2023 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton) | 979-8-3503-2814-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/Allerton58177.2023.10313413

Generative Plug and Play: Posterior Sampling for Inverse Problems

Charles A. Bouman
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
bouman@purdue.edu

Gregery T. Buzzard
Department of Mathematics
Purdue University
West Lafayette, IN, USA
buzzard@purdue.edu

***Abstract*—Over the past decade, Plug-and-Play (PnP) [1], [2] has become a popular method for reconstructing images using a modular framework consisting of a forward and prior model. The great strength of PnP is that an image denoiser can be used as a prior model while the forward model can be implemented using more traditional physics-based approaches. However, a limitation of PnP is that it reconstructs only a single deterministic image.**

In this paper, we introduce Generative Plug-and-Play (GPnP), a generalization of PnP to sample from the posterior distribution. As with PnP, GPnP has a modular framework using a physics-based forward model and an image denoising prior model. However, in GPnP these models are extended to become proximal generators, which sample from associated distributions. GPnP applies these proximal generators in alternation to produce samples from the posterior. We present experimental simulations using the well-known BM3D denoiser [3]. Our results demonstrate that the GPnP method is robust, easy to implement, and produces intuitively reasonable samples from the posterior for sparse interpolation and tomographic reconstruction. Code to accompany this paper is available at https://github.com/gbuzzard/generative-pnp-allerton.

***Index Terms*—Plug-and-Play (PnP), Inverse problems, Generative model, Posterior sampling**

I. Introduction

The recent explosion in new sensors has led to growing interest in integrating both physical and data driven models for scientific applications. This approach captures the enormous power of modern machine learning methods to model empirical data while also incorporating the benefits of established physics models in imaging applications ranging from optics [4] to X-ray CT [5].

A popular method for integrating physics and machine learning models is Plug-and-Play (PnP) [1]. The key idea behind PnP is that an image denoising algorithm encodes prior information implicitly and can be used in place of a functional prior model commonly found in Bayesian approaches. In practice, PnP alternates the application of a forward model proximal map to fit data and a denoiser representing the prior model. When the denoiser is also a proximal map, then PnP can be viewed as an optimization algorithm [2]. However, more generally PnP is the solution to an equilibrium condition, and under appropriate technical conditions, the algorithm is known to converge to a unique [6] solution.

The desire to understand variation in possible solutions given limited, noisy measurements has driven interest in algorithms to sample from the posterior distribution. Generative adversarial networks (GAN) [7] and variational autoencoders [8] are two possible methods for generating samples from a distribution described by training data. However, while conditional GANs allow the samples to be conditioned on another random quantity [9], neither model provides a modular framework that can be decomposed as a forward and prior model, and GANs can be difficult to stably train [10].

More recently, generative diffusion methods [11] based on denoising score matching (DSM) [12], [13] and Langevin dynamics [14] have displayed remarkable generative capabilities. These algorithms do not require adversarial training and have been reported to produce very high quality results [15]. A number of groups have investigated the use of these generative diffusion methods as a prior model that works along with a separate physics-based forward model [16], [17], [18], [19].

In this paper, we introduce Generative Plug-and-Play (GPnP), a method for sampling from the posterior distribution of a model. As with PnP, GPnP has a modular framework based on a forward and prior model in which the prior model is implemented with a denoiser. The GPnP algorithm alternately applies a forward model and a prior model, each in the form of a proximal generator. These proximal generators are similar in formulation to a proximal map but generate random rather than deterministic outputs.

Our primary theoretical result is a theorem that this alternating sequence of random functions generates a Markov chain (MC) with the desired stationary distribution. We then show how the methods of denoising score matching [12] can be used to approximate the prior proximal generator with a denoiser plus some AWGN. We also describe how to compute or approximate the forward model proximal generator in several common cases.

We note that GPnP differs from generative diffusion methods in that it (a) formulates the solution as the stationary distribution of a discrete-time MC; (b) does not use a Langevin dynamics to generate the solution; (c) incorporates proximal generators rather then gradient updates. However, we do show that in the special case of a null forward model, GPnP generates an MC that is exactly the Langevin dynamics for generation of samples from a prior distribution.

We present experimental simulations using the well-known

BM3D denoiser [3]. These results demonstrate that the GPnP method is robust, easy to implement, and produces intuitively reasonable samples from the posterior for sparse interpolation and tomographic reconstruction.

II. Generative PnP Theory

Let $u_0(x)$ and $u_1(x)$ be two non-negative integrable energy functions; that is, $u_0, u_1 : \mathbb{R}^p \to [0, \infty)$ and

$$Z = \int_{\mathbb{R}^p} \exp\{-u_1(x) - u_0(x)\}\, dx < \infty \ .$$

Then our goal will be to generate samples from the distribution

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

with the interpretation that u_1, u_0 are the energy functions for the data distribution and prior distribution, respectively.

A. Proximal Distributions and Generators

To do this, we introduce the *proximal distributions* given by

$$q_0(x|v) = \frac{1}{Z_0(v)} \exp\left\{-u_0(x) - \frac{1}{2\gamma^2}\|x - v\|^2\right\} \tag{2}$$

$$q_1(x|v) = \frac{1}{Z_1(v)} \exp\left\{-u_1(x) - \frac{1}{2\gamma^2}\|x - v\|^2\right\} \tag{3}$$

where γ is a parameter of the proximal distribution and again $Z_0(v)$ and $Z_1(v)$ are normalizing constants that depend on v. By assumption, $u_0 \geq 0$ and $u_1 \geq 0$, so the quadratic term implies that $Z_i(v) < \infty$ for all $v \in \mathbb{R}^N$.

With the proximal distributions, we define *proximal generators* denoted by $F_0(v)$ and $F_1(v)$. Intuitively, a proximal generator generates a new independent random variable with the proximal distribution. More specifically, let

$$Y_0 = F_0(V) \tag{4}$$

$$Y_1 = F_1(V) \ , \tag{5}$$

where V is a random vector in $\mathbb{R}^p$. Then Y_0 and Y_1 are assumed conditionally independent of any previously generated random vectors given V, and the conditional densities of Y_0 and Y_1 given V are given above in (2) and (3), respectively.

B. Markov Chains from Proximal Generators

We can produce a Markov chain (MC) by repeatedly applying the proximal generators. More specifically, each new state of the MC is generated from the previous state by applying the two proximal generators in sequence.

$$X_n = F_1(F_0(X_{n-1})). \tag{6}$$

Ideally, by repeatedly applying this sequence of operations, the random vector X_n will converge in distribution to samples from $p(x)$. This isn't quite true, but the following theorem, proved in the appendix, states that when γ is small, then the MC has a stationary distribution near $p(x)$ in (1) with u_0 replaced by a Gaussian convolution approximation to u_0.

Theorem 1. *Let X_n be a Markov chain given by*

$$X_n = F_1(F_0(X_{n-1})) \ . \tag{7}$$

Then X_n forms a reversible Markov chain with a stationary distribution given by

$$X_n \sim \tilde{p}_{\gamma^2}(x) = \frac{1}{Z'} \exp\left\{-u_1(x) - \tilde{u}_0(x;\gamma^2)\right\} \ , \tag{8}$$

where

$$\tilde{u}_0(x;\gamma^2) = -\log\left(e^{-u_0(x)} * g_{\gamma^2}(x)\right) \ , \tag{9}$$

and $*$ denotes multidimensional convolution with a Gaussian density of variance γ^2 given by

$$g_{\gamma^2}(x) = \frac{1}{(2\pi\gamma^2)^{p/2}} \exp\left\{-\frac{1}{2\gamma^2}\|x\|^2\right\} \ . \tag{10}$$

This theorem serves as the basis for the generative Plug-and-Play (GPnP) algorithm. Assuming that the MC is ergodic, as $n \to \infty$, the GPnP algorithm will converge to the stationary distribution, $\tilde{p}_{\gamma^2}(x)$. Furthermore, this stationary distribution has the property that

$$p(x) = \lim_{\gamma \to 0} \tilde{p}_{\gamma^2}(x) \ , \tag{11}$$

so the samples of the MC become close to the desired distribution as $n \to \infty$ and $\gamma \to 0$.

III. Sampling from the Posterior

In this section, we show how GPnP can be used to generate samples from the posterior distribution for a canonical inverse problem with data y and object of interest x. Given a prior distribution $p_0(x)$ and a forward model $p_{y|x}(y|x)$, we define

$$u_0(x) = -\log p_0(x) + C_0 \tag{12}$$

$$u_1(x) = -\log p_{y|x}(y|x) + C_1 \ . \tag{13}$$

By Bayes' rule, the posterior distribution of X given Y can be expressed as

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

Note that this has the same form as (1). So Theorem 1 implies that the GPnP algorithm can be used to sample from the posterior distribution.

In order to implement the GPnP algorithm, we will need to implement both the forward proximal generator $F_1(v)$ and the prior proximal generator $F_0(v)$.

To implement the prior proximal generator, we use the recent theory of denoising score matching [12]. This theory relates the MMSE denoiser for noise variance of σ^2 to a modified noisy prior distribution given by

$$\tilde{p}_{0,\sigma^2}(x) = (p_0 * g_{\sigma^2})(x) \ ,$$

which is a blurred version of the true prior distribution $p_0(x)$. The associated energy function for $\tilde{p}_{0,\sigma^2}$ is then given by

$$\tilde{u}_0(x;\sigma^2) = -\log \tilde{p}_{0,\sigma^2}(x) + C_0 \ .$$

As before, $\tilde{u}_0(x;\sigma^2)$ is not exactly the desired energy function of $u_0(x)$, but as $\sigma \to 0$ it becomes a good approximation. Hence we use this energy function to implement the prior proximal generator $\tilde{F}_0(v;\sigma)$ in the GPnP algorithm.

Note that the blur introduced from this σ^2-denoiser is independent from the noise introduced by γ in the GPnP Algorithm as specified in Theorem 1. This means that if we use u_1 and $\tilde{u}_0$ as the forward and prior energy functions, then GPnP will generate samples from the posterior distribution

$$\tilde{p}_{x|y}(x|y;\sigma^2+\gamma^2) = p_{y|x}(y|x)\tilde{p}_{0,\sigma^2+\gamma^2}(x) \ ,$$

where $\tilde{p}_{0,\sigma^2+\gamma^2}(x) = (\tilde{p}_{0,\sigma^2} * g_{\gamma^2})(x)$ is a version of the prior distribution that is blurred with a Gaussian of variance $\sigma^2+\gamma^2$. Again, as σ and γ become small, we get that

$$p_{x|y}(x|y) = \lim_{\sigma\to 0}\lim_{\gamma\to 0}\tilde{p}_{x|y}(x|y;\sigma^2+\gamma^2) \ .$$

So we can use the GPnP algorithm to generate samples from the true posterior distribution of X given Y.

The following sections provide more details on how to implement the proximal generators $\tilde{F}_0(v)$ and $F_1(v)$.

A. Prior Model Proximal Generator

In this section, we show how to implement the proximal generator, $X = \tilde{F}_0(v)$ of the previous section. We first define the score of the blurred distribution as

$$s(x;\sigma^2) = -\nabla_x \tilde{u}_0(x;\sigma^2) \ . \tag{14}$$

Vincent showed the amazing result that this score can be estimated by minimizing the Denoising Score Matching (DSM) loss [12]. For the special case of AWGN, the DSM has the form [13],

$$\text{Loss}(\theta;\sigma) = E\left[\left\|\frac{W}{\sigma} + s_\theta(X+\sigma W)\right\|^2\right] \ , \tag{15}$$

where s_θ is a function parameterized by θ, $X \sim p_0(x)$ is a random image from the desired prior distribution, and $W \sim N(0,I)$ is independent Gaussian white noise.

The key result of Vincent's work is that the loss in (15) is minimized when the function $s_{\theta_\sigma}(x)$ is equal to the score, $s(x;\sigma^2)$. To best estimate the score of the blurred distribution, we choose θ to be

$$\theta_\sigma = \arg\min_\theta \text{Loss}(\theta;\sigma) \ .$$

A more traditional point of view is that (15) implies that the MMSE denoiser with AWGN of variance σ^2 is given by

$$\text{Denoise}(x;\sigma) = x + \sigma^2 s_{\theta_\sigma}(x) \ . \tag{16}$$

From this, we see that if we have an MMSE denoiser designed for a noise variance of σ^2, then we can compute an estimate of the score as

$$s_{\theta_\sigma}(x) = \frac{1}{\sigma^2}\left[\text{Denoise}(x;\sigma) - x\right] \ . \tag{17}$$

Then a first order Taylor series and completing the square yields an approximate proximal distribution given by

$$\begin{aligned}
&\tilde{q}_0(x|v;\sigma^2) \\
&= \frac{1}{Z(v)}\exp\left\{-\tilde{u}_0(x;\sigma^2) - \frac{1}{2\gamma^2}\|x-v\|^2\right\} \\
&\approx \frac{1}{Z'(v)}\exp\left\{(x-v)^t s_{\theta_\sigma}(v) - \frac{1}{2\gamma^2}\|x-v\|^2\right\} \\
&= \frac{1}{Z''(v)}\exp\left\{-\frac{1}{2\gamma^2}\|x-[v+\gamma^2 s_{\theta_\sigma}(v)]\|^2\right\} \ .
\end{aligned} \tag{18}$$

Notice that for this approximation to be accurate, we need that $\gamma << \sigma$ so that the second derivative of the score function is small relative to $1/\gamma^2$. In order to ensure this, we will express our results in terms of $\beta = \gamma^2/\sigma^2$, where we will pick the parameter $\beta < 1$.

Combining (17) and (18), we can rewrite the proximal generator as

$$\tilde{F}_0(v;\beta,\sigma) \approx (1-\beta)v + \beta\,\text{Denoise}(v;\sigma) + \sqrt{\beta}\sigma W \ , \tag{19}$$

where $W \sim N(0,I)$, $\beta < 1$, and $\text{Denoise}(v,\sigma)$ is an MMSE denoiser designed to remove AWGN of variance σ^2.

B. Forward Model Proximal Generator

We first consider the case in which $u_1(x)$ has two continuous derivatives. In this case, we denote the proximal map for u_1 as

$$\bar{F}_1(v;\gamma) = \arg\min_{x\in\mathbb{R}^p}\left\{u_1(x) + \frac{1}{2\gamma^2}\|x-v\|^2\right\} \ . \tag{20}$$

Again, a first order approximation for u_1, this time centered at the proximal point $\bar{F}_1(v;\gamma)$, implies that for γ small, we can express the proximal distribution as

$$q_1(x|v;\gamma) \approx \frac{1}{Z}\exp\left\{-\frac{1}{2\gamma^2}\|x-\bar{F}_1(v;\gamma)\|^2\right\} \ .$$

So then for small γ, the forward model proximal generator can be implemented as

$$F_1(v;\gamma) \approx \bar{F}_1(v;\gamma) + \gamma W \ , \tag{21}$$

where $W \sim N(0,I)$.

From this we see that for sufficiently small values of γ, we can approximate the forward model proximal generator as the forward model proximal map plus Gaussian white noise. However, in some cases we can practically implement a more accurate proximal generator for larger values of γ as discussed in Sections IV-A and IV-B.

C. The GPnP Algorithm

Algorithm 1 provides a pseudo-code implementation of the GPnP algorithm that starts with a large value of σ and then iterates the GPnP proximal generators $\tilde{F}_0$ and F_1 for each value of σ. Notice that the prior proximal generator, $\tilde{F}_0$, is implemented with the approximation of (19), and the forward proximal generator, F_1, is implemented as described in (21) with $\gamma = \sqrt{\beta}\sigma$.

procedure GPNP-BASIC($\alpha, \beta, \sigma_{max}, \sigma_{min}, N$)
  $X \leftarrow \sigma_{max}\text{RandN}(0, I) + 1/2$
  $a \leftarrow \left(\frac{\sigma_{min}}{\sigma_{max}}\right)^{1/N}$
  for $n = 0$ to $N-1$ **do**
    $\sigma \leftarrow a^n \sigma_{max}$
    $X \leftarrow (1-\beta)X + \beta\,\text{Denoise}(X; \alpha\sigma) + \sqrt{\beta}\sigma\,\text{RandN}(0, I)$
    $X \leftarrow \bar{F}_1(X; \sqrt{\beta}\sigma) + \sqrt{\beta}\sigma\,\text{RandN}(0, I)$
  end for
  return X
end procedure

Fig. 1. Generative Plug-and-Play where $\alpha \in [1, 1.5]$, $0 < \beta << 1$, σ decreases by the multiplicative factor a on each iteration, $\bar{F}_1$ denotes the forward model proximal map, and Denoise$(x; \sigma)$ performs MMSE denoising of an image x with AWGN of variance σ^2.

The decreasing sequence of σ values is known as annealing and has been shown to dramatically speed convergence to the stationary distribution of the Markov chain by more stably modeling the distribution in low probability regions of the space [11]. In our experiments, we have found that $\beta = 0.25$ works well. We also incorporate a parameter α to modulate the strength of the denoiser. This is used to account for inaccuracies in denoiser calibration and to account for the fact that σ decreases on each iteration, which means we need to denoise at a higher rate than the current value of σ. We use $\alpha \approx 1.3$ in our experiments.

IV. Special Proximal Generators

In this section, we discuss some special cases of proximal generators that can be useful in practice.

A. Proximal Generator: Linear Forward Model

In this section, we show how to implement the proximal generator, $X = F_1(v)$, where $X \sim q_1(x|v)$, for a general linear forward model. To do this, consider a linear forward model of the form

$$Y = AX + W \ , \tag{22}$$

where $W \sim N(0, \Lambda^{-1})$, A is a linear forward operator, and Λ is a positive definite precision matrix. Then the energy function associated with this forward model is given by

$$u_1(x) = \frac{1}{2}\|y - Ax\|^2_\Lambda \ . \tag{23}$$

The first-order optimality conditions imply that the proximal map for this function is given by

$$\bar{F}_1(v; \gamma) = v + \left(A^T\Lambda A + \frac{1}{\gamma^2}I\right)^{-1} A^t\Lambda(y - Av) \ . \tag{24}$$

We define R to be the conditional covariance given by

$$R = \left(A^T\Lambda A + \frac{1}{\gamma^2}I\right)^{-1} \ . \tag{25}$$

The energy function for the proximal distribution for u_1 is the objective in (20), which has a minimum at the conditional mean $\bar{F}_1(v; \gamma)$ and which has Hessian R. Since the energy function is quadratic, the proximal distribution is exactly

$$q_1(x|v; \gamma) = \frac{1}{Z}\exp\left\{-\frac{1}{2}\|x - \bar{F}_1(v; \gamma)\|^2_{R^{-1}}\right\} \ .$$

From this, we see that the proximal generator is given by

$$F_1(v; \gamma) = \bar{F}_1(v; \gamma) + W_R \ ,$$

where $W_R \sim N(0, R)$.

	N	σ_{max}	σ_{min}	β	α	σ_y
Subsampling	100	0.5	0.005	0.25	1.3	0.005
Tomography	100	0.5	0.005	0.25	1.3	0.25

TABLE I
PARAMETERS USED IN EXPERIMENTS

B. Proximal Generator: Subsampling

Another useful special case occurs when our measurements are samples at selected pixels. Let S_m be a set of measurement points so that

$$Y_s = X_s + W_s \ ,$$

where $W_s \sim N(0, \sigma_y^2)$ are i.i.d. noise samples. In this case, the energy function associated with this forward model is given by

$$u_1(x) = \sum_{s \in S_m} \frac{1}{2\sigma_y^2}(y_s - x_s)^2 \ . \tag{26}$$

The first-order optimality conditions imply that the associated proximal map is given by

$$\left[\bar{F}_1(v; \gamma)\right]_s = \begin{cases} v_s + \frac{\gamma^2}{\sigma_y^2 + \gamma^2}(y_s - v_s) & \text{if } s \in S_m \\ v_s & \text{if } s \notin S_m \ . \end{cases} \tag{27}$$

From the result of Section IV-A the forward model proximal generator is given by

$$[F_1(v; \gamma)]_s = \begin{cases} \left[\bar{F}_1(v; \gamma)\right]_s + \sqrt{\frac{\sigma_y^2\gamma^2}{\sigma_y^2 + \gamma^2}}W_s & \text{if } s \in S_m \\ \left[\bar{F}_1(v; \gamma)\right]_s + \gamma W_s & \text{if } s \notin S_m \end{cases} \ .$$

where $W_s \sim N(0, 1)$ are i.i.d. Gaussian random variables.

C. Sampling from the Prior

Below we derive the update equations for sampling from the prior distribution. In this case, we set $u_1(x) = 0$, so from (21), the forward proximal generator is exactly

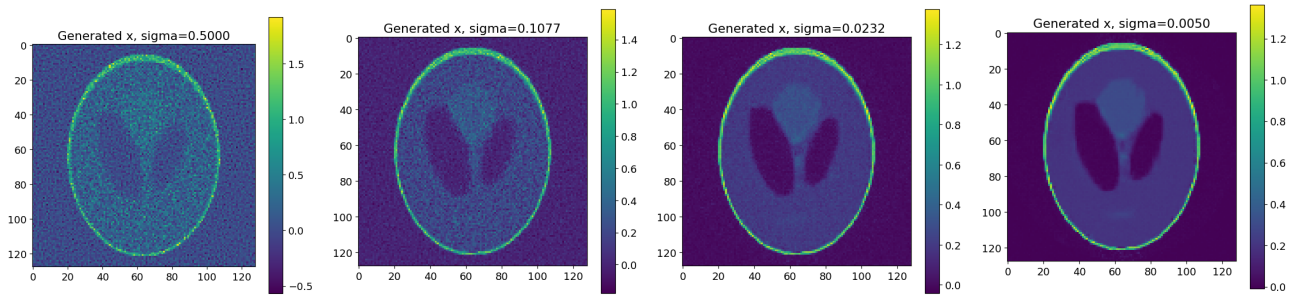
$$F_1(v; \gamma) = v + \gamma W \ ,$$

where $W \sim N(0, I)$. Then for small γ, (16), (19), and the definition of β imply that the prior model proximal generator is

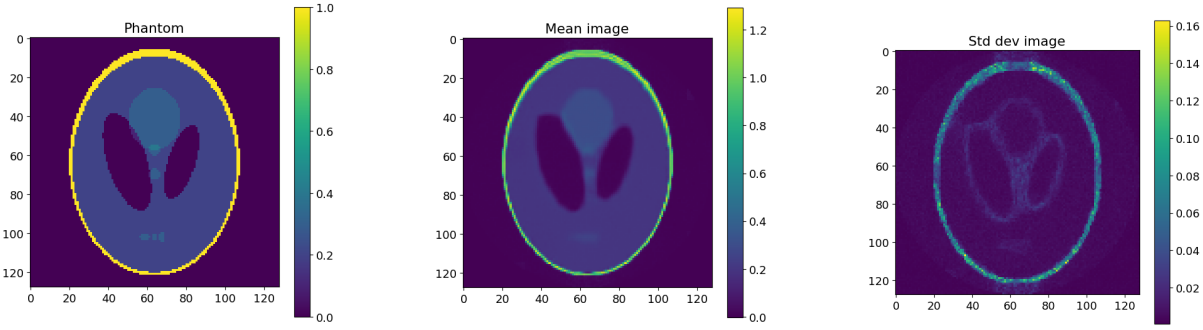
$$\tilde{F}_0(v) = v + \gamma^2 s_{\theta_\sigma}(v) + \gamma W' \ ,$$

where $W' \sim N(0, I)$ is independent of W. Taking the composition of $\tilde{F}_0$ followed by F_1 results in the update

$$X_n = X_{n-1} + \gamma^2 s_{\theta_\sigma}(X_{n-1}) + \sqrt{2}\gamma W \ ,$$



(a) $\sigma = 0.500$ (b) $\sigma = 0.107$ (c) $\sigma = 0.023$ (d) $\sigma = 0.005$



(e) Ground Truth (f) Mean over 10 samples (g) Std Dev over 10 samples

Fig. 2. Tomographic experiment for 128×128 phantom with 16 views: (a) - (d) GPnP outputs with decreasing values of σ; (e) phantom; (f) mean and (g) standard deviation over 10 trials.

which is the familiar Langevin update equation [14]. Rewriting in terms of the denoiser and the parameters β, σ results in the following recursion that generates samples from the posterior distribution for small σ

$$X_n = (1-\beta)X_{n-1} + \beta\,\text{Denoise}(X_{n-1};\sigma) + \sqrt{2\beta}\sigma W \ , \quad (28)$$

where $W \sim N(0, I)$.

V. Results

In this section, we present experimental results using the GPnP algorithm to sample from the posterior distribution of a model. We consider the cases of sparse image interpolation from Section IV-B and 2D parallel-beam, sparse-view tomographic reconstruction from Section IV-A. Table I lists the parameters used for both experiments. For both experiments, the BM3D denoiser [3] was used as an implicit prior model. However, we have found that more advanced, domain-specific denoisers such those used in [13] can yield better results.

Figure 2 shows the results for the case of tomographic reconstruction using 8 views of a 128×128 phantom. The algorithm was implemented using the SVMBIR tomographic software package [20]. Figures 2(a) to (d) show a typical progression of samples for the GPnP algorithm as σ decreases. For large values of σ, the prior is essentially white noise, so the reconstructed image has similar attributes. As σ decreases, the image stabilizes to a less noisy image, but each trial produces a somewhat different result that represents the variation in the posterior distribution. Figures 2(f) and (g) show the mean and standard deviation over 10 trails. Notice that Figure 2(g) shows that most of the variation occurs near edges, which is what one might expect.

Figure 3 shows similar results for sparse interpolation from 10% of the pixels from a color 256×256 ground-truth image. This gives results qualitatively similar to the tomography case, with most of the variation occurring along image edges.

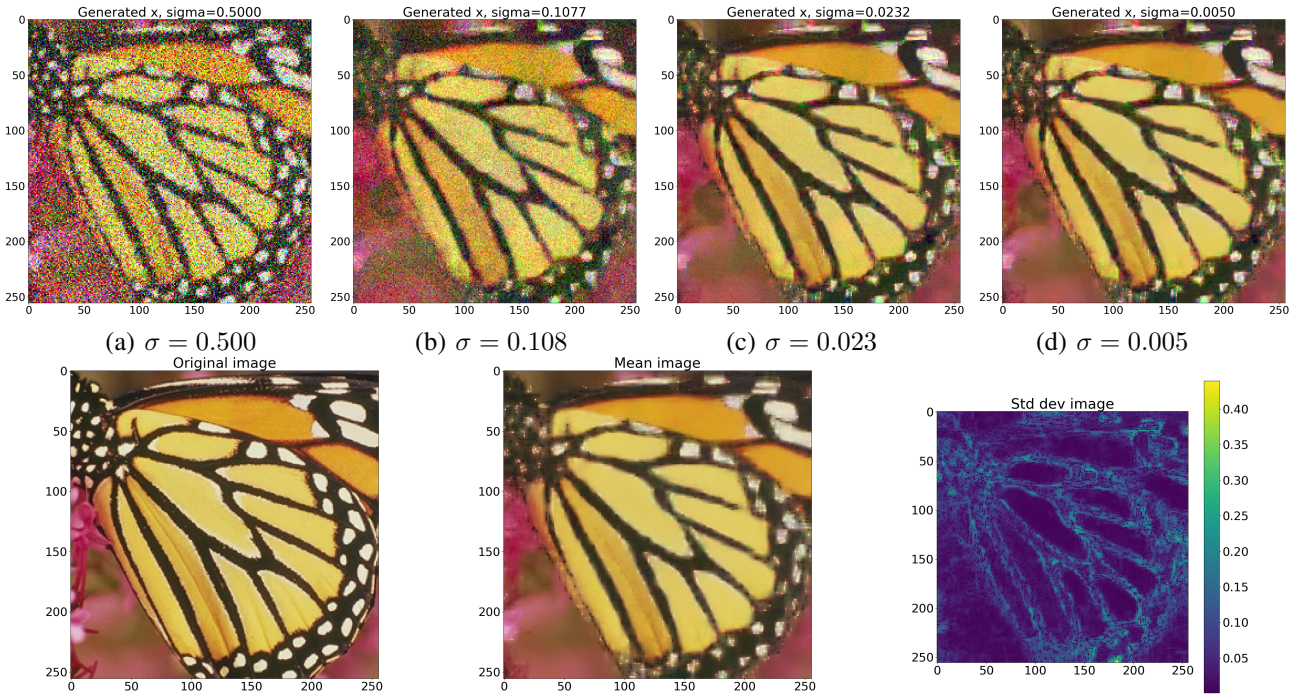
VI. Conclusion

In this paper, we presented a novel theory for Generative PnP, a generalization of the PnP that allows for sampling from the posterior distribution given a forward model and a prior specified using a MMSE denoising algorithm. As with PnP, GPnP has a modular implementation in which two proximal generators are alternately applied. The proximal generators generate conditionally independent random variables from a distribution inspired by the proximal map and in practice can be implemented by adding noise to the conventional proximal map.

Our key theoretical result is that the sequence generated by GPnP forms a reversible Markov chain with the desired posterior distribution. Our experimental results indicate that the algorithm can be robustly implemented for simple inverse problems such as sparse interpolation and 2D parallel beam tomographic reconstruction.

Appendix

We first prove the following lemma



(a) $\sigma = 0.500$ (b) $\sigma = 0.108$ (c) $\sigma = 0.023$ (d) $\sigma = 0.005$

(e) Ground Truth (f) Mean over 10 samples (g) Std Dev over 10 samples

Fig. 3. Interpolation experiment for 256×256 RGB image with 10% of pixels sampled uniformly at random, starting from radial basis function interpolation plus AWGN with $\sigma = 0.5$: (a) - (d) GPnP outputs with decreasing values of σ; (e) original image; (f) mean and (g) standard deviation over 10 trials.

Lemma 1. *Let* $[X_{n,0}, X_{n,1}]$ *be a Markov chain such that*

$$V_n = F_0(X_{n-1,1})$$
$$[X_{n,0}, X_{n,1}] = [V_n, F_1(V_n)] \ .$$

Then $[X_{n,0}, X_{n,1}]$ *is a Markov chain in time,* n*, with a stationary distribution given by*

$$[X_{n,0}, X_{n,1}] \sim p(x_0, x_1)$$

where

$$p(x_0, x_1) = \frac{1}{Z} \exp\left\{-u_1(x_1) - u_0(x_0) - \frac{1}{2\gamma^2}\|x_0 - x_1\|^2\right\} \ .$$

Proof of Lemma 1. Note that the distribution $p(x_0, x_1)$ has conditional distributions

$$\begin{aligned} p_{0|1}(x_0|x_1) &= q_0(x_0|x_1) \\ p_{1|0}(x_1|x_0) &= q_1(x_1|x_0) \ . \end{aligned}$$

Hence the Markov chain is an implementation of a Gibbs sampler that first replaces $X_{k,0}$ with a conditionally independent random variable from its conditional distribution, and then replaces $X_{k,1}$ with a conditionally independent random variable from its conditional distribution [21], [22].

For notational compactness, we denote the state at time $n-1$ by (x_0, x_1), and the state at time n by (x_0', x_1'), and let $q(x_0', x_1')$ denote the distribution of the state at time n. Defining $p_1(x_1) = \int p(x_0, x_1) dx_0$, we have $p(x_0, x_1) = q_0(x_0|x_1)p_1(x_1)$. Then standard manipulations give

$$\begin{aligned} q(x_0',x_1') &= \int\int q_1(x_1'|x_0')q_0(x_0'|x_1)p(x_0,x_1)dx_0dx_1 \\ &= \int\int q_1(x_1'|x_0')q_0(x_0'|x_1)q_0(x_0|x_1)p_1(x_1)dx_0dx_1 \\ &= \int q_1(x_1'|x_0')q_0(x_0'|x_1)\int q_0(x_0|x_1)dx_0\, p_1(x_1)dx_1 \\ &= \int q_1(x_1'|x_0')q_0(x_0'|x_1)p_1(x_1)dx_1 \\ &= \int q_1(x_1'|x_0')p(x_0',x_1)dx_1. \end{aligned}$$

Since $q_1(x_1'|x_0')$ does not depend on x_1, while $p(x_0', x_1)dx_1$ integrates to $p_0(x_0')$, this simplifies to give

$$q_1(x_1'|x_0')p_0(x_0') = q(x_0', x_1') = p(x_0', x_1').$$

Hence $p(x_0, x_1)$ is a stationary distribution of the Markov chain. □

Proof of Theorem 1. Recall that $X_n = F_1(F_0(X_{n-1})$ from Theorem 1, so that X_n is a Markov chain that equals $X_{n,1}$ in Lemma 1. By Lemma 1 we know that $X_{n,1}$ has a stationary

distribution given by

$$
\begin{aligned}
p(x_1) &= \int p(x_0, x_1) dx_0 \\
&= \int \frac{1}{Z} \exp\left\{-u_1(x_1) - u_0(x_0) - \frac{1}{2\gamma^2}\|x_0 - x_1\|^2\right\} dx_0 \\
&= \frac{1}{Z} \exp\left\{-u_1(x_1)\right\} \cdot \\
&\qquad \int \exp\left\{-u_0(x_0) - \frac{1}{2\gamma^2}\|x_0 - x_1\|^2\right\} dx_0 \ .
\end{aligned}
$$

Then notice that

$$
\begin{aligned}
&\int \exp\left\{-u_0(x_0) - \frac{1}{2\gamma^2}\|x_0 - x_1\|^2\right\} dx_0 \\
&\quad = \int e^{-u_0(x_0)} \exp\left\{-\frac{1}{2\gamma^2}\|x_0 - x_1\|^2\right\} dx_0 \\
&\quad = \left(e^{-u_0(\cdot)} * \exp\left\{-\frac{1}{2\gamma^2}\|\cdot\|^2\right\}\right)(x_1) \\
&\quad = (2\pi\gamma^2)^{p/2} \left(e^{-u_0(\cdot)} * g_{\gamma^2}\right)(x_1) \\
&\quad = (2\pi\gamma^2)^{p/2} \exp\left\{-\tilde{u}_0(x_1)\right\} \ ,
\end{aligned}
$$

where

$$
\tilde{u}_0 = -\log\left(e^{-u_0} * g_{\gamma^2}\right) \ .
$$

So we have that

$$
\begin{aligned}
p(x_1) &= \frac{1}{Z'} \exp\left\{-u_1(x_1)\right\} \exp\left\{-\tilde{u}_0(x_0)\right\} \\
&= \frac{1}{Z'} \exp\left\{-u_1(x_1) - \tilde{u}_0(x_0)\right\} \ ,
\end{aligned}
$$

where $Z' = Z/(2\pi\gamma^2)^{p/2}$.

To show reversibility, we denote the joint distribution of $(X_{n,1}, X_{n-1,1})$ as $q(x_1', x_1)$. As in Lemma 1, we define $p_0(x_0) = \int p(x_0, x_1) dx_1$ and note that $p(x_0, x_1) = q_1(x_1|x_0)p_0(x_0)$. Then we have

$$
\begin{aligned}
q(x_1', x_1) &= \int\int q_1(x_1'|x_0') q_0(x_0'|x_0) p(x_0, x_1) dx_0 dx_0' \\
&= \int\int q_1(x_1'|x_0') q_0(x_0'|x_1) q_0(x_0|x_1) p_1(x_1) dx_0 dx_0' \\
&= \int q_1(x_1'|x_0') q_0(x_0'|x_1) \int q_0(x_0|x_1) dx_0 \, p_1(x_1) dx_0' \\
&= \int q_1(x_1'|x_0') q_0(x_0'|x_1) p_1(x_1) dx_0' \\
&= \int q_1(x_1'|x_0') p(x_0', x_1) dx_0' \\
&= \int q_1(x_1'|x_0') q_1(x_1|x_0') p_0(x_0') dx_0' \ .
\end{aligned}
$$

Since $q_1(x_1'|x_0')q_1(x_1|x_0')$ is symmetric in x_1 and x_1', this implies that

$$
q(x_1', x_1) = q(x_1, x_1') \ ,
$$

which means that the Markov chain $X_{n,1}$ is reversible. □

REFERENCES

[1] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2013*. IEEE, 2013, pp. 945–948.

[2] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 408–423, Dec 2016.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[4] C. J. Pellizzari, T. J. Bate, K. P. Donnelly, G. T. Buzzard, C. A. Bouman, and M. F. Spencer, "Coherent plug-and-play artifact removal: Physics-based deep learning for imaging through aberrations," *Optics and Lasers in Engineering*, vol. 164, 2023.

[5] S. Majee, T. Balke, C. A. J. Kemp, G. T. Buzzard, and C. A. Bouman, "Multi-slice fusion for sparse-view and limited-angle 4d ct reconstruction," *IEEE Transactions on Computational Imaging*, vol. 7, 2021.

[6] G. T. Buzzard and C. A. B. Stanley H. Chan, Suhas Sreehari, "Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium," *SIAM Journal on Imaging Sciences*, vol. 11, no. 3, pp. 2001–2020, 2018.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS*, vol. 27, 2014.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[9] M. Mirza and S. Osindero, "Conditional generative adversarial nets," in *International Conference on Learning Representations (ICLR)*, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784

[10] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[11] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 11 895–11 907.

[12] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Computation*, vol. 23, no. 7, p. 1661–1674, 2011.

[13] Y. Song and S. Ermon, "Improved techniques for training score-based generative models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[14] U. Grenander and M. Miller, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *Journal of the Royal Statistical Society B*, vol. 56, no. 4, pp. 549–581, 1994.

[15] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Robust compressed sensing MRI with deep generative priors," in *International Conference on Learning Representations (ICLR)*, 2021.

[16] B. T. Feng, J. Smith, M. Rubinstein, H. Chang, K. L. Bouman, and W. T. Freeman, "Score-based diffusion models as principled priors for inverse imaging," 2017.

[17] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. I. Tamir, "Robust compressed sensing MRI with deep generative priors," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[18] Y. Song, L. Shen, L. Xing, and S. Ermon, "Solving inverse problems in medical imaging with score-based generative models," in *International Conference on Learning Representations (ICLR)*, 2022.

[19] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in *International Conference on Learning Representations (ICLR)*, 2023.

[20] S. D. Team, "Super-Voxel Model Based Iterative Reconstruction (SVMBIR)," Software library available from https://github.com/cabouman/svmbir, 2020.

[21] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, 1984.

[22] C. A. Bouman, *Foundations of Computational Imaging: A Model Based Approach*. Philadelphia: Society for Industrial and Applied Mathematics, 2022.