Projected Low-Rank Gradient in Diffusion-based Models for Inverse Problems

Rayhan Zirvi[1*] rayhanzirvi@caltech.edu
Bahareh Tolooshams[1*] btoloosh@caltech.edu
Anima Anandkumar[1] anima@caltech.edu

Abstract

Recent advancements in diffusion models have demonstrated their potential as powerful learned data priors for solving inverse problems. A popular Bayesian approach leverages diffusion sampling steps for inducing a data prior, generating images from noise while incorporating measurement gradient updates at each step to impose data consistency. However, diffusion models exhibit high sensitivity to measurement gradient step size and face challenges in preserving the process on the manifold, leading to performance degradation and artifact introduction in the sampled posterior. We propose a *Projected Low-Rank Gradient* (PLoRG) method, approximating the data manifold structure to enhance the performance and robustness of diffusion models in solving inverse problems. Our approach leverages singular value decomposition to approximate the measurement gradient in a lower-rank subspace defined by the current state, effectively preserving the manifold structure and filtering out artifact-inducing components. In addition to superior robustness, we show that PLoRG improves the performance of diffusion models on a range of linear and nonlinear inverse problems, especially those that are inherently challenging such as phase retrieval.

1 Introduction

Inverse problems are ubiquitous in science and engineering, playing a crucial role in simulation-based scientific discovery and real-world applications [1–3]. They arise in fields such as medical imaging [4–8], remote sensing [9, 10], astrophysics [11], molecular dynamics simulations [12, 13], and solving partial differential equations (PDEs) [14]. Inverse problems are also relevant in autonomous vehicles, where visual perception systems must contend with noisy or motion-blurred visual scenes [15–17].

Inverse problems aim to recover an unknown signal $\boldsymbol{x}^\star \in \mathbb{R}^n$ from noisy observations $\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}^\star) + \boldsymbol{n} \in \mathbb{R}^m$, where $\mathcal{A}$ denotes the measurement operator, and $\boldsymbol{n}$ is the noise. These problems are generally ill-posed, i.e., in the absence of a structure governing the underlying desired signal $\boldsymbol{x}$, many solutions can explain the measurements $\boldsymbol{y}$. In the Bayesian framework, this structure is translated into a *prior distribution* $p(\boldsymbol{x})$, which can be combined with the likelihood term $p(\boldsymbol{y}|\boldsymbol{x})$ to define a *posterior distribution* $p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$. Hence, solving the inverse problem translates into performing a Maximum a Posteriori (MAP) estimation or drawing high-probability samples from the posterior [18]. Given the forward model, the critical step is to define the prior $p(\boldsymbol{x})$.

Prior works consider sparse priors and provide a theoretical analysis of conditions for the unique recovery of data where $\mathcal{A}$ is a linear operator, a problem known as compressed sensing [19, 20]. Sparse priors have shown usefulness in medical fields such as computed tomography (CT) [21] and magnetic resonance imaging (MRI) [22], and engineering applications such as radar imaging [23]. This approach is categorized into model-based priors where a structure is assumed on the signal, e.g., exploiting the low-rank structure of data [24], as opposed to being learned [25].

[*]Equal contribution. [1]Computing and Mathematical Sciences, California Institute of Technology.

D3S3: Data-driven and Differentiable Simulations, Surrogates, and Solvers @ NeurIPS 2024.

Recent literature goes beyond such model-based priors and leverages data structures captured by a pre-trained deep generative model [26, 27] to solve inverse problems [28–31]. The latest works employ generative diffusion models [31–34], which implicitly learn the data prior $p(\boldsymbol{x})$ by learning a process that transforms noise into structured data. For inverse problems, this reverse generation process is guided by a data-consistency term $p(\boldsymbol{y}|\boldsymbol{x})$. This approach has shown promising performance in generating high-quality samples [35–37], and solving PDEs [38, 39].

Despite their promise, diffusion models are still facing challenges solving inverse problems, limiting their widespread applications. At a high level, the main challenge concerns the preservation of the process on the data manifold; the measurement consistency gradient step gradually pushes away the process from this manifold, leading to the introduction of "artifacts" in the reconstructed data [40]. Prior work [41, 42] aims to address this challenge by decoupling the sampling from the measurement process. However, challenges remain when a) latent diffusion models (LDMs) [43], as opposed to pixel-based, are employed due to the nonlinearity of their latent-to-pixel decoder, and b) the measurement consistency gradient step size is not delicately tuned for desired performance.

We propose a *Projected Low-Rank Gradient* (PLoRG)[2] step that can be utilized by a wide range of diffusion-based methods to enhance their performance and robustness in solving inverse problems. PLoRG filters out artifact-inducing components by projecting the measurement gradient (MG) onto a lower-rank subspace enforced by the sample at the current step of the diffusion. PLoRG helps the samples to remain on or close to the manifold, hence improving reconstruction quality (Figure 1). We demonstrate that PLoRG significantly improves the robustness of diffusion-based inverse methods to the data-consistency gradient step size; this is particularly crucial for enabling the wider real-world application of diffusion models for solving inverse problems. We apply PLoRG to both linear and nonlinear inverse problems for image restoration tasks; our results show that PLoRG can significantly improve state-of-the-art (SOTA) performance, particularly in challenging inverse problem tasks. For example, PLoRG drastically improves the box inpainting (PSNR $19.91 \rightarrow 23.59$ dB) and phase retrieval (PSNR $27.61 \rightarrow 31.19$ dB) performance of ReSample [41].

2 Background

Diffusion models conceptualize the generation of data as the reverse of a noising process, where a data sample $\boldsymbol{x}_t$ at time t within the interval $[0, T]$ follows a specified stochastic differential equation (SDE). This forward SDE [34] for the data noising process is described by $d\boldsymbol{x} = -\frac{\beta_t}{2}\boldsymbol{x}\,dt + \sqrt{\beta_t}\,d\boldsymbol{w}$, where $\beta_t \in (0, 1)$ is a positive, monotonically increasing function of time t, and $\boldsymbol{w}$ represents a standard Wiener process. The process begins with an initial data distribution $\boldsymbol{x}_0 \sim p_{\text{data}}$ and transitions to an approximately Gaussian distribution $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ by time T. The objective of regenerating the original data distribution from this Gaussian distribution involves reversing the noising process through a reverse SDE of the form $d\boldsymbol{x} = \left[-\frac{\beta_t}{2}\boldsymbol{x} - \beta_t \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right] dt + \sqrt{\beta_t} d\bar{\boldsymbol{w}}$, where dt indicates time moving backward and $\bar{\boldsymbol{w}}$ is the reversed Wiener process. To approximate $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$, a neural network s_θ trained through denoising score matching [44] is used.

Diffusion-based approaches to inverse problems seek to reconstruct the original data $\boldsymbol{x}_0$ from the measurement $\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}_0) + \boldsymbol{n}$. In this case, the reverse SDE implements $d\boldsymbol{x} = \left[-\frac{\beta_t}{2}\boldsymbol{x} - \beta_t(\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t))\right] dt + \sqrt{\beta_t} d\bar{\boldsymbol{w}}$. Conceptually, the learned score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ guides the reverse diffusion process from noise to the data distribution, and the likelihood term $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$ ensures measurement consistency. The main challenge of this approach is the lack of an explicit analytical expression for $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{y}|\boldsymbol{x}_t)$, as the exact relationship between $\boldsymbol{y}$ and intermediate states $\boldsymbol{x}_t$ is typically not well-defined, except at the initial state $\boldsymbol{x}_0$. To address this issue, prior works, notably Diffusion Posterior Sampling (DPS) [35], approximate the probability $p(\boldsymbol{y}|\boldsymbol{x}_t) \approx p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0 := \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t])$ using the conditional expectation of the data [40]. In the case of latent diffusion models, we can naturally extend this DPS assumption to approximate $p(\boldsymbol{y}|\boldsymbol{z}_t) \approx p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0 := \mathcal{D}(\mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t]))$ where $\boldsymbol{z}_t$ is the latent representation of $\boldsymbol{x}_t$, and $\mathcal{D}$ denotes the latent-to-pixel space decoder. We will refer to this as *Latent-DPS*. Such assumptions can be a source of the issue discussed above, and this work aims to alleviate its adverse effects on performance.

[2] PLoRG is a preliminary version of DiffStateGrad. See https://arxiv.org/abs/2410.03463.

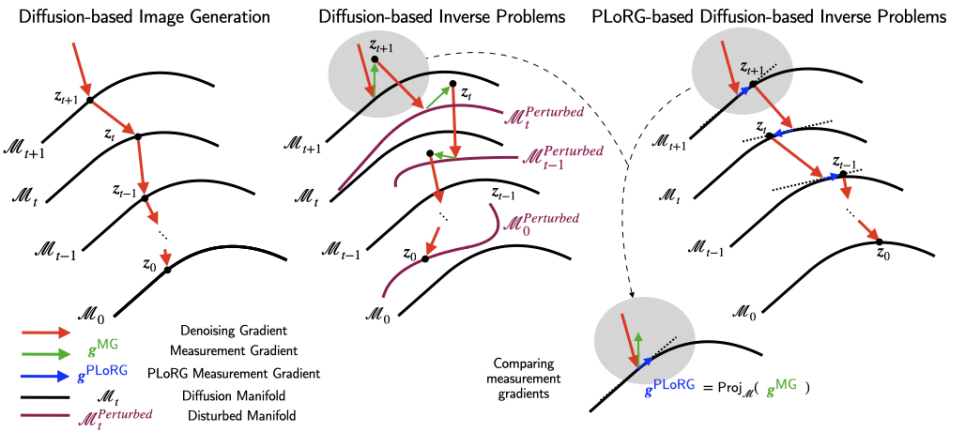


Figure 1: **High-level interpretation of PLoRG.** PLoRG helps to stay closer to the data manifold.

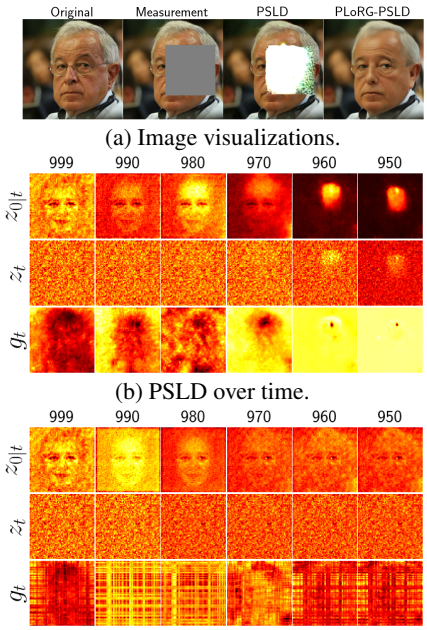
3 Projected Low-Rank Gradient (PLoRG)

Without loss of generality, we explain the proposed method in the context of Latent-DPS [35] and note that our method applies to a wide range of pixel or latent diffusion-based inverse solvers.

Method Given $\boldsymbol{z}_{t+1}$, we sample $\boldsymbol{z}_t$ from the unconditional reverse process, and then compute the estimate $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t) := \mathcal{D}(\mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t])$. Then, the data-consistency term can be incorporated as follows

$$\boldsymbol{z}_t \leftarrow \boldsymbol{z}_t - \eta_t \mathcal{P}_{\mathcal{S}_t}(\boldsymbol{G}_t), \tag{1}$$

where $\boldsymbol{G}_t = \nabla_{\boldsymbol{z}_{t+1}} \log p(\boldsymbol{y}|\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t))$ is the measurement gradient (MG), η_t is the step-size, and $\mathcal{P}_{\mathcal{S}_t}$ is a projection step onto the low-rank subspace $\mathcal{S}_t$, defined by $\boldsymbol{z}_t$. We implement the projection $\mathcal{P}_{\mathcal{S}_t}$ by computing the singular value decomposition (SVD) of $\boldsymbol{z}_t$ (i.e., $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V} \leftarrow \text{SVD}(\boldsymbol{z}_t)$). Then, we compute an adaptive rank $r \leftarrow \arg\min_k \{\frac{\sum_{j=1}^{k} s_j^2}{\sum_j s_j^2} \geq \tau\}$ leveraging a fixed variance retention threshold τ. The gradient $\boldsymbol{G}_t$ is projected onto a subspace defined by the highest r singular values of $\boldsymbol{z}_t$ as follows: $\boldsymbol{G}_t \leftarrow \boldsymbol{U}_r \boldsymbol{U}_r^{\mathrm{T}} \boldsymbol{G}_t \boldsymbol{V}_r^{\mathrm{T}} \boldsymbol{V}_r$. where $\boldsymbol{U}_r$ and $\boldsymbol{V}_r$ contain the first r left and right singular vectors, respectively. We refer to this projection step as *Projected Low-Rank Gradient* (PLoRG), and argue that it helps to preserve $\boldsymbol{z}_t$ after the MG update on the manifold $\mathcal{M}_t$ (Figure 1).



(a) Image visualizations.

(b) PSLD over time.

(c) PLoRG-PSLD over time.

Figure 2: Effect of large MG step size.

Interpretation Let the initial latent state $\boldsymbol{z}_t$ be artifact-free, as the diffusion is trained on clean data samples. We argue that the artifacts can only be introduced via the data-consistency gradient step. Hence, by projecting the MG onto a lower-rank subspace defined by the current $\boldsymbol{z}_t$, we filter out artifact-inducing components and encourage the process to remain within the manifold. This creates an inductive process: if $\boldsymbol{z}_t$ is artifact-free, and we only allow updates that align with its structure, subsequent $\boldsymbol{z}_t$ will remain artifact-free. Figure 2 demonstrates the effectiveness of PLoRG in removing artifacts when the MG step size is large; artifacts are introduced onto the gradient and stay within the latent representation in Posterior Sampling with Latent Diffusion (PSLD) [40], whereas the reverse process via PLoRG-PSLD (our method applied to PSLD) stays artifact-free.

4 Results

We evaluate the performance of PLoRG in conjunction with existing latent diffusion models, specifically PSLD [40] and ReSample [41]. Our evaluation is made based on key quantitative metrics,

Table 1: **Robustness comparison of PSLD and PLoRG-PSLD on linear tasks under different gradient step sizes**. Values represent mean performance over 100 images on FFHQ 256×256.

Method	Inpaint (Box)			Inpaint (Random)			Gaussian deblur			Motion deblur			SR (×4)		
	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
Normal MG step size															
PSLD	0.158	0.819	24.22	0.246	0.809	29.05	0.357	0.537	22.87	0.322	0.615	24.25	**0.313**	**0.650**	24.51
PLoRG-PSLD (ours)	0.095	0.875	23.76	0.265	0.793	28.14	0.366	0.516	22.24	0.335	0.585	23.34	0.392	0.516	22.12
Large MG step size															
PSLD	0.252	0.728	11.99	0.463	0.551	20.62	0.549	0.318	17.47	0.514	0.390	18.81	0.697	0.075	7.700
PLoRG-PSLD (ours)	**0.092**	**0.880**	**24.32**	**0.165**	**0.898**	**31.68**	**0.355**	**0.542**	**22.95**	**0.319**	**0.620**	**24.31**	0.320	0.640	**24.56**

Table 2: **Performance comparison of ReSample and PLoRG-ReSample on linear and nonlinear tasks.** Values represent mean performance over 100 images on FFHQ 256×256.

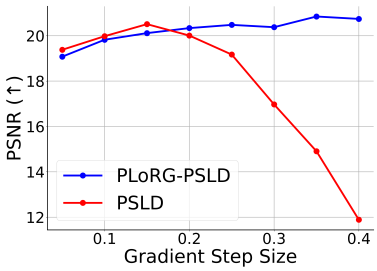
Method	Inpaint (Box)			Inpaint (Random)			Phase Retrieval		
	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
ReSample	0.198	0.807	19.91	0.115	0.892	31.27	0.237	0.750	27.61
PLoRG-ReSample (ours)	**0.156**	**0.841**	**23.59**	**0.106**	**0.913**	**31.91**	**0.154**	**0.855**	**31.19**

including LPIPS (Learned Perceptual Image Patch Similarity), PSNR (Peak Signal-to-Noise Ratio), and SSIM (Structural Similarity Index). We use the FFHQ 256×256 dataset [45] and the pre-trained latent diffusion model LDM-VQ-4 trained on FFHQ [43].

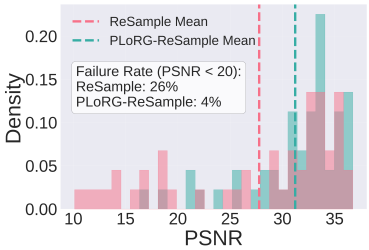
We consider linear (box inpainting, random inpainting, Gaussian deblurring, motion deblurring, and super-resolution), and nonlinear (phase retrieval) inverse problems (see Appendix). We evaluate the performance of PLoRG applied to two SOTA methods of PSLD [40] (Table 1) and ReSample [41] (Table 2).

Robustness Table 1 and Figure 3a exhibit the sensitivity of PSLD to the choice of MG step size; the performance of PSLD significantly deteriorates when a relatively large MG step size is used, leading to poor results across all tasks. In contrast, PLoRG-PSLD shows robustness and maintains high performance over a wide range of MG step sizes.

Performance Tables 1 and 2 demonstrate the significant outperformance of PLoRG-PSLD and PLoRG-ReSample against their respective SOTA counterparts for inpainting tasks across all three metrics. Moreover, Table 2 shows that PLoRG-ReSample substantially outperforms the ReSample SOTA for phase retrieval. We attribute the effectiveness of PLoRG, particularly in challenging tasks (i.e., inpainting and phase retrieval), to a reduced rate of failure cases (see Figure 3b) resulting from keeping reconstructions closer to the latent manifold of natural images. By constraining solutions to this manifold, PLoRG minimizes extreme failures and enhances consistency in reconstruction quality, and improves overall performance metrics.



(a) PLoRG robustness to MG step size.



(b) PSNR histogram of ReSample and PLoRG-ReSample for phase retrieval.

Figure 3: Robustness and performance comparison of PLoRG for PSLD [40] and ReSample [41].

5 Conclusion

We introduce a *Projected Low-Rank Gradient* (PLoRG) method to enhance the performance and robustness of diffusion models in solving inverse problems. We show that PLoRG improve SOTA methods such as PSLD and ReSample. PLoRG addresses the introduction of artifacts and deviations from the data manifold by constraining gradient updates to a lower-rank subspace. PLoRG is versatile, applicable across various diffusion models and sampling algorithms, and includes an adaptive rank selection mechanism that dynamically adjusts to the gradient's complexity. PLoRG reduces the need for excessive tuning of gradient step sizes and significantly boosts performance for more challenging inverse problems. Finally, PLoRG enhances reconstruction quality with minimal computational overhead, making it a practical addition to diffusion modeling for reliable real-world applications.

References

[1] C. W. Groetsch and C. Groetsch, *Inverse problems in the mathematical sciences*, vol. 52. Springer, 1993.

[2] M. Bertero, P. Boccacci, and C. De Mol, *Introduction to inverse problems in imaging*. CRC press, 2021.

[3] F. Yaman, V. G. Yakhno, and R. Potthast, "A survey on inverse problems for applied sciences," *Mathematical problems in engineering*, vol. 2013, no. 1, p. 976837, 2013.

[4] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, "Robust Compressed Sensing MRI with Deep Generative Priors," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 14938–14954, Curran Associates, Inc., 2021.

[5] A. L. Y. Hung, K. Zhao, H. Zheng, R. Yan, S. S. Raman, D. Terzopoulos, and K. Sung, "Med-CDiff: Conditional Medical Image Generation with Diffusion Models," *Bioengineering*, vol. 10, no. 11, p. 1258, 2023.

[6] Z. Dorjsembe, H.-K. Pao, S. Odonchimed, and F. Xiao, "Conditional Diffusion Models for Semantic 3D Brain MRI Synthesis," *IEEE Journal of Biomedical and Health Informatics*, 2024.

[7] L. Pineda, S. Basu, A. Romero, R. Calandra, and M. Drozdzal, "Active MR k-space Sampling with Reinforcement Learning," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020.

[8] J. A. Fessler, "Model-Based Image Reconstruction for MRI," *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 81–89, 2010.

[9] W. Liu, X. Xia, L. Xiong, Y. Lu, L. Gao, and Z. Yu, "Automated Vehicle Sideslip Angle Estimation Considering Signal Measurement Characteristic," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21675–21687, 2021.

[10] W. Liu, K. Quijano, and M. M. Crawford, "YOLOv5-Tassel: Detecting Tassels in RGB UAV Imagery with Improved YOLOv5 Based on Transfer Learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 8085–8094, 2022.

[11] K. Akiyama, K. Kuramochi, S. Ikeda, V. L. Fish, F. Tazaki, M. Honma, S. S. Doeleman, A. E. Broderick, J. Dexter, M. Mościbrodzka, *et al.*, "Imaging the Schwarzschild-radius-scale Structure of M87 with the Event Horizon Telescope using Sparse Modeling," *The Astrophysical Journal*, vol. 838, no. 1, p. 1, 2017.

[12] E. Jonas, "Deep Imitation Learning for Molecular Inverse Problems," in *Advances in Neural Information Processing Systems*, 2019.

[13] D. Schwalbe-Koda, A. R. Tan, and R. G'omez-Bombarelli, "Differentiable Sampling of Molecular Geometries with Uncertainty-Based Adversarial Attacks," *Nature Communications*, 2021.

[14] V. Isakov, *Inverse problems for Partial Differential Equations*, vol. 127. Springer, 2006.

[15] A. Spielberg, F. Zhong, K. Rematas, K. M. Jatavallabhula, C. Oztireli, T.-M. Li, and D. Nowrouzezahrai, "Differentiable visual computing for inverse problems and machine learning," *Nature Machine Intelligence*, vol. 5, no. 11, pp. 1189–1199, 2023.

[16] J. Janai, F. Güney, A. Behl, A. Geiger, *et al.*, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, no. 1–3, pp. 1–308, 2020.

[17] F. Ding, K. Yu, Z. Gu, X. Li, and Y. Shi, "Perceptual enhancement for autonomous vehicles: Restoring visually degraded images for context prediction via adversarial training," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9430–9441, 2021.

[18] A. M. Stuart, "Inverse problems: a bayesian perspective," *Acta numerica*, vol. 19, pp. 451–559, 2010.

[19] D. L. Donoho, "Compressed Sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[20] E. J. Candès, J. Romberg, and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[21] G.-H. Chen, J. Tang, and S. Leng, "Prior image constrained compressed sensing (piccs): a method to accurately reconstruct dynamic ct images from highly undersampled projection data sets," *Medical physics*, vol. 35, no. 2, pp. 660–663, 2008.

[22] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse mri: The application of compressed sensing for rapid mr imaging," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[23] L. C. Potter, E. Ertin, J. T. Parker, and M. Cetin, "Sparsity and compressed sensing in radar imaging," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1006–1020, 2010.

[24] S. Negahban, B. Yu, M. J. Wainwright, and P. Ravikumar, "A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers," *Advances in neural information processing systems*, vol. 22, 2009.

[25] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on information theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[26] D. P. Kingma, "Auto-encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," *Advances in neural information processing systems*, vol. 27, 2014.

[28] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed Sensing using Generative Models," in *International conference on machine learning*, pp. 537–546, PMLR, 2017.

[29] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep Image Prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9446–9454, 2018.

[30] A. G. Dimakis, "Deep generative models and inverse problems," *Mathematical Aspects of Deep Learning*, vol. 400, 2022.

[31] Y. Song and S. Ermon, "Generative Modeling by Estimating Gradients of the Data Distribution," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[32] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep Learning Techniques for Inverse Problems in Imaging," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 39–56, 2020.

[33] Y. Song and S. Ermon, "Improved Techniques for Training Score-Based Generative Models," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 12438–12448, 2020.

[34] Y. Song, J. Sohl-Dickstein, D. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based Generative Modeling through Stochastic Differential Equations," in *The International Conference on Learning Representations*, 2021.

[35] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion Posterior Sampling for General Noisy Inverse Problems," in *The Eleventh International Conference on Learning Representations*, 2023.

[36] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising Diffusion Restoration Models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23593–23606, 2022.

[37] J. Song, A. Vahdat, M. Mardani, and J. Kautz, "Pseudoinverse-Guided Diffusion Models for Inverse Problems," in *International Conference on Learning Representations*, 2023.

[38] D. Shu, Z. Li, and A. B. Farimani, "A Physics-informed Diffusion Model for High-fidelity Flow Field Reconstruction," *Journal of Computational Physics*, vol. 478, p. 111972, 2023.

[39] J. Huang, G. Yang, Z. Wang, and J. J. Park, "DiffusionPDE: Generative PDE-Solving Under Partial Observation," *arXiv:2406.17763*, 2024.

[40] L. Rout, N. Raoof, G. Daras, C. Caramanis, A. Dimakis, and S. Shakkottai, "Solving Linear Inverse Problems Provably via Posterior Sampling with Latent Diffusion Models," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[41] B. Song, S. M. Kwon, Z. Zhang, X. Hu, Q. Qu, and L. Shen, "Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency," in *Conference on Parsimony and Learning (Recent Spotlight Track)*, 2023.

[42] B. Zhang, W. Chu, J. Berner, C. Meng, A. Anandkumar, and Y. Song, "Improving Diffusion Inverse Problem Solving with Decoupled Noise Annealing," *arXiv preprint arXiv:2407.01521*, 2024.

[43] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution Image Synthesis with Latent Diffusion Models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[44] P. Vincent, "A Connection Between Score Matching and Denoising Autoencoders," *Neural Computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[45] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 43, pp. 4217–4228, Dec 2021.

[46] H. Chung, B. Sim, D. Ryu, and J. C. Ye, "Improving Diffusion Models for Inverse Problems using Manifold Constraints," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25683–25696, 2022.

[47] J. Zhao, Z. Zhang, B. Chen, Z. Wang, A. Anandkumar, and Y. Tian, "GaLore: Memory-efficient LLM Training by Gradient Low-rank Projection," *arXiv preprint arXiv:2403.03507*, 2024.

[48] B. T. Feng, K. L. Bouman, and W. T. Freeman, "Event-horizon-scale imaging of m87* under different assumptions via deep generative image priors," *arXiv preprint arXiv:2406.02785*, 2024.

A Additional Background

Solving Inverse Problems with Latent Diffusion Models For complex scenarios where direct application of pixel-based models is inefficient or ineffective, latent diffusion models (LDMs) offer a promising alternative [43]. Given original data $\boldsymbol{x} \in \mathbb{R}^n$, the LDM framework utilizes an encoder $\mathcal{E} : \mathbb{R}^n \to \mathbb{R}^k$ and a decoder $\mathcal{D} : \mathbb{R}^k \to \mathbb{R}^n$, with $k \ll n$, to work in a compressed latent space. $\boldsymbol{x}_T$ is encoded into $\boldsymbol{z}_T = \mathcal{E}(\boldsymbol{x}_T)$ and serves as the starting point for the reverse diffusion process. Then, $\boldsymbol{z}_0$ is decoded to $\boldsymbol{x}_0 = \mathcal{D}(\boldsymbol{z}_0)$, the final clean image.

The most direct extension of DPS [35] would be to approximate $p(\boldsymbol{y}|\boldsymbol{z}_t) \approx p(\boldsymbol{y}|\hat{\boldsymbol{x}}_0 := \mathcal{D}(\mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t]))$. However, this approach fails due to the nonlinear nature of the decoder [40]. This failure is exacerbated by the decoder's one-to-many mapping, where multiple latent vectors $\boldsymbol{z}_t$ can potentially decode to images $\hat{\boldsymbol{x}}_0$ that are consistent with the measurements $\boldsymbol{y}$. During reverse diffusion, when updating based on $\nabla_{\boldsymbol{z}_t} \log p_t(\boldsymbol{y}|\boldsymbol{z}_t)$ (we refer to this as the *measurement gradient*), the decoder may push $\boldsymbol{z}_t$ towards various $\hat{\boldsymbol{x}}_0$ values in potentially conflicting directions. This occurs because different $\hat{\boldsymbol{x}}_0$ might equally well explain the measurements $\boldsymbol{y}$ while corresponding to different regions in the latent space. Consequently, the measurement gradient update could drive $\boldsymbol{z}_t$ away from the manifold of realistic images, leading to several issues such as the introduction of artifacts, reduced sample quality, and potential convergence problems.

Previous approaches have attempted to address these challenges through various means, including imposing manifold constraints [46], enforcing fixed-point properties on latent representations [40], and directly imposing "hard" measurement consistency by solving optimization problems with gradient descent [41]. However, these solutions often produce inconsistent reconstructions and are highly sensitive to the MG step size [43]. The complexity of balancing the MG with the score function guidance remains a significant challenge in measurement-guided generation in latent diffusion models.

B Discussion

B.1 Efficiency

Our method incurs minimal computational overhead as we perform SVD only once per iteration in the latent space, typically on 64×64 matrices. Additionally, by adaptively selecting a low rank based on a variance threshold, the subsequent operations—projection and reconstruction—are performed on reduced matrices, further reducing complexity. The more significant computational cost occurs during the nonlinear decoding of the latent that is outside of our algorithm's scope. Finally, we note that PLoRG can improve the runtime and computational efficiency of diffusion frameworks that use Adam optimizers for data consistency [41, 47].

B.2 Limitations

Our PLoRG method assumes that the learned prior is a relatively good prior for the task at hand. PLoRG encourages the process to stay close to the manifold structure captured by the generative prior and may introduce the biases of the prior into image restoration tasks. Hence, PLoRG may not be recommended for delicate inverse problems such as black hole imaging [48].

C Additional Results

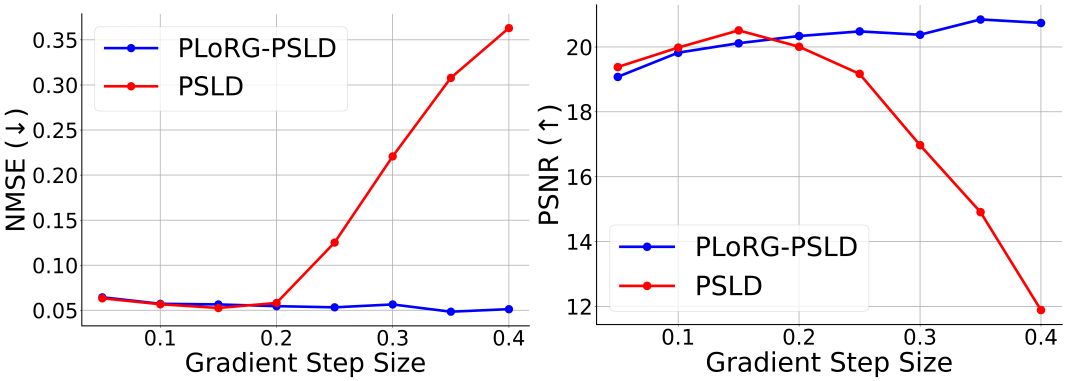
Table 3: **Performance comparison of PSLD and PLoRG-PSLD on various linear tasks in the image domain.** Values represent mean performance over 100 images on FFHQ 256×256.

Method	Inpaint (Box)		Inpaint (Random)		Gaussian deblur		Motion deblur		SR (×4)	
	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑
PSLD	0.158	24.22	0.246	29.05	0.357	22.87	0.322	24.25	**0.313**	24.51
PLoRG-PSLD (ours)	**0.092**	**24.32**	**0.165**	**31.68**	**0.355**	**22.95**	**0.319**	**24.31**	0.320	**24.56**

Table 4: **Performance comparison of different latent diffusion methods on the phase retrieval task.** Values represent mean performance over 100 images on FFHQ 256×256.

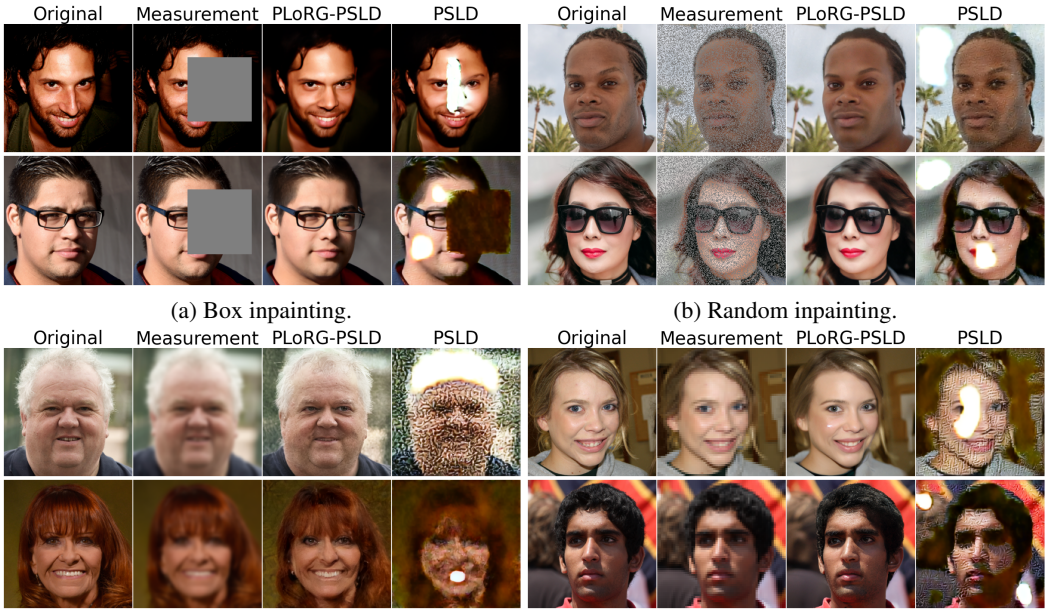
Method	Phase Retrieval	
	LPIPS↓	PSNR↑
LatentDAPS	0.199	29.16
ReSample	0.237	27.61
PLoRG-ReSample (ours)	**0.154**	**31.19**

D Visualizations



(a) NMSE (b) PSNR

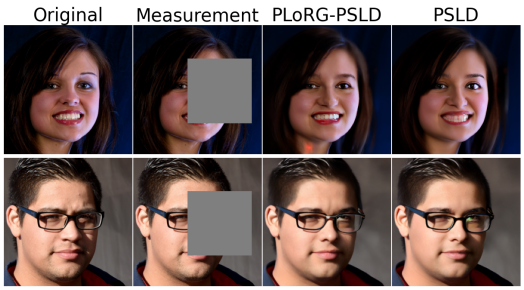
Figure 4: Robustness of PLoRG to step size guidance.



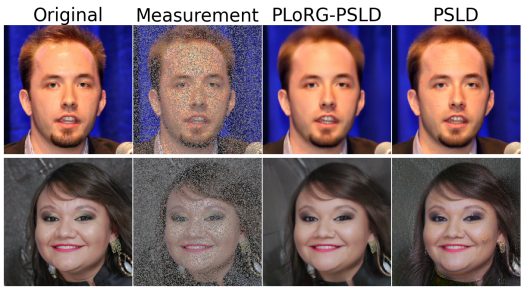
(a) Box inpainting. (b) Random inpainting.

(c) Gaussian deblur. (d) Super-resolution ($\times 4$).

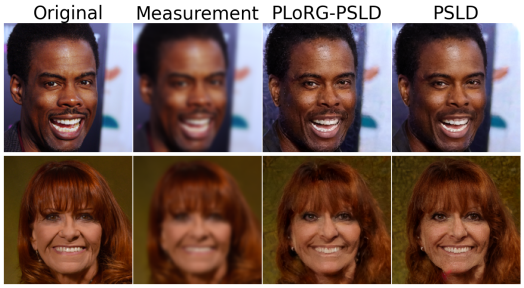
Figure 5: **Performance comparison of PLoRG-PSLD (our method) and PSLD for large MG step size**. Images are chosen at random for visualization.



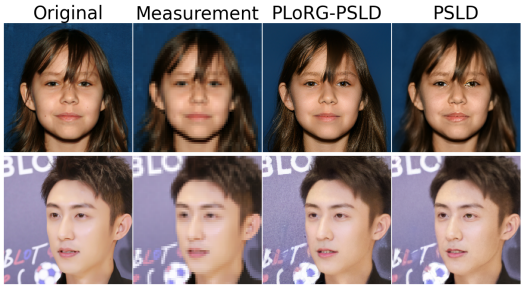
(a) Box inpainting.



(b) Random inpainting.

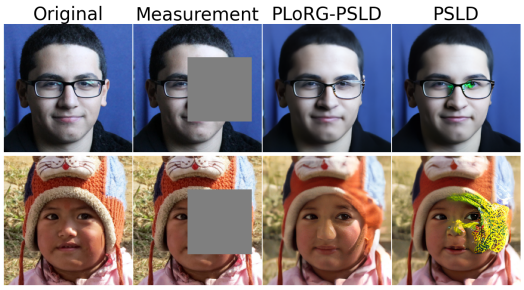


(c) Gaussian deblur.

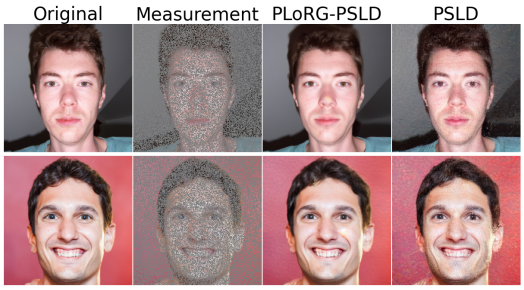


(d) Super-resolution ($\times 4$).

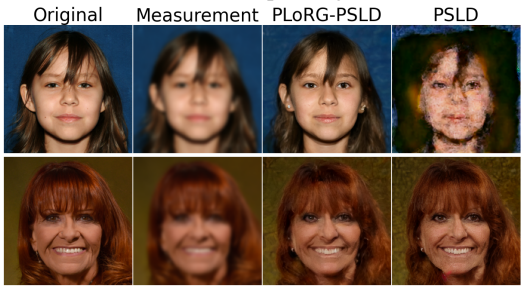
Figure 6: **Performance comparison of PLoRG-PSLD (our method) and PSLD for their best-performing MG step size**. Images are chosen at random for visualization.



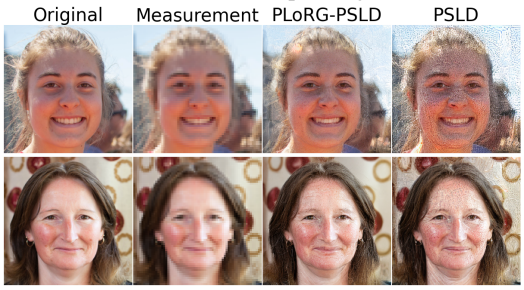
(a) Box inpainting.



(b) Random inpainting.

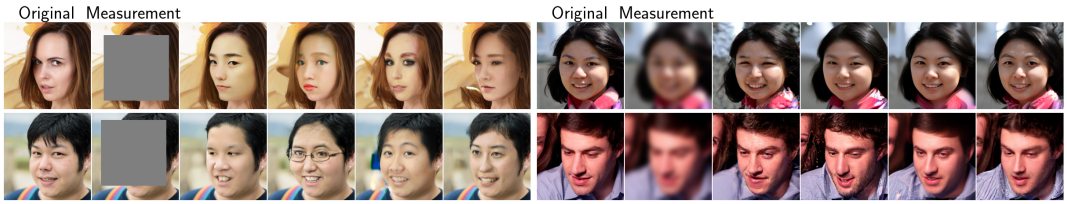


(c) Gaussian deblur.



(d) Super-resolution ($\times 4$).

Figure 7: **Performance comparison of PLoRG-PSLD (our method) and PSLD for their best-performing MG step size**. Images are chosen at random for visualization.



(a) Box inpainting (192 × 192 box). (b) Gaussian deblur (81 × 81 kernel, SD of 7).

Figure 8: **Reconstruction diversity of PLoRG.** PLoRG-PSLD with a large MG step size can produce a diverse range of images from multimodal posteriors. Generated images have distinctive facial features.

E Additional Algorithms

Here, we provide the general algorithm for integrating PLoRG into existing latent diffusion methods. We note that when we project the gradient, we use the *full* SVD projection (i.e., combining both left and right projection). However, in practice, one may decide to do either left or right projection.

Algorithm 1 Projected Low-Rank Gradient (PLoRG) for Latent Diffusion-based Inverse Problems

Require: Normal input + variance retention threshold τ
Let T = number of total iterations of sampling algorithm, and assume we calculate latent image representation $\boldsymbol{z}_t$ for each iteration
for $t = T-1$ **to** 0 **do**
  Compute measurement gradient $\boldsymbol{G}_t$ according to sampling algorithm
  $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V} \leftarrow \text{SVD}(\boldsymbol{z}_t)$ ▷ Perform SVD on latent representation
  $\lambda_j \leftarrow s_j^2$ (where s_j are the singular values of $\boldsymbol{S}$) ▷ Calculate eigenvalues
  $c_k \leftarrow \frac{\sum_{j=1}^{k} \lambda_j}{\sum_j \lambda_j}$ ▷ Cumulative sum of eigenvalues
  $r \leftarrow \arg\min_k \{c_k \geq \tau\}$ ▷ Determine rank r based on threshold τ
  $\boldsymbol{A}_t \leftarrow \boldsymbol{U}[:, :r]$ ▷ Left singular vectors
  $\boldsymbol{B}_t \leftarrow \boldsymbol{V}[:, :r]^T$ ▷ Right singular vectors
  $\boldsymbol{R}_t \leftarrow \boldsymbol{A}_t^T \boldsymbol{G}_t \boldsymbol{B}_t^T$ ▷ Project gradient
  $\boldsymbol{G}_t' \leftarrow \boldsymbol{A}_t \boldsymbol{R}_t \boldsymbol{B}_t$ ▷ Reconstruct approximated gradient
  Use updated gradient $\boldsymbol{G}_t'$ in sampling algorithm
end for
return $\mathcal{D}(\hat{\boldsymbol{z}}_0)$

F Implementation Details

F.1 Experimental Setup

Our experiments consider both linear and nonlinear inverse problems using natural images. We use the FFHQ 256×256 dataset [45] and the pre-trained latent diffusion model LDM-VQ-4 trained on FFHQ [43]. For evaluation, we sample a fixed set of 100 images from the FFHQ validation set. Images are normalized to the range [0, 1], and Gaussian noise is applied to the measurement with a standard deviation of $\sigma_y = 0.01$.

For linear inverse problems, we consider (1) box inpainting, (2) random inpainting, (3) Gaussian deblur, (4) motion deblur, and (5) super-resolution. In the box inpainting task, a random 128×128 box is used, while the random inpainting task employs a 70% random mask. Gaussian and motion deblurring tasks utilize kernels sized 61×61, with standard deviations of 3.0 and 0.5, respectively. For super-resolution, images are downscaled by a factor of 4 using a bicubic resizer.

For nonlinear inverse problems, we consider phase retrieval. We use an oversampling rate of 2.0, and due to the instability and non-uniqueness of reconstruction, we adopt the strategy from DPS [35] and DAPS [42], generating four separate reconstructions and reporting the best result. Like DAPS [42],

we normalize the data to lie in the range [0, 1] before applying the discrete Fourier transform. We note that PSLD is not designed to handle nonlinear inverse problems.

F.2 Code Release

We will release the GitHub repository with our code upon acceptance to the workshop.

F.3 Hyperparameters

We set the variance retention threshold τ to 0.99 for all experiments.

F.4 PSLD

Our PLoRG-PSLD algorithm integrates the low-rank gradient approximation directly into the PSLD update process. At each iteration of the main loop, after computing the standard PSLD update $\boldsymbol{z}'_{t-1}$, we introduce our PLoRG method. First, we calculate the full gradient $\boldsymbol{G}_t$ according to PSLD, combining both the measurement consistency term and the fixed-point constraint. We then perform SVD on the current latent representation $\boldsymbol{z}_t$. Using the variance retention threshold τ, we determine the appropriate rank for our approximation. We construct projection matrices from the truncated singular vectors and use these to approximate the gradient. This approximated gradient $\boldsymbol{G}'_t$ is then used for the final update step, replacing the separate gradient updates in standard PSLD. This process is repeated at every iteration, allowing for adaptive, low-rank updates throughout the entire diffusion process.

Algorithm 2 PSLD

Require: $T, \boldsymbol{y}, \{\eta_t\}_{t=1}^T, \{\gamma_t\}_{t=1}^T, \{\tilde{\sigma}_t\}_{t=1}^T$
Require: $\mathcal{E}, \mathcal{D}, \mathcal{A}\boldsymbol{x}_0^*, \mathcal{A}, \boldsymbol{s}_\theta$
$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T-1$ **to** 0 **do**
  $\hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{z}_t, t)$
  $\hat{\boldsymbol{z}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{z}_t + (1-\bar{\alpha}_t)\hat{\boldsymbol{s}})$
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{z}'_{t-1} \leftarrow \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{z}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\boldsymbol{z}}_0 + \tilde{\sigma}_t\epsilon$
  $\boldsymbol{z}''_{t-1} \leftarrow \boldsymbol{z}'_{t-1} - \eta_t \nabla_{\boldsymbol{z}_t}\|\boldsymbol{y} - \mathcal{A}(\mathcal{D}(\hat{\boldsymbol{z}}_0))\|_2^2$
  $\boldsymbol{z}_{t-1} \leftarrow \boldsymbol{z}''_{t-1} - \gamma_t \nabla_{\boldsymbol{z}_t}\|\hat{\boldsymbol{z}}_0 - \mathcal{E}(\mathcal{A}^T\mathcal{A}\boldsymbol{x}_0^* + (\mathbf{I} - \mathcal{A}^T\mathcal{A})\mathcal{D}(\hat{\boldsymbol{z}}_0))\|_2^2$
end for
return $\mathcal{D}(\hat{\boldsymbol{z}}_0)$

Algorithm 3 PLoRG-PSLD

Require: $T, \boldsymbol{y}, \{\eta_t\}_{t=1}^T, \{\gamma_t\}_{t=1}^T, \{\tilde{\sigma}_t\}_{t=1}^T$
Require: $\mathcal{E}, \mathcal{D}, \mathcal{A}\boldsymbol{x}_0^*, \mathcal{A}, \boldsymbol{s}_\theta$, Variance retention threshold τ

$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T-1$ **to** 0 **do**
  $\hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{z}_t, t)$
  $\hat{\boldsymbol{z}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{z}_t + (1-\bar{\alpha}_t)\hat{\boldsymbol{s}})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\boldsymbol{z}'_{t-1} \leftarrow \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{z}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\boldsymbol{z}}_0 + \tilde{\sigma}_t\boldsymbol{\epsilon}$
  $\boldsymbol{G}_t \leftarrow \eta_t \nabla_{\boldsymbol{z}_t}\|\boldsymbol{y} - \mathcal{A}(\mathcal{D}(\hat{\boldsymbol{z}}_0))\|_2^2 + \gamma_t \nabla_{\boldsymbol{z}_t}\|\hat{\boldsymbol{z}}_0 - \mathcal{E}(\mathcal{A}^T\mathcal{A}\boldsymbol{x}_0^* + (\boldsymbol{I} - \mathcal{A}^T\mathcal{A})\mathcal{D}(\hat{\boldsymbol{z}}_0))\|_2^2$
  $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V} \leftarrow \mathrm{SVD}(\boldsymbol{z}_t)$
  $\lambda_j \leftarrow s_j^2$ (s_j are the singular values of $\boldsymbol{S}$)
  $c_k \leftarrow \frac{\sum_{j=1}^k \lambda_j}{\sum_j \lambda_j}$
  $r \leftarrow \arg\min_k \{c_k \geq \tau\}$
  $\boldsymbol{A}_t \leftarrow \boldsymbol{U}[:, :r]$
  $\boldsymbol{B}_t \leftarrow \boldsymbol{V}[:r, :]$
  $\boldsymbol{R}_t \leftarrow \boldsymbol{A}_t^T \boldsymbol{G}_t \boldsymbol{B}_t^T$
  $\boldsymbol{G}'_t \leftarrow \boldsymbol{A}_t \boldsymbol{R}_t \boldsymbol{B}_t$
  $\boldsymbol{z}_{t-1} \leftarrow \boldsymbol{z}'_{t-1} - \boldsymbol{G}'_t$
end for
return $\mathcal{D}(\hat{\boldsymbol{z}}_0)$

F.5 ReSample

Our PLoRG-ReSample algorithm integrates the low-rank gradient approximation into the optimization process of ReSample [41]. We introduce two new hyperparameters: the variance retention threshold τ and a frequency F for applying the low-rank approximation. During each ReSample step, we first perform SVD on the current latent representation $\boldsymbol{z}'_t$. Note that we do not perform SVD within the gradient descent loop itself, meaning that we still only perform SVD once per iteration of the sampling algorithm. Then, within the gradient descent loop for solving $\hat{\boldsymbol{z}}_0(\boldsymbol{y})$, we apply our PLoRG method every F steps. This involves determining the appropriate rank based on τ, constructing projection matrices, and using these to approximate the gradient. On steps where PLoRG is not applied, we use the standard gradient. This adaptive, periodic application of PLoRG allows for a balance between the benefits of low-rank approximation and the potential need for full gradient information. The rest of the ReSample algorithm, including the stochastic resampling step, remains unchanged. This integration allows PLoRG to influence the critical optimization step in ReSample, improving its robustness to measurement errors while maintaining its core functionality.

We note that the ReSample algorithm employs a two-stage approach for its hard data consistency step. Initially, it performs pixel-space optimization. This step is computationally efficient and produces smoother, albeit potentially blurrier, results with high-level semantic information. As the diffusion process approaches $t = 0$, ReSample transitions to latent-space optimization to refine the image with finer details. Our PLoRG method is specifically integrated into this latter, latent-space optimization stage. By applying PLoRG to the latent optimization, we aim to mitigate the potential introduction of artifacts and off-manifold deviations that can occur due to the direct manipulation of latent variables. This strategic application of PLoRG allows us to benefit from the computational efficiency of initial pixel-space optimization while enhancing the robustness and quality of the final latent-space refinement. Importantly, PLoRG is not applied during the pixel-space optimization phase, as this stage already tends to produce smoother results and is less prone to artifact introduction.

Algorithm 4 ReSample

Require: Measurements $\boldsymbol{y}$, $\mathcal{A}(\cdot)$, Encoder $\mathcal{E}(\cdot)$, Decoder $\mathcal{D}(\cdot)$, Score function $\boldsymbol{s}_\theta(\cdot, t)$, Pretrained LDM Parameters β_t, $\bar{\alpha}_t$, η, δ, Hyperparameter γ to control σ_t^2, Time steps to perform resample C

$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ ▷ Initial noise vector
for $t = T-1, \ldots, 0$ **do**
  $\boldsymbol{\epsilon}_1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\hat{\boldsymbol{\epsilon}}_{t+1} = \boldsymbol{s}_\theta(\boldsymbol{z}_{t+1}, t+1)$ ▷ Compute the score
  $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_{t+1}) = \frac{1}{\sqrt{\bar{\alpha}_{t+1}}}(\boldsymbol{z}_{t+1} - \sqrt{1-\bar{\alpha}_{t+1}}\hat{\boldsymbol{\epsilon}}_{t+1})$ ▷ Predict $\hat{\boldsymbol{z}}_0$ using Tweedie's formula
  $\boldsymbol{z}'_t = \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{z}}_0(\boldsymbol{z}_{t+1}) + \sqrt{1-\bar{\alpha}_t-\eta\delta^2}\hat{\boldsymbol{\epsilon}}_{t+1} + \eta\delta\boldsymbol{\epsilon}_1$ ▷ Unconditional DDIM step
  if $t \in C$ **then** ▷ ReSample time step
    $\hat{\boldsymbol{z}}_0(\boldsymbol{y}) \in \arg\min_{\boldsymbol{z}} \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}(\mathcal{D}(\boldsymbol{z}))\|_2^2$ ▷ Solve with initial point $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_{t+1})$
    $\boldsymbol{z}_t = \text{StochasticResample}(\hat{\boldsymbol{z}}_0(\boldsymbol{y}), \boldsymbol{z}'_t, \gamma)$ ▷ Map back to t
  else
    $\boldsymbol{z}_t = \boldsymbol{z}'_t$ ▷ Unconditional sampling if not resampling
  end if
end for
$\boldsymbol{x}_0 = \mathcal{D}(\boldsymbol{z}_0)$ ▷ Output reconstructed image
return $\boldsymbol{x}_0$

Algorithm 5 PLoRG-ReSample

Require: Measurements $\boldsymbol{y}$, $\mathcal{A}(\cdot)$, Encoder $\mathcal{E}(\cdot)$, Decoder $\mathcal{D}(\cdot)$, Score function $\boldsymbol{s}_\theta(\cdot, t)$, Pretrained LDM Parameters β_t, $\bar{\alpha}_t$, η, δ, Hyperparameter γ to control σ_t^2, Time steps to perform resample C, Variance retention threshold τ, Frequency F

$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ ▷ Initial noise vector
for $t = T-1, \ldots, 0$ **do**
  $\boldsymbol{\epsilon}_1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\hat{\boldsymbol{\epsilon}}_{t+1} = \boldsymbol{s}_\theta(\boldsymbol{z}_{t+1}, t+1)$ ▷ Compute the score
  $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_{t+1}) = \frac{1}{\sqrt{\bar{\alpha}_{t+1}}}(\boldsymbol{z}_{t+1} - \sqrt{1-\bar{\alpha}_{t+1}}\hat{\boldsymbol{\epsilon}}_{t+1})$ ▷ Predict $\hat{\boldsymbol{z}}_0$ using Tweedie's formula
  $\boldsymbol{z}'_t = \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{z}}_0(\boldsymbol{z}_{t+1}) + \sqrt{1-\bar{\alpha}_t-\eta\delta^2}\hat{\boldsymbol{\epsilon}}_{t+1} + \eta\delta\boldsymbol{\epsilon}_1$ ▷ Unconditional DDIM step
  if $t \in C$ **then** ▷ ReSample time step
    Initialize $\hat{\boldsymbol{z}}_0(\boldsymbol{y})$ with $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_{t+1})$
    $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V} \leftarrow \text{SVD}(\boldsymbol{z}'_t)$ ▷ Perform SVD on latent representation
    for each step in gradient descent **do**
      if step number $\mod F = 0$ **then**
        $\lambda_j \leftarrow s_j^2$ (where s_j are the singular values of $\boldsymbol{S}$) ▷ Calculate eigenvalues
        $c_k \leftarrow \frac{\sum_{j=1}^{k}\lambda_j}{\sum_j \lambda_j}$ ▷ Cumulative sum of eigenvalues
        $r \leftarrow \arg\min_k \{c_k \geq \tau\}$ ▷ Determine rank r based on threshold τ
        $\boldsymbol{A} \leftarrow \boldsymbol{U}[:, :r]$ ▷ Left singular vectors
        $\boldsymbol{B} \leftarrow \boldsymbol{V}[:, :r]^T$ ▷ Right singular vectors
        $\boldsymbol{G} \leftarrow \nabla_{\hat{\boldsymbol{z}}_0(\boldsymbol{y})} \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}(\mathcal{D}(\hat{\boldsymbol{z}}_0(\boldsymbol{y})))\|_2^2$ ▷ Compute gradient
        $\boldsymbol{R} \leftarrow \boldsymbol{A}^T\boldsymbol{G}\boldsymbol{B}^T$ ▷ Project gradient
        $\boldsymbol{G}' \leftarrow \boldsymbol{A}\boldsymbol{R}\boldsymbol{B}$ ▷ Reconstruct approximated gradient
      else
        $\boldsymbol{G}' \leftarrow \nabla_{\hat{\boldsymbol{z}}_0(\boldsymbol{y})} \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}(\mathcal{D}(\hat{\boldsymbol{z}}_0(\boldsymbol{y})))\|_2^2$ ▷ Compute gradient without modification
      end if
      Update $\hat{\boldsymbol{z}}_0(\boldsymbol{y})$ using gradient $\boldsymbol{G}'$
    end for
    $\boldsymbol{z}_t = \text{StochasticResample}(\hat{\boldsymbol{z}}_0(\boldsymbol{y}), \boldsymbol{z}'_t, \gamma)$ ▷ Map back to t
  else
    $\boldsymbol{z}_t = \boldsymbol{z}'_t$ ▷ Unconditional sampling if not resampling
  end if
end for
$\boldsymbol{x}_0 = \mathcal{D}(\boldsymbol{z}_0)$ ▷ Output reconstructed image
return $\boldsymbol{x}_0$