



école  
normale  
supérieure  
paris-saclay

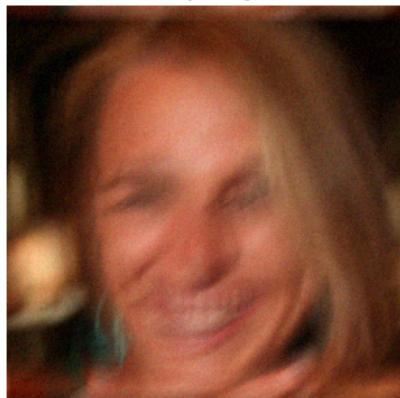
# Plug-and-Play Split Gibbs Sampler

Generative Modeling

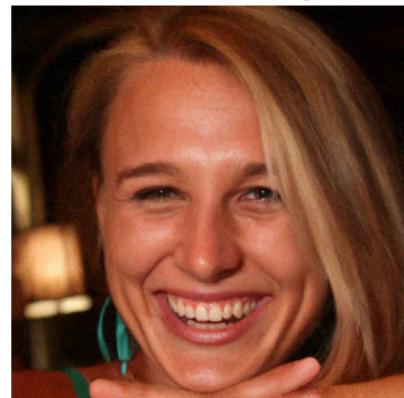
March 2025

Ibrahim Ridene & Martin Jolif

Noisy image



Z Reconstructed image



## 1 Introduction

We worked on the **Plug-and-Play split Gibbs sampler** (PnP-SGS) project. To do so, we studied the following paper: "Plug-and-Play split Gibbs sampler: embedding deep generative priors in Bayesian inference" [2]. In this paper, the authors present a stochastic plug-and-play sampling algorithm that utilizes variable splitting to efficiently sample from a posterior distribution. We studied this algorithm theoretically through the paper and developed an implementation of this algorithm to solve inverse imaging problems like deblurring and inpainting.

The proposed algorithm (PnP-SGS) simplifies the difficult task of posterior sampling by splitting it into two easier subproblems. The first one relates to the likelihood function, while the second one can be viewed as a Bayesian denoising problem, which is solved using a deep generative model. In our case, this deep generative model will be a Denoising Diffusion Probabilistic Model (DDPM) [3]. In contrary to deterministic PnP algorithm, the proposed method has the significant advantage of not requiring an explicit prior distribution, as it is instead encoded in a pretrained generative model. Unlike optimization methods like PnP-ADMM, which typically provide only point estimates, this approach allows for conventional Bayesian estimators to be paired with confidence intervals, only with a reasonable increase in computational cost.

## 2 Inverse Problems formulation

The goal of some inverse problems is to infer an unknown object of interest  $x \in \mathbb{R}^N$  from a partial and noisy observation  $y \in \mathbb{R}^M$ . This problem can be formulated as follows:

$$\min_x f(x, y) + g(x) \quad (1)$$

where  $f(x, y) = -\log p(y|x)$  represents the data-fitting term and  $g(x) \propto -\log p(x)$  acts as a regularization term. However, solving (1) is difficult, due to high-dimensional data for example.

Therefore, to simplify the optimization problem, some variable splitting methods such as Half-Quadratic-Splitting (HQS) or Alternating Direction Method of Multipliers (ADMM) [1] introduce an auxiliary variable  $z$  to decouple the optimization of  $f$  and  $g$ . Solving (1) is equivalent to solve the following minimization problem, thanks to the equality constraint:

$$\min_{\substack{x, z \\ x=z}} f(x, y) + g(z)$$

By adopting an alternate minimization strategy, the subproblem with respect to  $z$  can now be solved by using the proximal operator of the regularization term  $g$ , which can be interpreted as a denoiser [4]. Recent Plug-and-Play (PnP) approaches replaced this proximal operator by a deep-neural-network denoiser and reach state of the art performance on several applications.

However, PnP algorithms typically produces only one estimation. Moreover, they do not quantify uncertainties and do not give any information on the posterior distribution:

$$\pi(x) = p(x|y) \propto \exp[-f(x, y) - g(x)] \quad (2)$$

## 3 Plug-and-Play split Gibbs sampler algorithm

Starting from the target posterior distribution (2), the introduction of a splitting variable  $z \in \mathbb{R}^N$  results in the following augmented distribution:

$$\pi_\rho(x, z) = p(x, z|y; \rho^2) \propto \exp[-f(x, y) - g(z) - \frac{1}{2\rho^2} \|x - z\|^2] \quad (3)$$

where  $\rho$  is a positive coupling parameter. As established in the literature,  $\|\pi - \pi_\rho\|_{TV} \xrightarrow[\rho^2 \rightarrow 0]{} 0$ . It indicates that the original target distribution  $\pi$  is recovered from the marginal distribution  $\pi_\rho$  when  $\rho \rightarrow 0$ . This makes this technique an exact data augmentation method. SGS proposes to sample according to the augmented data distribution  $\pi_\rho$  using Gibbs steps:

$$p(x|z, y; \rho^2) \propto \exp[-f(x, y) - \frac{1}{2\rho^2}\|x - z\|^2] \quad (4)$$

$$p(z|x; \rho^2) \propto \exp[-g(z) - \frac{1}{2\rho^2}\|x - z\|^2] \quad (5)$$

The sampling process from equation (4), can be interpreted as addressing the initial inverse problem with a gaussian prior:  $\mathcal{N}(z, \rho^2 I)$ . As it relies on a standard gaussian prior, sampling from this modified posterior is simpler than sampling from the the original posterior (2).

The conditional distribution (5) can be interpreted as the posterior distribution associated with a Bayesian denoising problem. Indeed, the goal is to reconstruct  $z$  from a noisy observation  $x$  corrupted with gaussian noise. Instead of directly sample from the conditional distribution (5), the authors of the paper propose to use a deep generative model as stochastic denoiser like GAN, VAE or a diffusion model.

The proposed PnP-SGS method alternatively samples from the conditional posterior distributions from equations (4) and (5). Throughout this iterative process, SGS generates a set of  $N_{MC}$  samples  $\{x^{(n)}, z^{(n)}\}_{1 \leq n \leq N_{MC}}$  which asymptotically follow the augmented posterior  $\pi_\rho(x, z)$ . From this sample set, various Bayesian quantities can be estimated, such as Bayesian estimators and confidence intervals.

Regarding the first step of SGS, sampling according to (4) is problem dependent and should be suitably adapted to the targeted task. In the second step of SGS, at the  $n$ -th iteration, sampling according to equation (5) can be understood as a stochastic denoising of the current value  $x^{(n)}$ . As said before, this sampling is carried out in a Plug-and-Play (PnP) fashion, leveraging a previously pretrained model like a diffusion model.

## 4 Denoising Diffusion Probabilistic Model (DDPM)

### 4.1 Diffusion Process

Intuitively, at each step during the diffusion process, we will add noise to the original sample  $u_0$ . The proposed diffusion process is the following:

$$p(u_t|u_{t-1}) = \mathcal{N}(u_t; \sqrt{1 - \beta_t}u_{t-1}, \beta_t I_d)$$

where the variance  $\beta_t$  schedule can be chosen for example as a linearly increasing function. As the diffusion process corresponds to a Markov chain, we obtain:

$$p(u_t|u_0) = \mathcal{N}(u_t; \sqrt{\bar{\alpha}_t}u_0, (1 - \bar{\alpha}_t)I_d) \quad (6)$$

where  $\alpha_t = 1 - \beta_t$  and  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ .

### 4.2 Denoising process

The goal of the denoising process is to reverse the noising process and to be able to reconstruct a sample given its noisy version. To do so, we use the pretrained parameters  $\theta$ :

$$q_\theta(u_{t-1}|u_t) = \mathcal{N}(u_{t-1}; \mu_\theta(u_t, t), \Sigma_\theta(u_t, t)) \quad (7)$$

$$\text{where, } \mu_\theta(u_t, t) = \frac{1}{\sqrt{\alpha_t}} \left( u_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(u_t, t) \right) \text{ and } \Sigma_\theta(u_t, t) = \frac{(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \beta_t I$$

$\epsilon_\theta$  is a function predicting the noise  $\varepsilon$  added to  $u_{t-1}$ . In our case, with an image dataset, the authors designed a U-Net architecture with a transformer sinusoidal positional embedding to process time as well as the input image.

Therefore, a pretrained DDPM can be used as a stochastic gaussian denoiser. Unlike the typical use of a DDPM as a generator, the main idea here is to initiate the reverse diffusion process from a noisy image  $u_{\hat{t}^*}$  at a specific time  $\hat{t}^*$ , rather than from a noise sample  $u_T$  as it is usually done. The noise-free image  $u_0$  can then be recovered by applying the reverse process, as defined in equation (7), starting from time  $\hat{t}^*$  until time  $t = 1$ .

## 5 DDPM-based Plug-and-Play split Gibbs sampler algorithm

Note that DDPMs used as generators are known to be generally computationally demanding due to the number  $T$  of overall steps involved in the backward process. The proposed approach overcome this issue by initiating the process from a generally weakly noisy image, which significantly reduces the necessary number  $t^* \ll T$  of denoising steps to be applied.

To compute  $t^*$ , the authors propose to capitalize on the mapping between  $t^*$  and the variance  $1 - \bar{\alpha}_{t^*}$  of the noise from  $u_{\hat{t}^*}$  (equation (6)). Therefore, the variance of the noise in the image is estimated using the `estimate_sigma()` function from the `scikit-image` package:  $\hat{\sigma} = \Phi(x^{(n)})$ . By previous definitions, we obtain an estimation of  $t^*$ :  $\hat{t}^* = (1 - \bar{\alpha})^{-1}(\hat{\sigma}^2)$ .

---

**Algorithm 1** PnP-SGS using DDPM

---

**Input:** Parameter  $\rho^2$ , total number of iterations  $N_{MC}$ , number of burn-in iterations  $N_{bi}$ , pre-trained DDPM  $\epsilon_\theta(\cdot, \cdot)$ , scheduling variance function  $\alpha(\cdot)$ , initialization  $z^{(1)}$ , observation  $y$ .

```

1: for  $n = 1$  to  $N_{MC}$  do
2:   Draw  $x^{(n)} \sim p(x | z^{(n)}, y; \rho^2)$             $\triangleright$  Sampling the variable of interest  $x^{(n)}$  according to (4)
3:   Set  $\hat{\sigma} = \Phi(x^{(n)})$                           $\triangleright$  Estimating noise level in  $x^{(n)}$ 
4:   Set  $\hat{t}^* = (1 - \bar{\alpha})^{-1}(\hat{\sigma}^2)$            $\triangleright$  Setting the number of diffusion steps to denoise  $x^{(n)}$ 
5:   Set  $u_{\hat{t}^*} = x^{(n)}$ 
6:   for  $j = \hat{t}^*$  downto 1 do
7:     Draw  $u_{j-1} \sim q_\theta(u_{j-1} | u_j)$             $\triangleright$  Sampling the splitting variable  $z^{(n)}$  according to (7)
8:   end for
9:   Set  $z^{(n)} = u_0$ 
10: end for
11: Output: Collection of samples  $\{x^{(n)}, z^{(n)}\}_{n=N_{bi}+1}^{N_{MC}}$  asymptotically distributed.

```

---

## 6 Application to inverse problems in imaging

We model an imaging inverse problems as follows:

$$y = H^T x + n \tag{8}$$

where  $H$  is a linear operator,  $y$  the observation,  $x$  the target image we want to infer, and  $n \sim \mathcal{N}(0, \Omega^{-1})$  a gaussian noise. Additionally, we can also assume that  $\Omega^{-1} = \sigma^2 I$ . Through simple operations on gaussian distributions, we get that:

$$p(y|x) = \mathcal{N}(y; H^T x, \Omega^{-1}) \tag{9}$$

## 6.1 Image deblurring

In this context, we have  $x \in \mathbb{R}^N$ ,  $y \in \mathbb{R}^N$  and  $H \in \mathbb{R}^{N \times N}$  is a circulant convolution matrix associated to a blurring kernel. After some computations and thanks to equation 9, we obtain:

$$p(x|z, y; \rho^2) = \mathcal{N}(x; \mu_x, Q_x^{-1}) \quad (10)$$

$$\mu_x = Q_x^{-1}(H\Omega y + \frac{1}{\rho^2}z) \text{ and } Q_x = H\Omega H^T + \frac{1}{\rho^2}I_N$$

In our experiments, we used RGB images of size (256, 256) and  $N$  should be equal to  $256 * 256 * 3$ . Therefore, to avoid memory issues with saving a big matrix  $H$ , we just stored the kernel of size (61, 61) for example and use Fourier Transform to sample from equation (10).

## 6.2 Image inpainting

In this context, we have  $x \in \mathbb{R}^N$ ,  $y \in \mathbb{R}^M$  and  $H \in \{0, 1\}^{N \times M}$  is a binary mask matrix, where  $M \ll N$  is the number of visible pixels. We get the same results as for image deblurring; equation (10). Moreover, under the assumption that  $\Omega^{-1} = \sigma^2 I_M$ , and knowing that the operator  $H^T$  consists, in this case, of a subset of rows of the identity matrix  $I_N$ , we observe that  $H^T H = I_M$  and the Sherman-Morrison-Woodbury formula yields:

$$Q_x^{-1} = \rho^2(I_N - \frac{\rho^2}{\sigma^2 + \rho^2} H H^T)$$

## 6.3 Image Super-Resolution

In the image super-resolution context, we assume that  $\Omega^{-1} = I$ ,  $y$  is a low-resolution observation,  $x$  a high-resolution target and the forward operator is defined as:

$$H = SB \quad (11)$$

where  $B$  is a circulant convolution matrix corresponding to a **blurring kernel** and  $S$  is a **binary mask** associated with the downsampling process. To efficiently sample from the posterior distribution, the problem is reformulated using two auxiliary variables,  $z_1$  and  $z_2$ , leading to the following augmented posterior:

$$p(x, z_1, z_2 | y; \rho_1^2, \rho_2^2) \propto \exp \left( -\frac{1}{2\sigma^2} \|y - Sz_1\|^2 - \frac{1}{2\rho_1^2} \|z_1 - Bx\|^2 - g(z_2) - \frac{1}{2\rho_2^2} \|z_2 - z_1\|^2 \right) \quad (12)$$

This introduces a three-step sampling process:

1. **Sampling  $z_1$  given  $x$  and  $y$**  enforces data consistency by aligning  $z_1$  with the observed low-resolution image.
2. **Sampling  $x$  given  $z_1$  and  $z_2$**  the intermediate variable  $z_1$  helps estimate the high-resolution image while ensuring consistency with the blurred observations.
3. **Sampling  $z_2$  given  $z_1$** : A pretrained **DDPM denoiser** is used to refine the reconstruction, enforcing image priors.

## 7 Experiments & Results

In this section, we present various experiments conducted to evaluate the effectiveness of the Plug-and-Play Split Gibbs Sampler (PnP-SGS) on different inverse imaging tasks. The main objectives of these experiments are to assess the quality of image reconstruction under different degradation models and study the impact of various algorithmic parameters.

We investigate three primary inverse problems: **image deblurring**, **image inpainting**, and **image super-resolution**, each requiring a specific forward operator. Additionally, we analyze the influence of **the coupling parameter  $\rho$**  and the choice of **diffusion scheduler type** (linear vs. cosine) on the reconstruction quality.

We conduct experiments using the **FFHQ** dataset: it consists of high-quality facial images, which serve as the basis for evaluating the different image restoration tasks implemented in the framework. The structure of the code can also be directly adapted to various other datasets.

## 7.1 Image Deblurring

The forward operator corresponds to a **Gaussian or motion blur kernel**, applied to the original image, followed by additive Gaussian noise. Specifically, we consider two types of blur:

1. **Gaussian blur:** The kernel size is set to **61**, and the blur standard deviation is **3.0**.
2. **Motion blur:** The kernel size is set to **61**, with an intensity of **0.5**.

Both cases involve three-channel images with a resolution of  **$256 \times 256$** . The reconstruction process involves alternating between the likelihood-based sampling step and the diffusion prior step to recover a sharp image from the blurred observation. The quality of restoration is assessed using **PSNR** and **SSIM** metrics.



Figure 1: Image Deblurring Results for Gaussian Blur.

For **Gaussian blur** (Figure 1), the noisy image exhibits a uniform blur across the entire image, resulting in a loss of sharpness and fine details. The **PnP-SGS** reconstruction effectively restores most of the lost details, particularly in the facial features and background textures. However, some slight artifacts and residual noise are visible in the final reconstruction, which suggests that while the model successfully mitigates the blur, some high-frequency information may still be missing.



Figure 2: Image Deblurring Results for Motion Blur.

For **motion blur** (Figure 2), the degradation is more structured and directional, causing a smearing effect along a specific trajectory. Compared to the Gaussian blur case, the **PnP-SGS** reconstruction appears to recover the sharpness well, particularly in the facial features and contours. However, due to the nature of motion blur, some distortions remain, especially in

regions with high contrast. The model performs well in compensating for the blur but struggles with completely recovering fine details.

Overall, the results demonstrate the effectiveness of **PnP-SGS** in handling different types of blurs, with **better performance on Gaussian blur compared to motion blur**. The artifacts in the motion blur case suggest that additional refinement steps or a more specialized prior might be beneficial for further improvement.

## 7.2 Image Inpainting

We consider a **masking operator** where **80% of the pixels** are randomly removed, corresponding to a **mask ratio of 0.8**. The images used in this experiment have a resolution of **256 × 256**. The goal is to recover the missing regions while preserving the structure and texture of the image. The masked image serves as input to PnP-SGS, which iteratively refines the reconstruction using the diffusion prior. The effectiveness of the method is analyzed by comparing the reconstructed images against the ground truth using **PSNR** and **SSIM** metrics.



Figure 3: Image Inpainting Results.

The **Noisy Image** clearly shows significant pixel removal, making direct human interpretation difficult. The **Z Reconstructed Image** provides an initial estimation, but noticeable artifacts, especially in complex regions like hair, remain. The **X Reconstructed Image** further refines the details, successfully recovering most of the missing structure while reducing the noise.

However, some inconsistencies remain in high-detail areas, such as hair strands, where artifacts and unnatural patterns appear. This suggests that while the model effectively recovers smooth textures and general structure, finer details in highly textured regions still pose a challenge. Further improvements might be achieved by incorporating more advanced priors or increasing the number of sampling iterations.

## 7.3 Image Super-Resolution

The degradation model consists of a **blurring kernel** followed by a **subsampling operation**, which reduces the spatial resolution of the input image. The subsampling factor used is **equal to 4**. The reconstruction process involves iteratively refining the solution using the learned diffusion prior while incorporating the degradation model constraints.

Figure 4 presents the results of the super-resolution experiment. The reconstructed images demonstrate a clear improvement in sharpness and detail preservation compared to the input low-resolution image. However, some noise artifacts can be observed, especially in regions with fine textures.

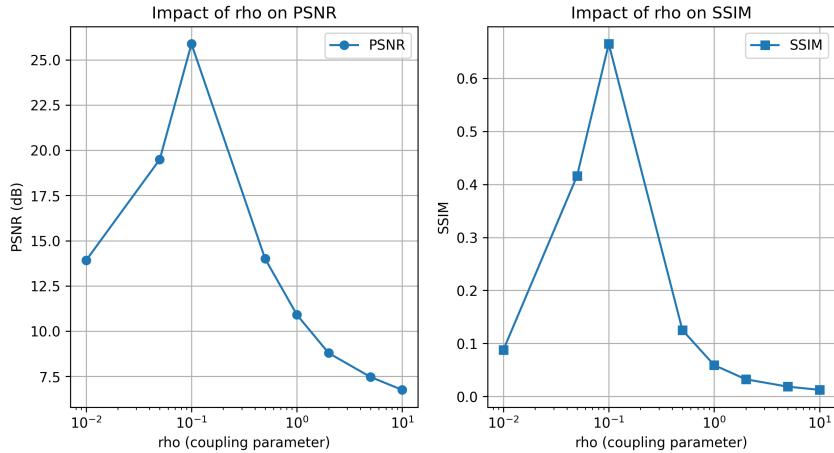
The results suggest that while the PnP-SGS framework effectively enhances the image resolution, the quality of the restoration is influenced by the complexity of the underlying degradation process. Future improvements could involve incorporating adaptive upsampling strategies or fine-tuning the diffusion model for better texture reconstruction.



Figure 4: Results of the super-resolution experiment.

#### 7.4 Impact of Coupling Parameter

The coupling parameter  $\rho$  plays a crucial role in balancing the likelihood term and the diffusion prior in the PnP-SGS framework. We conduct experiments by varying  $\rho$  over a range of values and evaluate its impact on reconstruction quality.

Figure 5: Impact of Coupling Parameter  $\rho$  on Reconstruction Quality for **motion blur kernel**.

To study its effect, we evaluate reconstruction quality across different values of  $\rho$  for the **motion blur kernel**, measuring Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). From Figure 5, we observe a characteristic trend:

- For **small**  $\rho$ , the term  $\frac{1}{\rho^2} \|x - z\|^2$  becomes dominant, enforcing a strong coupling between  $x$  and  $z$ . This means the reconstructed image remains close to the noisy input, preventing the diffusion prior from properly guiding the reconstruction. As a result, the recovered image retains much of the degradation, leading to poor PSNR and SSIM.
- As  $\rho$  increases, the influence of the prior gradually strengthens, improving the reconstruction. The highest PSNR and SSIM values are achieved around  $\rho \approx 0.1$ , where an optimal trade-off between the likelihood and prior is established.
- However, for **large**  $\rho$ , the coupling term  $\frac{1}{\rho^2} \|x - z\|^2$  diminishes, weakening the likelihood constraint. In this case, the reconstruction becomes overly dependent on the learned prior, which may not align perfectly with the corrupted observation. This results in an oversmoothed image, leading to a sharp decline in PSNR and SSIM.

This study highlights the importance of tuning  $\rho$  appropriately. Choosing a small  $\rho$  makes the reconstruction overly faithful to the degraded observation, limiting the benefits of the prior, while a large  $\rho$  results in reconstructions that may deviate from the actual structure of the

observed data. The optimal  $\rho$  balances these effects to yield the best possible image quality. Same results are found for the other kernels, you can refer to the appendix for that in Section 9.1.

## 7.5 Impact of Scheduler Type

We compare the performance of two different scheduling strategies for the diffusion model: **linear and cosine schedules**.

In the **linear schedule**, the variance is increased linearly over time  $\beta(t) = \beta(0) + rt$  where  $\beta(0) = 10^{-4}$  and the slope  $r$  is chosen such that  $\beta(T) = 2.0 \times 10^{-2}$ . The function  $\beta(t)$  controls the noise variance of the forward transition kernel.

In contrast, the **cosine schedule** follows a squared cosine function which is a smoothly decreasing function that governs the noise level, with  $s$  being a small offset to prevent singularities:

$$\bar{\alpha}(t) = 1 - \frac{\gamma(t)}{\gamma(0)} \text{ where } \gamma(t) = \cos^2 \left( \frac{\pi}{2} \frac{t/T + s}{1+s} \right) \quad (13)$$

By **analyzing the estimated values of  $t^*$**  across PnP-SGS iterations, we compare the noise removal efficiency and the influence of the scheduler type on the final reconstruction. Figure 6 illustrates the evolution of  $t^*$  over the iterations for both scheduling strategies. The results indicate that the cosine scheduler maintains higher values of  $t^*$  at the beginning, leading to a more stable denoising process, while the linear scheduler decreases  $t^*$  more aggressively. This difference may influence the overall reconstruction quality, depending on the nature of the degradation.

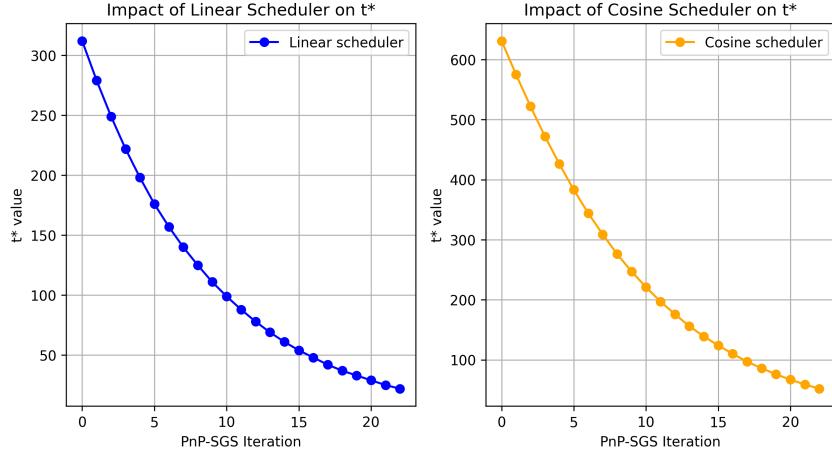
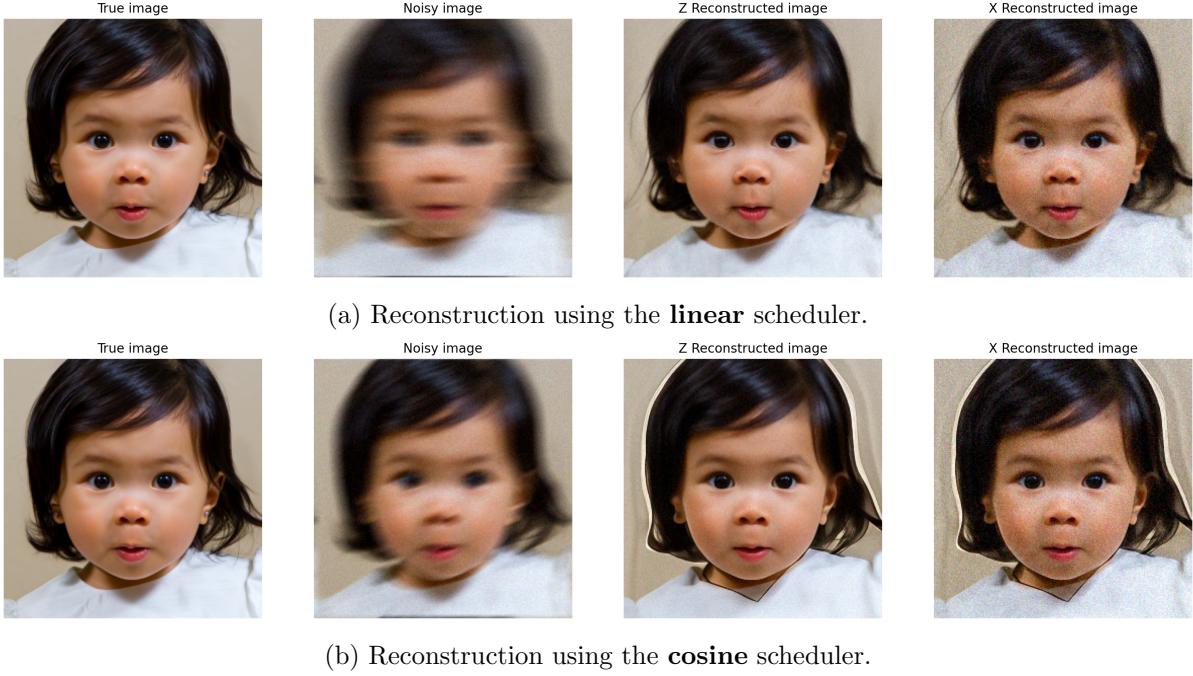


Figure 6: Comparison of the impact of the **linear** and **cosine** schedulers on  $t^*$  values over iterations. The cosine scheduler starts with a higher  $t^*$  and decreases more smoothly, while the linear scheduler follows a steady decay.

To further validate this observation, we analyze the reconstructed images obtained using each scheduler. Figure 7 compares the reconstruction quality on a face degraded by motion blur for both schedulers.

The results show that the **cosine scheduler** produces a smoother reconstruction, reducing noise but introducing unwanted artifacts around the edges, particularly around the contours of the face. On the other hand, the **linear scheduler** retains finer details but at the cost of increased noise. This aligns with the previous curve analysis: the linear scheduler decreases  $t^*$  more aggressively, leading to sharper but noisier reconstructions, while the cosine scheduler maintains a higher  $t^*$  initially, resulting in excessive smoothing and structural inconsistencies.

Figure 7: Comparison of reconstructed images for the **linear** and **cosine** schedulers.

## 8 Conclusion

In this work, we explored the application of the **Plug-and-Play Split Gibbs Sampler (PnP-SGS)** for solving inverse problems in image restoration. We evaluated its effectiveness on three tasks: **image deblurring**, **image inpainting**, and **super-resolution**, demonstrating its ability to recover high-quality images from degraded observations. Additionally, we studied the impact of the **coupling parameter** and the choice of **diffusion scheduler**, analyzing their influence on reconstruction quality using PSNR and SSIM metrics. The results highlight the importance of carefully selecting these parameters to balance noise suppression and detail preservation. Future work may focus on optimizing parameter selection and extending the approach to more complex real-world degradations.

## 9 Appendix

### 9.1 Generalization of the Coupling Parameter Study

In addition to the results presented for the motion blur kernel, we conducted the same study on the **Gaussian blur** and **inpainting** operators. The obtained results exhibit the same trend: the performance in terms of PSNR and SSIM improves as  $\rho$  increases up to an optimal value around  $\rho \approx 0.1$ , after which it starts degrading. This confirms that the coupling parameter  $\rho$  plays a crucial role in the convergence of the PnP-SGS method across different inverse problems.

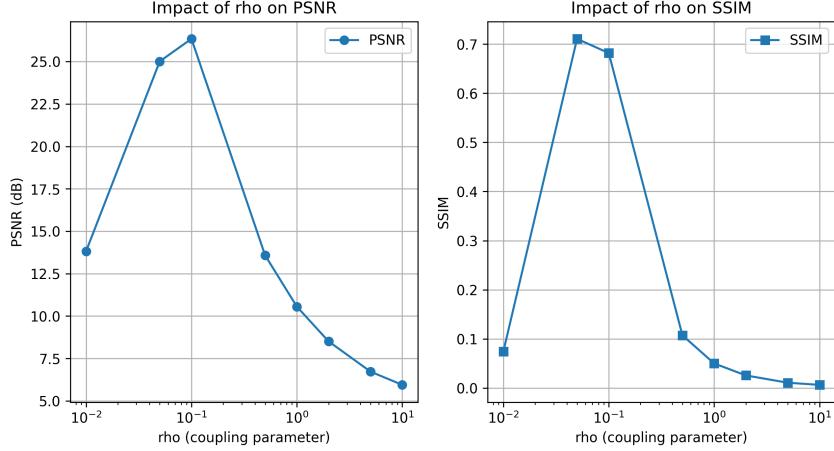


Figure 8: Impact of  $\rho$  on PSNR and SSIM for the Gaussian blur kernel.

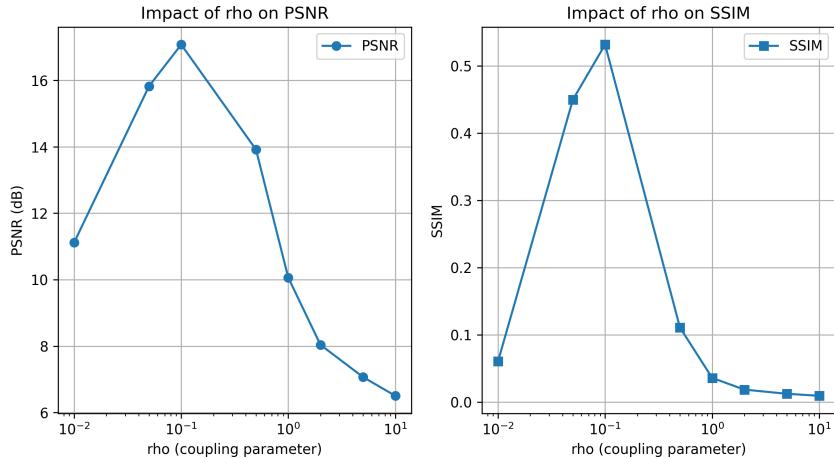


Figure 9: Impact of  $\rho$  on PSNR and SSIM for the inpainting operator.

## References

- [1] Stephen Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. In *Foundations and Trends in Machine Learning*, volume 3, pages 1–122, 2010.
- [2] Florentin Coeurdoux, Nicolas Dobigeon, and Pierre Chainais. Plug-and-play split gibbs sampler: embedding deep generative priors in bayesian inference, 2023.
- [3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.
- [4] Singanallur V. Venkatakrishnan, Charles A. Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948, 2013.