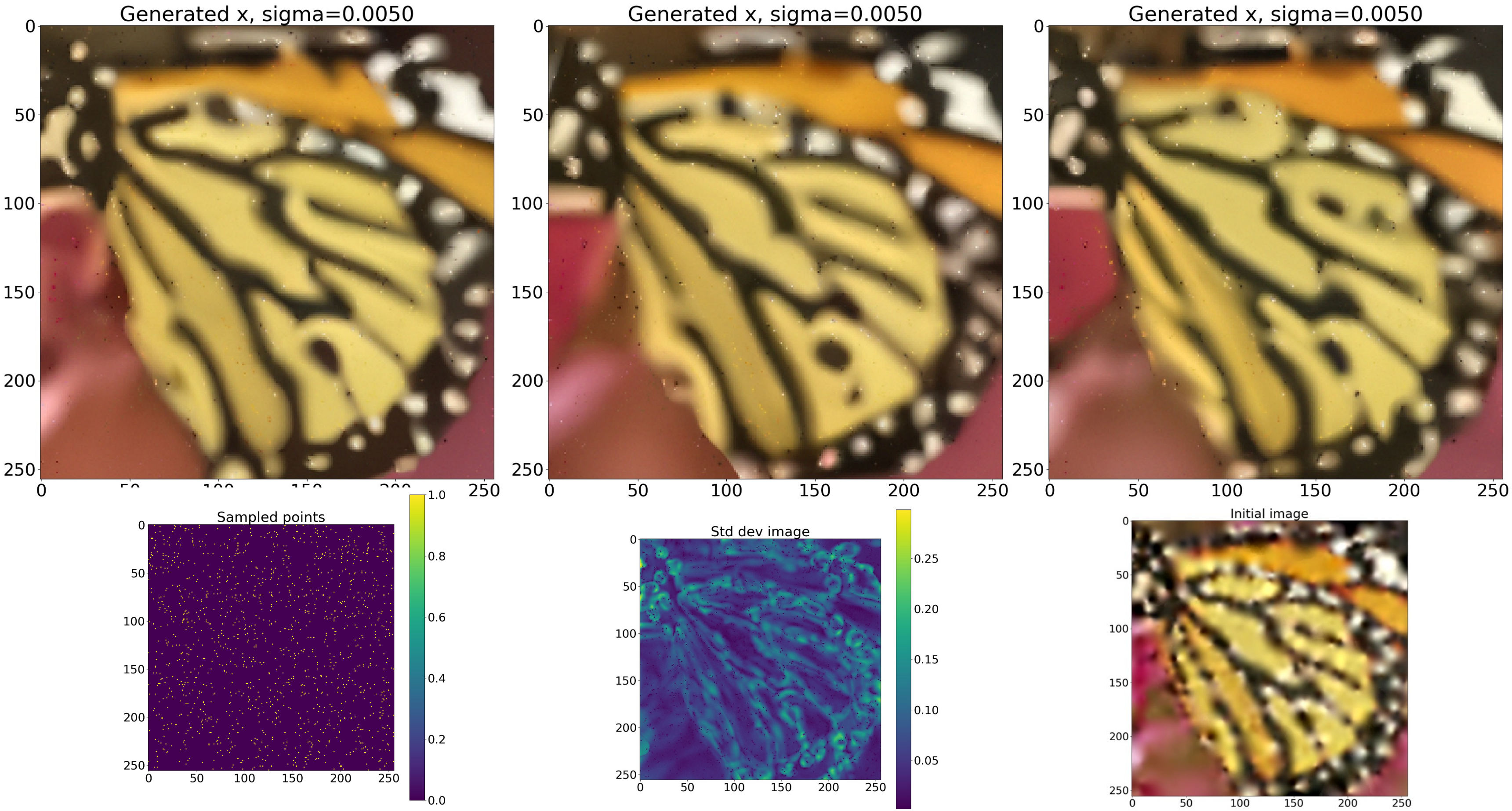
Sparse interpolation: 10% of pixels sampled, DRUNet prior (Std dev intensity window changes)



Sparse interpolation: 5% of pixels sampled, DRUNet prior (Std dev intensity window changes)

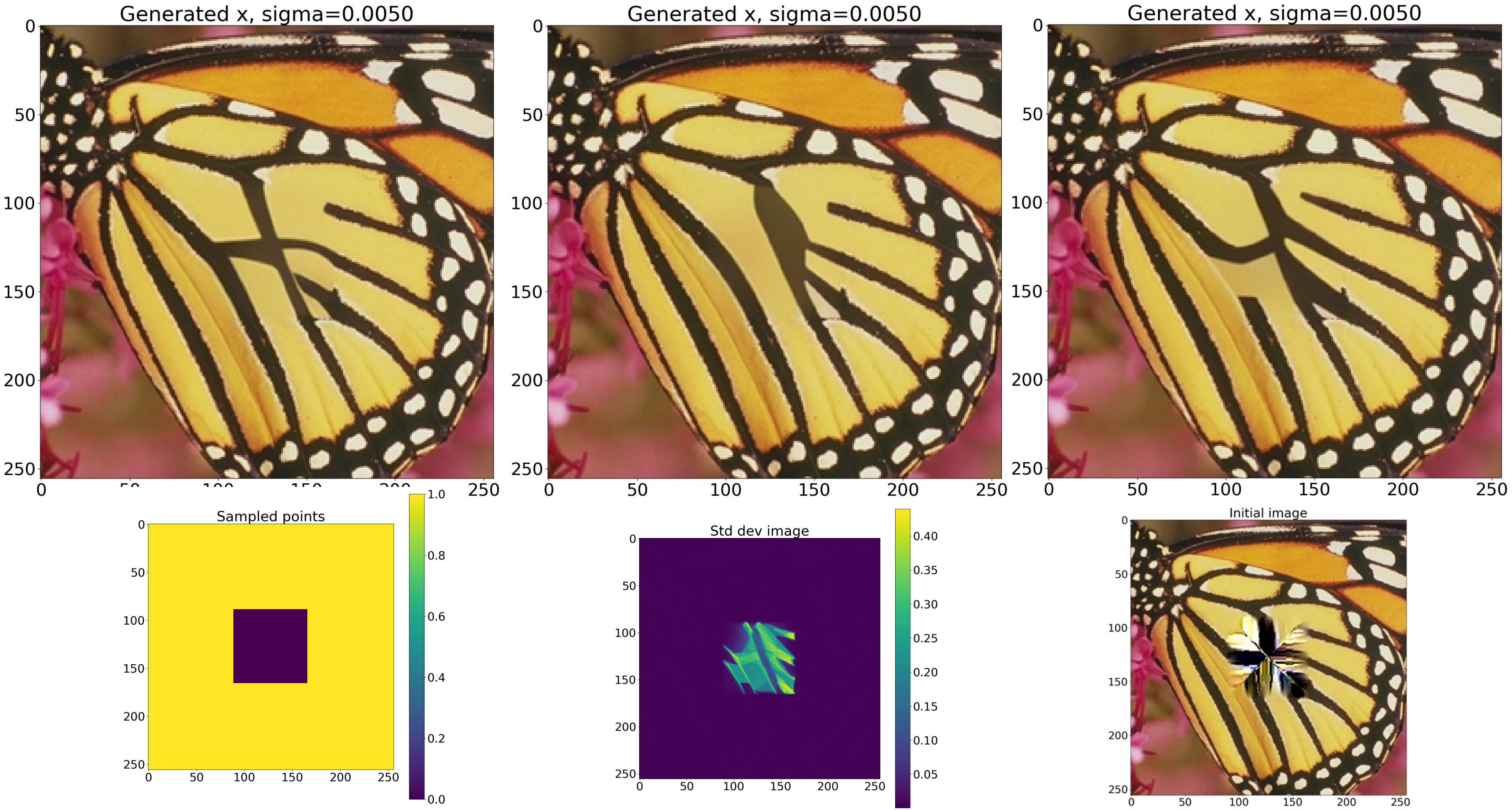


Sparse interpolation: 2% of pixels sampled, DRUNet prior (Std dev intensity window changes)



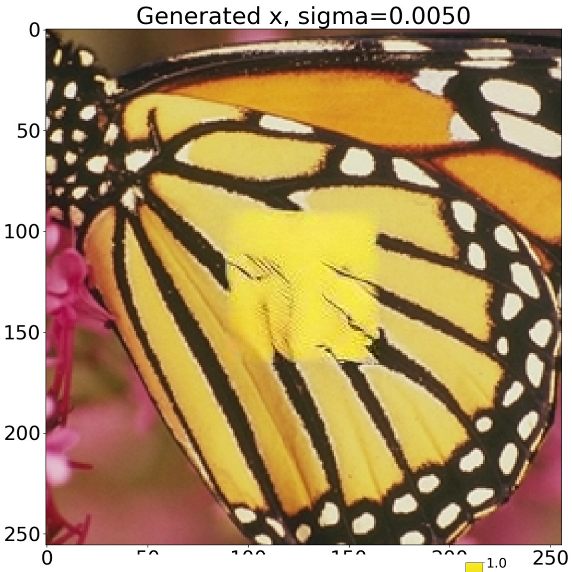
Inpainting:

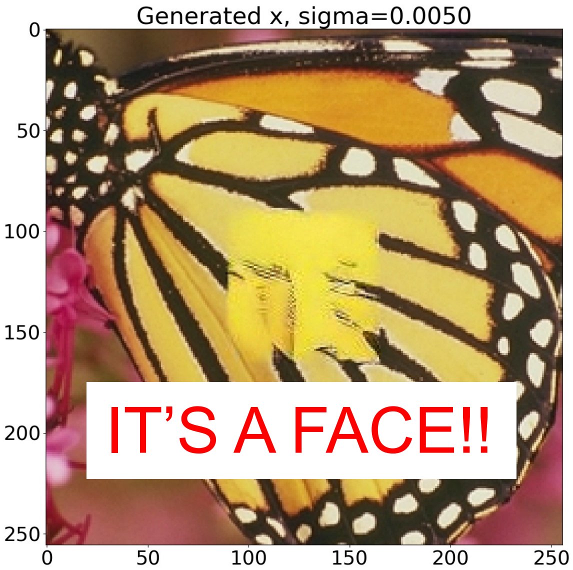
Center rectangle omitted - 3 samples, DRUNet prior
(Std dev intensity window changes)

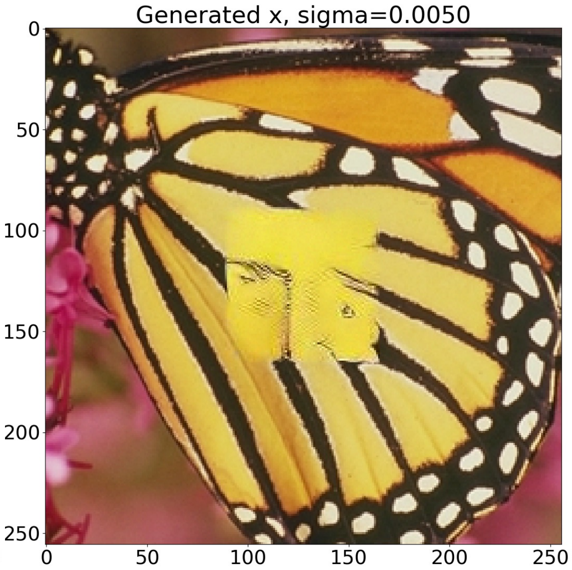


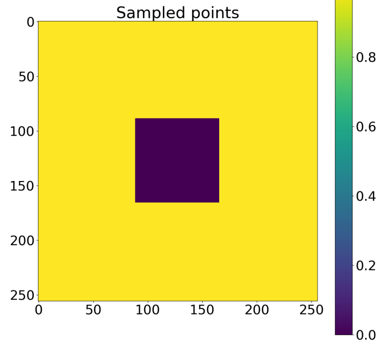
Inpainting:

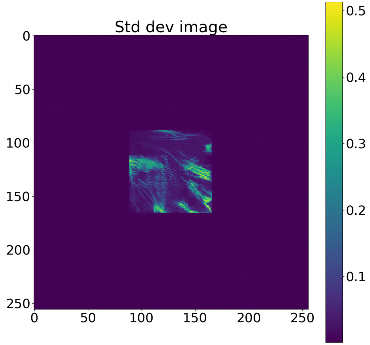
Center rectangle omitted - 3 samples, DDPM denoiser trained on CelebAHQ-256 prior (Std dev intensity window changes)

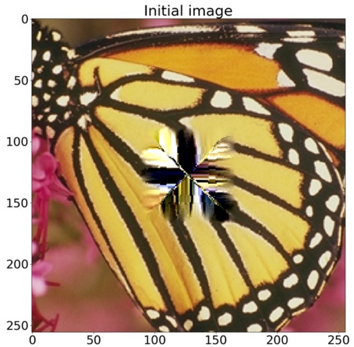












Conclusions

- Generative PnP: A natural generalization of PnP original recipe
  - Denoiser for prior
  - Proximal map for forward model
  - Iterate and add noise
- GPnP vs Langevin Dynamics*:
  - Discrete Markov Chain vs Stochastic Differential Equation
  - Proximal Maps vs Gradient Descent
  - New Approach vs Established Method

*Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, Ben Poole, "Score Based Generative Modeling Through Stochastic Differential Equations," ICLR 2021.