Adapt and Diffuse: Sample-adaptive Reconstruction via Latent Diffusion Models

Zalan Fabian[*,1]
zfabian@usc.edu

Berk Tinaz[*,1]
tinaz@usc.edu

Mahdi Soltanolkotabi[1]
soltanol@usc.edu

Abstract

Inverse problems arise in a multitude of applications, where the goal is to recover a clean signal from noisy and possibly (non)linear observations. The difficulty of a reconstruction problem depends on multiple factors, such as the structure of the ground truth signal, the severity of the degradation, the implicit bias of the reconstruction model and the complex interactions between the above factors. This results in natural sample-by-sample variation in the difficulty of a reconstruction task, which is often overlooked by contemporary techniques, resulting in long inference times, subpar performance and wasteful resource allocation. We propose a novel method to estimate the degradation severity of noisy, degraded signals in the latent space of an autoencoder. We show that the estimated severity has strong correlation with the true corruption level and can give useful hints at the difficulty of reconstruction problems on a sample-by-sample basis. Furthermore, we propose a reconstruction method based on latent diffusion models that leverages the predicted degradation severities to fine-tune the reverse diffusion sampling trajectory and thus achieve sample-adaptive inference.

1 Introduction

Inverse problems arise in a multitude of computer vision [24, 23, 40], biomedical imaging [1, 37] and scientific [12, 10] applications, where the goal is to recover a clean signal from noisy and degraded observations. Data-driven supervised and unsupervised deep learning methods have established new state-of-the-art in tackling most inverse problems (see an overview in [28]).

A key shortcoming of available techniques is their inherent inability to adapt their compute power allocation to the difficulty of reconstructing a given corrupted sample. There is a natural sample-by-sample variation in the difficulty of recovery due to multiple factors. First, variations in the measurement process (e. g. more or less additive noise, different blur kernels) greatly impact the difficulty of reconstruction. Second, a sample can be inherently difficult to reconstruct for the particular model, if it is different from examples seen in the training set (out-of-distribution samples). Third, the amount of information loss due to the interaction between the specific sample and the applied degradation can vary vastly. Finally, the implicit bias of the model architecture towards certain signal classes (e.g. smooth for convolutional architectures) can be a key factor in determining the difficulty of a recovery task. Therefore, expending the same amount of compute to reconstruct all examples is potentially wasteful, especially on datasets with varied corruption parameters.

To the best of our knowledge, sample-adaptive methods have not been studied extensively in the literature. *Unrolled networks* [41, 38] map the iterations of popular optimizers to learnable submodules, where deeper networks can be used to tackle more challenging reconstruction tasks. However, network size is determined in training time and therefore these methods are unable to adapt on a sample-by-sample basis. *Deep Equilibrium Models* [2, 11] can leverage networks of arbitrary depth

[*] : equal contribution

[1] : Dept. of Electrical and Computer Engineering, University of Southern California, Los Angeles

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

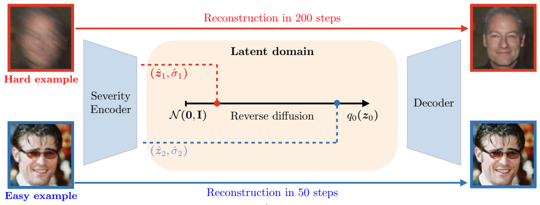
Reconstruction in 200 steps
Hard example
Severity Encoder
Latent domain
$(\hat{z}_1, \hat{\sigma}_1)$
$\mathcal{N}(\mathbf{0}, \mathbf{I})$
Reverse diffusion
$q_0(\boldsymbol{z}_0)$
$(\hat{z}_2, \hat{\sigma}_2)$
Decoder
Easy example
Reconstruction in 50 steps

Figure 1: We estimate the degradation severity of corrupted images in the latent space and find the optimal start time in the reverse diffusion process on a sample-by-sample basis. Inference time is automatically scaled by the difficulty of the reconstruction task at test time.

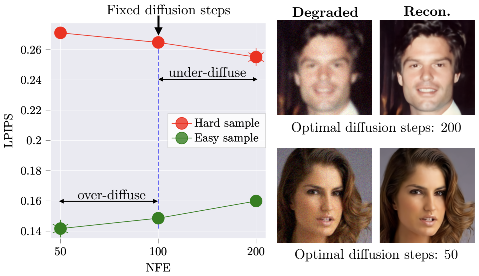

Figure 2: The optimal number of reverse diffusion steps varies depending on the severity of degradations. Fixing the number of steps results in over- or under-diffusing some samples.

through the construction of fixed-point iterations. These methods can adapt their compute in test time, however it is unclear how the optimal number of iterations correlates with degradation severity.

Diffusion models have established new state-of-the-art in synthesizing data of various modalities [8, 27, 29, 31, 34, 13, 33, 14, 22], inverse problem solving and image restoration [15, 35, 36, 7, 6, 4, 5, 20, 18, 19, 9]. Diffusion-based sampling techniques generate the missing information destroyed by the corruption step-by-step through a diffusion process that transforms pure noise into a target distribution. Recent work [6] has shown that the sampling trajectory can be significantly shortened by starting the reverse diffusion process from a good initialization, however even though sampling is accelerated, the same number of function evaluations are required to reconstruct any sample.

More recently, latent domain diffusion, that is a diffusion process defined in the low-dimensional latent space of a pre-trained autoencoder, has demonstrated great success in image synthesis [31] and has been successfully applied to solving linear inverse problems [32] and in high-resolution image restoration [26]. The latent space consists of compressed representations of relevant information in data and thus provides a natural space to quantify the loss of information due to image corruptions, which strongly correlates with the difficulty of the reconstruction task. In this work, we propose a novel reconstruction framework (Figure 1), where the cost of inference is automatically scaled based on the difficulty of the reconstruction task on a sample-by-sample basis. We call our method Flash-Diffusion: Fast Latent Sample-Adaptive Reconstruction ScHeme.

2 Method

Severity encoding – The goal of inverse problem solving is to recover the clean signal $\boldsymbol{x}$ from a corrupted observation $\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{n}$. The degradation $\mathcal{A}$ and additive noise $\boldsymbol{n}$ fundamentally destroy information in $\boldsymbol{x}$. The amount of information loss, or the *severity* of the degradation, strongly depends on the interaction between the signal structure and the specific degradation. For instance, applying a blur kernel to an image with abundant high-frequency detail (textures, hair, background clutter etc.) results in a *more severe degradation* compared to applying the same kernel to a smooth image with few details. In other words, the difficulty of recovering the clean signal does not solely depend on the degradation process itself, but also on the specific sample the degradation is applied to.

Quantifying the severity of a degradation is a challenging task in image domain. Consider the forward model $\boldsymbol{y} = c\boldsymbol{x}$ that simply rescales the clean signal. Recovery of $\boldsymbol{x}$ is trivial, however image similarity metrics such as PSNR or NMSE may indicate arbitrarily large error. Now, consider corrupting the clean signal with some additive Gaussian noise. Even though the image domain perturbation is (potentially) small, information is fundamentally lost and perfect reconstruction is no longer possible.

Autoencoders [21, 30] learn a latent representation from data by first summarizing the input image into a compressed latent vector $\boldsymbol{z} = \mathcal{E}_0(\boldsymbol{x})$ through an encoder. Then, the original image can be recovered from the latent via the decoder $\hat{\boldsymbol{x}} = \mathcal{D}_0(\boldsymbol{z})$ such that $\boldsymbol{x} \approx \hat{\boldsymbol{x}}$. As the latent space of autoencoders contains only the relevant information of data, it is a more natural space to quantify the loss of information due to the degradation than the image domain.

In particular, assume that we have access to the latent representation of clean images $\boldsymbol{z}_0 = \mathcal{E}_0(\boldsymbol{x}_0)$, $\boldsymbol{z}_0 \in \mathbb{R}^d$, for instance from a pre-trained autoencoder. We propose a *severity encoder* $\hat{\mathcal{E}}_\theta$ that achieves two objectives simultaneously: (1) it can predict the latent representation of a clean image, given a noisy and degraded observation and (2) it can quantify the error in its own latent

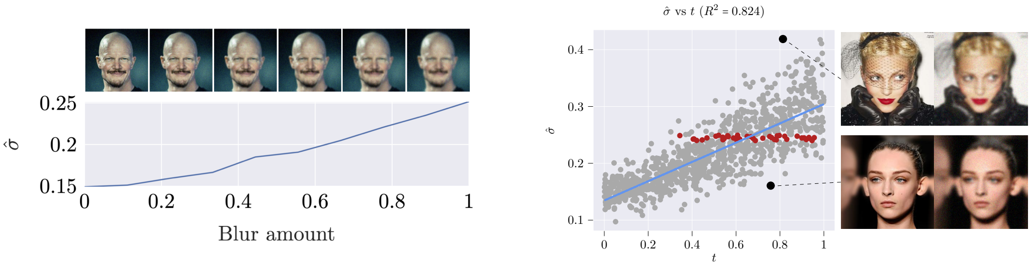

Figure 3: Left: given a ground truth image corrupted by varying amount of blur, $\hat{\sigma}$ is an increasing function of the blur amount. Right: Blur amount (t) vs. predicted degradation severity ($\hat{\sigma}$). Outliers indicate that the predicted degradation severity is not solely determined by the amount of blur. Red dots indicate samples with a wide range of degradations, but similar predicted severities.

estimation. We denote $\hat{\mathcal{E}}_{\boldsymbol{\theta}}(\boldsymbol{y}) = (\hat{\boldsymbol{z}},\ \hat{\sigma})$ with $\hat{\boldsymbol{z}} \in \mathbb{R}^d$ the estimate of $\boldsymbol{z}_0$ and $\hat{\sigma} \in \mathbb{R}$ the estimated degradation severity to be specified shortly. We use the notation $\hat{\mathcal{E}}_{\boldsymbol{z}}(\boldsymbol{y}) = \hat{\boldsymbol{z}}$ and $\hat{\mathcal{E}}_{\sigma}(\boldsymbol{y}) = \hat{\sigma}$ for the two conceptual components of our model, however in practice a single architecture is used to represent $\hat{\mathcal{E}}_{\theta}$. The first objective can be interpreted as image reconstruction in the latent space of the autoencoder: for $\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{n}$ and $\boldsymbol{z}_0 = \mathcal{E}_0(\boldsymbol{x})$, we have $\hat{\mathcal{E}}_{\boldsymbol{z}}(\boldsymbol{y}) = \hat{\boldsymbol{z}} \approx \boldsymbol{z}_0$. The second objective captures the intuition that recovering $\boldsymbol{z}_0$ from $\boldsymbol{y}$ exactly may not be possible, and the prediction error is proportional to the loss of information about $\boldsymbol{x}$ due to the corruption. Thus, even though the predicted latent $\hat{\boldsymbol{z}}$ might be away from the true $\boldsymbol{z}_0$, the encoder quantifies the uncertainty in its own prediction. More specifically, we make the assumption that the prediction error in latent space can be modeled as zero-mean *i.i.d.* Gaussian. That is, $\boldsymbol{e}(\boldsymbol{y}) = \hat{\boldsymbol{z}} - \boldsymbol{z}_0 \sim \mathcal{N}(\boldsymbol{0}, \sigma_*^2(\boldsymbol{y})\mathbf{I})$ and we interpret the variance in prediction error σ_*^2 as the measure of degradation severity. We optimize the joint objective

$$\mathbb{E}_{\boldsymbol{x}_0\sim q_0(\boldsymbol{x}_0),\boldsymbol{y}\sim\mathcal{N}(\mathcal{A}(\boldsymbol{x}_0),\sigma_y^2\mathbf{I})}\left[\left\|\boldsymbol{z}_0-\hat{\mathcal{E}}_{\boldsymbol{z}}(\boldsymbol{y})\right\|^2+\lambda_\sigma\left\|\bar{\sigma}^2(\boldsymbol{y},\boldsymbol{z}_0)-\hat{\mathcal{E}}_\sigma(\boldsymbol{y})\right\|^2\right]:=L_{lat.rec.}+\lambda_\sigma L_{err.},\qquad(1)$$

with $\boldsymbol{z}_0 = \mathcal{E}_0(\boldsymbol{x}_0)$ for a fixed, pre-trained encoder $\mathcal{E}_0$ and $\bar{\sigma}^2(\boldsymbol{y},\boldsymbol{z}_0) = \frac{1}{d-1}\sum_{i=1}^{d}(\boldsymbol{e}^{(i)} - \frac{1}{d}\sum_{j=1}^{d}\boldsymbol{e}^{(j)})^2$ is the sample variance of the prediction error estimating σ_*^2. Here $\lambda_\sigma > 0$ is a hyperparameter that balances between reconstruction accuracy ($L_{lat.rec.}$) and error prediction performance ($L_{err.}$). Training the severity encoder is fast, as one can fine-tune the pre-trained $\mathcal{E}_0$.

Sample-adaptive inference – Diffusion-based inverse problem solvers synthesize missing data that has been destroyed by the degradation process through diffusion. As shown in Figure 2, depending on the amount of missing information (easy vs. hard samples), the optimal number of diffusion steps may greatly vary. We aim to automatically find this "sweet spot" on a sample-by-sample basis.

We find the time index i_{start} in the latent diffusion process at which the signal-to-noise ratio (SNR) matches the SNR predicted by the severity encoder. Assume that the latent diffusion process is specified by the conditional distribution $q_i(\boldsymbol{z}_i|\boldsymbol{z}_0) \sim \mathcal{N}(a_i\boldsymbol{z}_0, b_i^2\mathbf{I})$, where a_i and b_i are determined by the specific sampling method. We also have $\hat{\boldsymbol{z}} \sim \mathcal{N}(\boldsymbol{z}_0, \sigma_*^2(\boldsymbol{y})\mathbf{I})$, where we estimate σ_*^2 by $\hat{\mathcal{E}}_\sigma(\boldsymbol{y})$. Then, SNR matching gives us the starting index

$$i_{start}(\boldsymbol{y}) = \arg\min_{i\in[1,2,..,N]}\left|\frac{a_i^2}{b_i^2} - \frac{1}{\hat{\mathcal{E}}_\sigma(\boldsymbol{y})}\right|$$

Thus, we start reverse diffusion from $\hat{\boldsymbol{z}}$ provided by the severity encoder and progressively denoise it using a pre-trained LDM, where the length of the sampling trajectory is directly determined by the predicted severity of the degraded example. Finally, we encourage data consistency through a latent space variant of DPS [4] (see details in Appendix C), which we call LDPS.

3 Experiments

We perform experiments on CelebA-HQ (256×256) [16]. For comparisons with image domain score models we test on FFHQ [17]. We investigate two degradations. Varying blur, fixed noise: we apply Gaussian blur with kernel size of 61 and sample kernel standard deviation uniformly on $[0, 3]$, where 0 corresponds to no blurring. Nonlinear blur, fixed noise: we deploy GOPRO motion blur [39]. This is a nonlinear forward model due to the camera response function. We randomly sample nonlinear blur kernels. We add Gaussian noise with standard deviation of 0.05 in both tasks.

	Gaussian Deblurring				Non-linear Deblurring			
Method	PSNR(↑)	SSIM(↑)	LPIPS(↓)	FID(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	FID(↓)
Flash-Diffusion (ours)	29.16	0.8191	**0.2241**	**29.46**	27.22	0.7705	**0.2694**	**36.92**
AE	29.43	0.8366	0.2668	58.47	27.15	0.7824	0.3351	73.81
SwinIR [25]	**30.71**	**0.8598**	0.2399	59.07	**27.66**	**0.7964**	0.3072	62.11
DPS	28.35	0.7806	0.2470	55.17	22.82	0.6247	0.3603	72.20
CCDF-DPS [6]	30.02	0.8365	0.2324	50.62	26.98	0.7445	0.2840	56.92
CCDF-L	29.55	0.8377	0.2346	49.06	27.25	0.7793	0.2833	55.85

Table 1: Experimental results on the FFHQ test split.

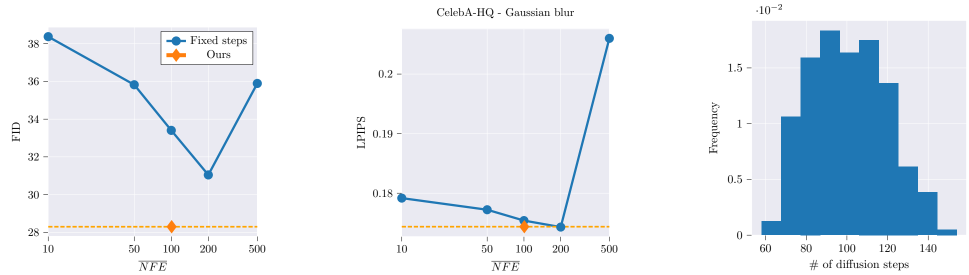

Figure 4: Left and center: Average # of reverse diffusion steps in our algorithm vs. CCDF-L with fixed number of steps. Right: Histogram of predicted # of reverse diffusion steps for our algorithm.

Comparison methods – We compare our method, with SwinIR [25], DPS [4], and CCDF [6] with two variants: (1) CCDF-DPS: we replace the projection-based data consistency method with DPS to facilitate nonlinear forward models and (2) CCDF-L: we deploy CCDF in latent space using the same LDM as for our method and we replace the data consistency step with LDPS . For all CCDF-variants we use the SwinIR reconstruction as initialization. We show results of decoding our severity encoder's latent estimate $\hat{z}$ directly, denoted by AE (autoencoded). Further details are in Appendix D and E.

Severity encoding – First, we isolate the effect of degradation on $\hat{\sigma}$ (Fig. 3, left). We fix the clean image and apply increasing amount of Gaussian blur. We observe that $\hat{\sigma}$ is an increasing function of the blur amount applied to the image. This implies that the severity encoder learns to capture the amount of information loss caused by the degradation.

Next, we investigate the relation between $\hat{\sigma}$ and the corruption level (Fig. 3, right). We parameterize the corruption level by t, and vary the blur kernel width linearly for $t \in (0, 1)$. We observe that $\hat{\sigma}$ strongly correlates with t. However, the existence of outliers suggests that factors other than the corruption level may also contribute to the predicted severities. The bottom image is predicted to be *surprisingly easy*, as other images of the same corruption level are typically assigned higher predicted severities. This sample has a lack of fine details and textures and shares common features with samples in the training set. On the other hand, the top image is considered *surprisingly difficult*, as it contains unexpected features and high-frequency details that are uncommon in the dataset.

Sample-adaptive reconstruction – We observe that Flash-Diffusion consistently outperforms other diffusion-based solvers in terms of LPIPS and FID (Table 1). SwinIR achieves higher PSNR and SSIM as diffusion methods, however the reconstructions lack detail. This phenomenon is due to the perception-distortion trade-off [3]: improving perceptual image quality is fundamentally at odds with distortion metrics. Visual comparison of samples can be found in Appendix H.

Efficiency of the method – We compare Flash-Diffusion to CCDF-L across various (fixed) number of reverse diffusion steps (Fig. 4). Our adaptive method achieves the best FID across any fixed number of steps and it achieves near-optimal LPIPS with often 2× less average number of diffusion steps. The predicted diffusion steps are spread around the mean and not closely concentrated, further highlighting the adaptivity of Flash-Diffusion. Results on nonlinear blur can be found in Appendix F.

4 Conclusions

In this work, we highlight that the difficulty of solving an inverse problem may vary greatly on a sample-by-sample basis, depending on the ground truth signal structure, the applied corruption, the model, the training set and the complex interactions between these factors. We propose Flash-Diffusion, a sample-adaptive method that predicts the degradation severity of corrupted signals, and utilizes this estimate to automatically tune the compute allocated to reconstruct the sample. We experimentally demonstrate that the proposed technique achieves performance on par with state-of-the-art diffusion-based reconstruction methods, but with greatly improved compute efficiency.

References

[1] Diego Ardila, Atilla P Kiraly, Sujeeth Bharadwaj, Bokyung Choi, Joshua J Reicher, Lily Peng, Daniel Tse, Mozziyar Etemadi, Wenxing Ye, Greg Corrado, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, 25(6):954–961, 2019.

[2] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018.

[4] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[5] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *arXiv preprint arXiv:2206.00941*, 2022.

[6] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12413–12422, 2022.

[7] Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated mri. *Medical Image Analysis*, 80:102479, 2022.

[8] Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

[9] Zalan Fabian, Berk Tinaz, and Mahdi Soltanolkotabi. Diracdiffusion: Denoising and incremental reconstruction with assured data-consistency. *arXiv preprint arXiv:2303.14353*, 2023.

[10] Berthy T Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L Bouman, and William T Freeman. Score-based diffusion models as principled priors for inverse imaging. *arXiv preprint arXiv:2304.11751*, 2023.

[11] Davis Gilton, Gregory Ongie, and Rebecca Willett. Deep equilibrium architectures for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:1123–1133, 2021.

[12] Paul Hand, Oscar Leong, and Vlad Voroninski. Phase retrieval under a generative prior. *Advances in Neural Information Processing Systems*, 31, 2018.

[13] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022.

[14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[15] Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34:13242–13254, 2021.

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv:1710.10196 [cs, stat]*, 2018.

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[18] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022.

[19] Bahjat Kawar, Jiaming Song, Stefano Ermon, and Michael Elad. Jpeg artifact correction using denoising diffusion restoration models. *arXiv preprint arXiv:2209.11888*, 2022.

[20] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[22] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[23] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018.

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[25] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using Swin Transformer. *arXiv:2108.10257*, 2021.

[26] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Refusion: Enabling large-size realistic image restoration with latent-space diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1680–1691, 2023.

[27] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[28] Gregory Ongie, Ajil Jalal, Christopher A. Metzler Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 2020.

[29] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[30] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[32] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alexandros G Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *arXiv preprint arXiv:2307.00619*, 2023.

[33] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[35] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image Super-Resolution via Iterative Refinement. *arXiv:2104.07636 [cs, eess]*, 2021.

[36] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005*, 2021.

[37] Anuroop Sriram, Jure Zbontar, Tullie Murrell, Aaron Defazio, C. Lawrence Zitnick, Nafissa Yakubova, Florian Knoll, and Patricia Johnson. End-to-end variational networks for accelerated MRI reconstruction. In *Medical Image Computing and Computer Assisted Intervention*, pages 64–73, 2020.

[38] Jian Sun, Huibin Li, Zongben Xu, et al. Deep ADMM-Net for compressive sensing MRI. *Advances in Neural Information Processing Systems*, 29, 2016.

[39] Phong Tran, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11956–11965, 2021.

[40] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. *Advances in neural information processing systems*, 31, 2018.

[41] Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1828–1837, 2018.

A Training loss details

As detailed in Section 2.1, we train the severity encoder on the joint objective

$$L_{sev} = \mathbb{E}_{\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0), \boldsymbol{y} \sim \mathcal{N}(\mathcal{A}(\boldsymbol{x}_0), \sigma_y^2 \mathbf{I})} \left[\left\| \boldsymbol{z}_0 - \hat{\mathcal{E}}_{\boldsymbol{z}}(\boldsymbol{y}) \right\|^2 + \lambda_\sigma \left\| \bar{\sigma}^2(\boldsymbol{y}, \boldsymbol{z}_0) - \hat{\mathcal{E}}_\sigma(\boldsymbol{y}) \right\|^2 \right] := L_{lat.rec.} + \lambda_\sigma L_{err.},$$

where $L_{lat.rec.}$ encourages accurate reconstruction in the latent space, whereas $L_{err.}$ imposes penalty on the latent error estimation. We empirically observe, that even small loss values of $L_{lat.rec.}$ (that is fairly good latent reconstruction) may correspond to visible reconstruction error in image domain as semantically less meaningful features are often not captured in the latent representation. Therefore, we utilize an extra loss term that imposes image domain consistency with the ground truth image in the form

$$L_{im.rec.} = \mathbb{E}_{\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0), \boldsymbol{y} \sim \mathcal{N}(\mathcal{A}(\boldsymbol{x}_0), \sigma_y^2 \mathbf{I})} \left[\left\| \boldsymbol{x}_0 - \mathcal{D}_0(\hat{\mathcal{E}}_{\boldsymbol{z}}(\boldsymbol{y})) \right\|^2 \right],$$

resulting in the final combined loss

$$L_{combined} = L_{lat.rec.} + \lambda_\sigma L_{err.} + \lambda_{im.} L_{im.rec.},$$

with $\lambda_{im.} \geq 0$ hyperparameter.

B Noise correction

Even though we assume that the prediction error in latent space is *i.i.d.* Gaussian in order to quantify the estimation error by a single scalar, in practice the error often has some structure. This can pose a challenge for the score model, as it has been trained to remove isotropic Gaussian noise. We observe that it is beneficial to mix $\hat{\boldsymbol{z}}$ with some *i.i.d. correction noise* in order to suppress structure in the prediction error. In particular, we initialize the reverse process by

$$\boldsymbol{z}_{start} = \sqrt{1 - c\hat{\sigma}^2}\hat{\boldsymbol{z}} + \sqrt{c\hat{\sigma}^2}\boldsymbol{\varepsilon}, \ \ \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where $c \geq 0$ is a tuning parameter.

C Maintaining consistency with measurements

Maintaining consistency with the measurements is non-trivial in the latent domain, as common projection-based approaches are not applicable directly in latent space. We propose Latent Diffusion Posterior Sampling (LDPS), a variant of diffusion posterior sampling [4] that guides the latent diffusion process towards data consistency in the original data space. In particular, by applying Bayes' rule the score of the posterior in latent space can be written as

$$\nabla_{\boldsymbol{z}_t} \log q_t(\boldsymbol{z}_t | \boldsymbol{y}) = \nabla_{\boldsymbol{z}_t} \log q_t(\boldsymbol{z}_t) + \nabla_{\boldsymbol{z}_t} \log q_t(\boldsymbol{y} | \boldsymbol{z}_t).$$

The first term on the r.h.s. is simply the unconditional score that we already have access to as pre-trained LDMs. As $q_t(\boldsymbol{y}|\boldsymbol{z}_t)$ cannot be written in closed form, following DPS we use the approximation $\nabla_{\boldsymbol{z}_t} \log q_t(\boldsymbol{y}|\boldsymbol{z}_t) \approx \nabla_{\boldsymbol{z}_t} \log q_t(\boldsymbol{y}|\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t))$, where $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t) = \mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t]$ is the posterior mean of $\boldsymbol{z}_0$, which is straightforward to estimate from the score model via Tweedie's formula. This form is similar to PSLD in [32], but without the "gluing" objective. As $\boldsymbol{y} \sim \mathcal{N}(\mathcal{A}(\mathcal{D}_0(\boldsymbol{z}_0)), \sigma_y^2 \mathbf{I})$, we approximate the score of the likelihood as

$$\nabla_{\boldsymbol{z}_t} \log q_t(\boldsymbol{y}|\boldsymbol{z}_t) \approx -\frac{1}{2\sigma_y^2} \nabla_{\boldsymbol{z}_t} \left\| \mathcal{A}(\mathcal{D}_0(\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t))) - \boldsymbol{y} \right\|^2. \tag{2}$$

LDPS is a general approach to impose data consistency in the latent domain that works with noisy and potentially nonlinear inverse problems.

D Training details

Here we provide additional details on the training setup and hyperparameters.

Dataset – We perform experiments on CelebA-HQ (256×256) [16] where we match the training and validation splits used to train LDMs in [31], and set aside $1k$ images from the validation split for testing. For comparisons involving image domain score models we test on FFHQ [17], as pre-trained image-domain score models have been trained on the complete CelebA-HQ dataset, unlike LDMs.

Degradations – We investigate two degradations of diverging characteristics. Varying blur, fixed noise: we apply Gaussian blur with kernel size of 61 and sample kernel standard deviation uniformly on $[0, 3]$, where 0 corresponds to no blurring. We add Gaussian noise to images in the $[0, 1]$ range with noise standard deviation of 0.05. Nonlinear blur, fixed noise: we deploy GOPRO motion blur simulated by a neural network model from [39]. This is a nonlinear forward model due to the camera response function. We randomly sample nonlinear blur kernels for each image and add Gaussian noise with standard deviation 0.05.

Model architecture – In all experiments, we use an LDM model pre-trained on the CelebA-HQ dataset out of the box. We fine-tune the severity encoder from the LDM's pre-trained encoder. We obtain the degradation severity estimate $\hat{\sigma} \in \mathbb{R}^+$ from the latent reconstruction $\hat{\boldsymbol{z}} \in \mathbb{R}^d$ as

$$\hat{\sigma} = \frac{1}{d}\sum_{i=1}^{d}[Conv(\hat{\boldsymbol{z}})^2]_i,$$

where $Conv(\cdot)$ is a learned 1×1 convolution with d input and d output channels.

Training setup – We train severity encoders using Adam optimizer with batch size 28 and learning rate 0.0001 for about $200k$ steps until the loss on the validation set converges. We use Quadro RTX 5000 and Titan GPUs.

Hyperparameters – We scale the reconstruction loss terms with their corresponding dimension (d for $L_{lat.rec.}$ and n for $L_{im.rec.}$), which we find to be sufficient without tuning for $\lambda_{im.rec.}$. We tune λ_σ via grid search on $[0.1, 1, 10]$ on the varying Gaussian blur task and set to 10 for all experiments.

For latent diffusion posterior sampling, following [4] we scale the posterior score estimate with the data consistency error as

$$\nabla_{\boldsymbol{z}_t} \log q_t(\boldsymbol{z}_t|\boldsymbol{y}) \approx s_\theta(\boldsymbol{z}_t) - \eta_t \nabla_{\boldsymbol{z}_t} \left\|\mathcal{A}(\mathcal{D}_0(\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t))) - \boldsymbol{y}\right\|^2,$$

where

$$\eta_t = \frac{\eta}{\left\|\mathcal{A}(\mathcal{D}_0(\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t))) - \boldsymbol{y}\right\|},$$

and $\eta > 0$ is a tuning parameter. We perform grid search over $[0.5, 1.0, 1.5, 2.0]$ over a small subset of the validation set (100 images) and find $\eta = 1.5$ to work the best for all tasks.

Similarly, we tune the noise correction parameter c on the same validation subset by grid search over $[0.5, 0.8, 1.0, 1.1, 1.2, 1.5]$ and find $c = 1.2$ for Gaussian blur and $c = 1.1$ for nonlinear blur to work the best.

E Comparison method details

For all methods, we use the train and validation splits provided for CelebA and FFHQ in the GitHub repo of "Taming Transformers" [1]. For the test split, we subsample 1000 ids from the corresponding validation ids file. Specific ids we used will be available when the codebase is released. We provide the specifics for each comparison method next.

SwinIR: For both Gaussian blur and non-linear blur experiments, we train SwinIR using Adam optimizer with batch size 28 for 100 epochs. We use learning rate 0.0002 for the first 90 epochs and drop it by a factor of 10 for the remaining 10 epochs. We use Quadro RTX 5000 and Titan GPUs.

CCDF-DPS: We implement this method by modifying the official GitHub repo [2] of DPS [4]. Instead of projection based conditioning, we replace it with the DPS updates to handle noise in the measurements and non-linear degradations. As the initial estimate, we use the output of SwinIR model that we trained. We tune the data consistency step size ζ' and number of reverse diffusion steps N' by doing 2D grid search over $[0.5, 1.0, 2.0, 3.0, 5.0, 10.0, 20.0] \times [1, 10, 20, 50, 100, 200, 1000]$ on the small subset of validation split of FFHQ dataset (100 images) based on LPIPS metric. For

[1] `https://github.com/CompVis/taming-transformers/tree/master/data`

[2] `https://github.com/DPS2022/diffusion-posterior-sampling`

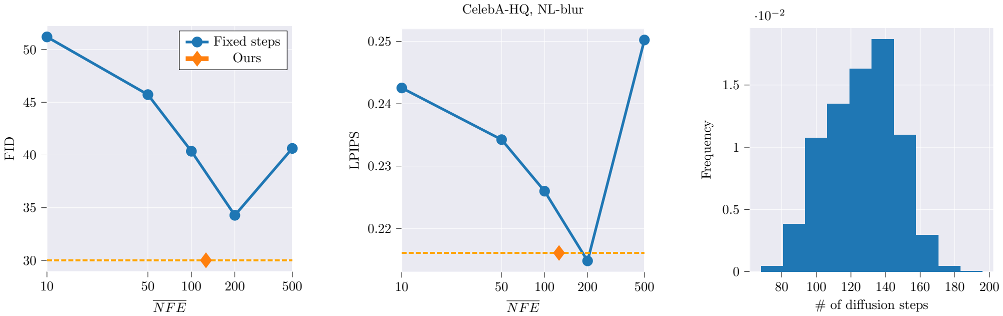

Figure 5: Comparison of adaptive reconstruction with fixed number of diffusion steps on the nonlinear blur task. Left and center: We plot the average number of reverse diffusion steps performed by our algorithm vs. CCDF-L with a fixed number of steps. Right: We plot the histogram of predicted number of reverse diffusion steps for our algorithm.

Gaussian blur, we find the optimal hyperparameters to be $(\zeta', N') = (5.0, 20)$. For non-linear blur optimal values are $(\zeta', N') = (3.0, 100)$.

CCDF-L: This method is the same as ours but with a fixed starting time and initial estimate provided by SwinIR model we trained. We tune the data consistency step size η and the number of reverse diffusion steps N' by doing grid search over $[0.5, 1.0, 1.5, 2.0] \times [20, 50, 100, 200]$ on the small subset of validation split of FFHQ (100 images) based on LPIPS metric. For varying blur experiments, we found the optimal value to be $(\eta, N') = (1.0, 100)$. For non-linear blur, we found it to be $(\eta, N') = (1.5, 200)$.

DPS: This method can be seen as a special case of CCDF-DPS where number of reverse diffusion steps is fixed to $N' = 1000$. From the same 2D grid search we performed for CCDF-DPS, we find the optimal data consistency step size ζ' to be 5.0 for Gaussian blur, 0.5 for non-linear blur.

Autoencoded (AE): We use the latent at severity encoders output and decode it without reverse diffusion to get the reconstruction.

F Further efficiency results

We provide further support for the efficiency of our proposed method in the nonlinear blur task in Figure 5.

G Robustness against forward model mismatch

Our method relies on a severity encoder that has been trained on paired data of clean and degraded images under a specific forward model. We simulate a mismatch between the severity encoder fine-tuning operator and test-time operator in order to investigate the robustness of our technique with respect to forward model perturbations. In particular, we run the following experiments to assess the test-time shift: 1) we train the encoder on Gaussian blur and test on non-linear blur and 2) we train the encoder on non-linear blur and test on Gaussian blur. The results on the FFHQ test set are in Table 2. We observe minimal loss in performance when non-linear blur encoder is used for reconstructing images corrupted by Gaussian blur. For the non-linear deblurring task, using Gaussian blur encoder results in a more significant drop in the performance, while still providing acceptable reconstructions. These results are expected, as Gaussian blur can be thought of as a special case of the non-linear blur model we consider. Therefore even when the encoder is swapped, it can provide meaningful mean and error estimation. However, the Gaussian blur encoder has never been trained on images corrupted by non-linear blur. As such, the mean estimate is worse, resulting in a larger performance drop. Note that we did not re-tune the hyper-parameters in these experiments and doing so may potentially alleviate the loss in performance.

	Gaussian Deblurring				Non-linear Deblurring			
Method	PSNR(↑)	SSIM(↑)	LPIPS(↓)	FID(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	FID(↓)
Ours + Gaussian blur encoder	29.16	0.8191	0.2241	29.467	25.36	0.7238	0.3416	54.90
Ours + NL blur encoder	28.96	0.8129	0.2362	30.34	27.22	0.7705	0.2694	36.92

Table 2: Robustness experiments on the FFHQ test split.

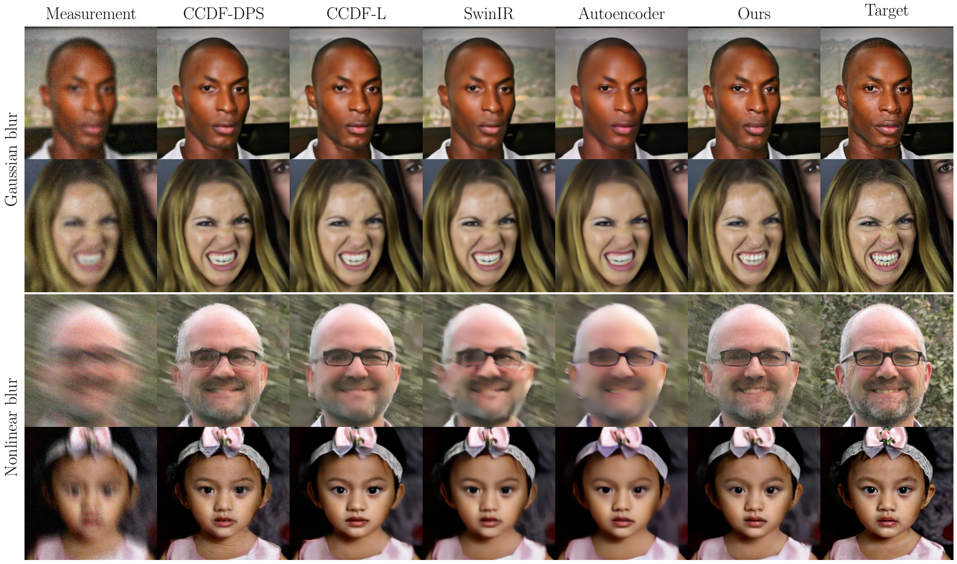

Figure 6: Visual comparison of FFHQ reconstructions under varying levels of Gaussian blur (top 2 rows) and nonlinear motion blur (bottom 2 rows), both with additive Gaussian noise ($\sigma = 0.05$).

H Visual comparison of reconstructions

We perform visual comparison of reconstructed samples in Figure 6. We observe that Flash-Diffusion can reconstruct fine details, significantly improving upon the autoencoded reconstruction used to initialize the reverse diffusion process. SwinIR produces reliable reconstructions, but with less details compared to diffusion-based methods. Moreover, note that diffusion-based solvers with fixed number of diffusion steps tend to under-diffuse (see 2nd row, lack of details) or over-diffuse (4th row, high-frequency artifacts) leading to subpar reconstructions.

I Limitations

We identify the following limitations of our framework.

1. The assumption of i.i.d. Gaussian prediction error provides a simple way to estimate the severity, however does not necessarily hold in practice. We believe that more realistic error models can further improve our technique, which we leave for future work.
2. The proposed severity estimation method estimates a scalar value for the latent embedding. One reason we choose to estimate a scalar value is to appeal to the intuition of a simple severity measure. We also want to have a clear mapping between the error and the time step to initiate the reverse diffusion. We believe that using a vector valued error estimate which corresponds to having a diagonal covariance estimate would add more flexibility to our method.
3. The proposed reconstruction method requires degraded-clean image pairs for fine-tuning the severity encoder. Fine-tuning has to be performed separately for each degradation, thus the method is less flexible than DPS and similar diffusion solvers. However, we argue that the fine-tuning step has fairly low cost.
4. Latent diffusion posterior sampling has some additional compute cost compared to DPS due to differentiating through the decoder. Maintaining data-consistency through only latent space information would alleviate this issue. However, it is not clear how to do this for arbitrary degredations, hence we leave it for future work.