Maximum A Posteriori Training of Diffusion Models for Image Restoration

1st Samuel Willingham
Mid Sweden University
Sundsvall, Sweden
samuel.willingham@miun.se

2nd Mårten Sjöström
Mid Sweden University
Sundsvall, Sweden
marten.sjostrom@miun.se

3rd Christine Guillemot
Inria Rennes
Rennes, France
christine.guillemot@inria.fr

***Abstract*—Inverse problems involve reconstructing clean images from degraded observations. Maximum a Posteriori (MAP) estimation reconstructs the most probable source image from noisy measurements. When combined with Plug-and-Play (PnP) priors defined by an image denoising algorithm, MAP estimation yields high-quality reconstructions. In contrast, Diffusion Models (DMs) address inverse problems by sampling from the posterior distribution using score functions trained on images perturbed by Gaussian noise. Prior work reformulated diffusion sampling as Deep Equilibrium (DEQ) models but did not fine-tune DMs for inverse problems. This work introduces MaximUm a PostEriori Training (MUPET), a framework that leverages PnP gradient descent to enable DEQ fine-tuning of DMs on inverse problems. By refining a generative prior at the fixed-point of MAP estimation, MUPET enhances image restoration via posterior sampling while maintaining quality when sampling from the prior.**

***Index Terms*—Inverse Problems, Diffusion Models, Posterior Sampling, Generative Methods, Deep Equilibrium Models**

I. Introduction

Inverse problems in imaging deal with the reconstruction of lost information from corrupted images. Diffusion Models (DMs) have recently been used to generate reconstructions with high perceptual quality [1]–[3]. Here, we investigate whether DM-based reconstruction benefits from Deep Equilibrium (DEQ)-based training on inverse problems, an approach that has improved Plug-and-Play (PnP) reconstruction methods [4]–[8].

For this paper, we consider inverse problems (e.g., super-resolution or completion) of the form

$$\boldsymbol{y} = \boldsymbol{A}\hat{\boldsymbol{x}} + \sigma_n \boldsymbol{\varepsilon}, \tag{1}$$

where $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ is the ground-truth image, sampled from the prior distribution p_{data}. Here, $\boldsymbol{A} \in \mathbb{R}^{d \times d'}$ is the degradation matrix and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ is Gaussian noise scaled by standard deviation $\sigma_n > 0$, where $d, d' \in \mathbb{N}$. $\mathcal{N}(0, \boldsymbol{I})$ denotes a multivariate Gaussian distribution with mean 0 and covariance matrix $\boldsymbol{I}$, where the dimensionality is clear from the context. For this work, we consider algorithms that leverage knowledge of the degradation matrix $\boldsymbol{A}$ as well as the noise-level σ_n.

Inverse problems as in (1) are commonly ill-posed and need regularization involving prior knowledge. When searching for the image that maximizes the posterior $p(\cdot \mid \boldsymbol{y})$, this leads to the Maximum a Posteriori (MAP) estimation problem

$$\hat{\boldsymbol{x}}_{\text{MAP}} = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sigma_n^2 R(\boldsymbol{x}), \tag{2}$$

where $\|\cdot\|_2$ is the Euclidean 2-norm. The minimization in (2) balances data fidelity $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ with the regularization term $R(\boldsymbol{x}) := -\log p_{\text{data}}(\boldsymbol{x})$, representing the prior distribution p_{data}.

Plug-and-Play (PnP) methods [9], [10] solve MAP estimation problems via the use of pre-trained or hand-crafted priors for the regularization in iterative algorithms. One such algorithm is the PnP gradient-descent (GD) [9]

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) := \boldsymbol{x} - \eta \left(\boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}) + \sigma_n^2 G_{\boldsymbol{\theta}}(\boldsymbol{x}) \right), \tag{3}$$

for input $\boldsymbol{x} \in \mathbb{R}^d$ with step-size $\eta > 0$. $G_{\boldsymbol{\theta}}$ serves as prior regularization with parameters $\boldsymbol{\theta}$, representing $-\nabla \log p_{\text{data}}$ [8], [9], the gradient of the regularizer R in (2). Commonly, $G_{\boldsymbol{\theta}}$ is trained in the context of a Gaussian denoiser, leading to excellent reconstructions [9]. For PnP approaches, $G_{\boldsymbol{\theta}}$ is trained independently of any inverse problem. Thus, the resulting PnP is highly general but can be further improved via DEQ models [4]–[8] by training the whole iterative method on its fixed-point, i.e. the MAP estimate. Furthermore, refining a PnP algorithm via DEQ on a range of inverse problems leads to improved multi-task reconstruction performance [7], [8].

Diffusion Models (DMs) [1]–[3], instead, sample from the posterior $p(\cdot \mid \boldsymbol{y})$ of an inverse problem. They are trained to reverse a process that progressively corrupts a clean image with Gaussian noise, allowing for an incremental reduction of the effective noise in an image (i.e. reducing noise-level $\sigma > 0$ step-by-step). Originally, DMs were proposed to generate new samples from p_{data} [1], but they were quickly adopted for the solution of inverse problems.

Inspired by PnP approaches, algorithms like DiffPIR [2] condition the sampling process on the observation $\boldsymbol{y}$, generating samples from $p(\cdot \mid \boldsymbol{y})$ by using the score network as a deep generative prior. Since perceptual quality is central to inverse problems, performance is often measured via Learned Perceptual Image Patch Similarity (LPIPS) [11], which better aligns with human-perceived image quality than traditional pixel-wise metrics. Using LPIPS as the evaluation metric, Diff-

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956770, as well as the IMMERSE project - part of Interreg Aurora.

PIR achieves state-of-the-art perceptual quality, outperforming a PnP half-quadratic-splitting algorithm [2].

Despite the impressive performance of DMs and their similarities to PnP methods, they have not yet been DEQ fine-tuned for inverse problems. While prior work [12] has used DEQ models for parallel sampling with root solvers and model inversion, DEQ fine-tuning of DMs on inverse problems is novel. To address this, we propose **MaximUm a PostEriori Training (MUPET)**[1]:

- Refines the score network using the fixed-point of MAP estimation, bridging the gap between generative modeling and MAP estimation.
- Reduces LPIPS by up to 27% in restoration tasks while maintaining performance when sampling from the prior.
- Maintains inference time by serving as a drop-in replacement for the pre-trained model.

II. Theory and Related Works

To contextualize MUPET, we first review the theoretical foundations of DEQ models and DMs in the context of inverse problems. We discuss how DEQ principles enable fine-tuning of iterative solvers and motivate our choice of the variance exploding parametrization for MUPET.

A. Deep Equilibrium Models

Deep Equilibrium (DEQ) models [4], [5] enable the fine-tuning of an iterative procedure at its fixed-point (FP), avoiding the need for explicit backpropagation through iterations. DEQ models have been used to fine-tune PnP methods, improving both single-task [6] and multi-task [7], [8] reconstruction performance. To formalize this, we look at the PnP GD in (3), with (assuming convergence) FP $\boldsymbol{x}_g$, such that $g_{\boldsymbol{\theta}}(\boldsymbol{x}_g) = \boldsymbol{x}_g$. The gradient of a given loss function l can then be expressed as

$$\frac{\delta l(\hat{\boldsymbol{x}}, \boldsymbol{x}_g)}{\delta \boldsymbol{\theta}} = \frac{\delta l(\hat{\boldsymbol{x}}, \boldsymbol{x}')}{\delta \boldsymbol{x}'}\bigg|_{\boldsymbol{x}'=\boldsymbol{x}_g} \boldsymbol{J}^{-1} \frac{\delta g_{\boldsymbol{\theta}}(\boldsymbol{x}')}{\delta \boldsymbol{\theta}}\bigg|_{\boldsymbol{x}'=\boldsymbol{x}_g}, \quad (4)$$

where $\boldsymbol{J} := \boldsymbol{I} - \frac{\delta g_{\boldsymbol{\theta}}(\boldsymbol{x}')}{\delta \boldsymbol{x}'}|_{\boldsymbol{x}'=\boldsymbol{x}_g}$ is the Jacobian [4]. Note that the above expression is independent of the path taken. Furthermore, once the FP is obtained, approximating $\boldsymbol{J}$ as the identity further simplifies backpropagation while still yielding a direction of descent [5].

MUPET leverages this Jacobian-free backpropagation to fine-tune score networks across multiple inverse problems within a PnP GD framework. Since both PnP methods and DMs estimate the score function of an image distribution, MUPET uses DEQ models to fine-tune the score function in a PnP GD algorithm.

B. Diffusion Models for Inverse Problems

After looking at how DEQ models enable the fine-tuning of PnP methods, we now examine diffusion models (DMs): a framework that addresses inverse problems via generative sampling rather than MAP estimation. DMs aim at reversing a process that progressively corrupts clean images with Gaussian noise. For the reversal, a score network $s_{\boldsymbol{\theta}} : \mathbb{R}^d \times \mathbb{R}_{>0} \to \mathbb{R}^d$, maps a noisy image and the corresponding noise-level $\sigma > 0$ to an estimate of the added noise [1]. Training leverages Denoising Score Matching (DSM), minimizing

$$L_\sigma^{\text{DSM}} := \|\sigma s_{\boldsymbol{\theta}}(\boldsymbol{z}, \sigma) + \boldsymbol{\varepsilon}\|_2^2, \quad (5)$$

with $\boldsymbol{z} := \hat{\boldsymbol{x}} + \sigma\boldsymbol{\varepsilon}$, leading to $s_{\boldsymbol{\theta}}(\cdot, \sigma) \approx \nabla \log p_\sigma(\cdot)$. Here, $p_\sigma(\boldsymbol{x}') := \int g_\sigma(\boldsymbol{x}' - \boldsymbol{x}) p_{\text{data}}(\boldsymbol{x}) \,\mathrm{d}\,\boldsymbol{x}$ is the convolution of p_{data} with a Gaussian kernel g_σ with standard deviation $\sigma > 0$ [1].

Diffusion-based approaches are mainly presented via the variance preserving (VP) and the variance exploding (VE) frameworks. While they are mathematically equivalent and can be transposed to one-another [3], the parametrization can affect training and inference. In the variance preserving (VP) formulation [1], [13], noise is added while simultaneously scaling the image to keep variance constant. The trained network does not directly represent the score $\nabla \log p_{\text{data}}$ but instead learns a rescaled version. By contrast, the VE formulation adds increasing amounts of Gaussian noise without scaling the signal. The VE framework utilizes the DSM introduced in (5) and directly estimates the score function, eliminating the need for rescaling or re-weighting during inference. Because the score network directly estimates $\nabla \log p_{\text{data}}$, the VE framework presents a natural choice for the use in combination with the PnP GD used in MUPET.

Adaptations of diffusion-based approaches to inverse problems often draw inspiration from PnP methods. For example, DiffPIR [2] modifies the PnP half-quadratic-splitting algorithm for posterior sampling, achieving high reconstruction quality. While previous work has used DEQ models for inference improvements, such as parallel sampling and model inversion [12], MUPET leverages DEQ models to fine-tune the score network during training. MUPET refines the network using actual reconstruction errors rather than only relying on Gaussian-perturbed images, bridging the gap between generative modeling and MAP estimation without changing the inference procedure.

III. Maximum a Posteriori Training of Diffusion Models

This work investigates parallels between DMs and MAP estimation, motivating the use of a score network in a PnP GD algorithm. We introduce MUPET, which fine-tunes a DM via the fixed-point of PnP GD.

Optimizing L_σ^{DSM} in (5) for any fixed σ is equivalent to optimizing the denoising loss

$$L_\sigma^{\text{den}} := \|\boldsymbol{z} + \sigma^2 s_{\boldsymbol{\theta}}(\boldsymbol{z}, \sigma) - \hat{\boldsymbol{x}}\|_2^2, \quad (6)$$

for $\boldsymbol{z} := \hat{\boldsymbol{x}} + \sigma\boldsymbol{\varepsilon}$. Because this loss uses the sum of squared errors, the loss ensures $\boldsymbol{z} + \sigma^2 s_{\boldsymbol{\theta}}(\boldsymbol{z}, \sigma) \approx \mathbb{E}(\boldsymbol{x} \mid \boldsymbol{z})$. Also, by Tweedie's formula [14],

$$\mathbb{E}(\boldsymbol{x} \mid \boldsymbol{z}) = \boldsymbol{z} + \sigma^2 \nabla \log p_\sigma(\boldsymbol{z}), \quad (7)$$

we can now write (as is done in [1])

$$s_{\boldsymbol{\theta}}(\boldsymbol{z}, \sigma) \approx \nabla \log p_\sigma(\boldsymbol{z}). \quad (8)$$

[1] Our code is available at https://github.com/Realistic3D-MIUN/MUPET

This makes the score network a good candidate for the regularizer in (3), as $\nabla \log p_\sigma \to \nabla \log p_{\text{data}}$ for $\sigma \to 0$. Thus, we propose setting $G_{\boldsymbol{\theta}} := -s_{\boldsymbol{\theta}}(\cdot, \sigma_n)$ and using the resulting MAP estimate $\boldsymbol{x}_g$ via the loss

$$L^{\text{MUPET}} := \frac{\alpha}{\sigma_n^2} \|\boldsymbol{x}_g - \hat{\boldsymbol{x}}\|_2^2 + L_{\sigma_n}^{\text{DSM}} \tag{9}$$

to fine-tune $s_{\boldsymbol{\theta}}$. $\alpha > 0$ is a weighting parameter, but for this work we limit investigations to $\alpha = 1$ as both terms in (9) use the same underlying loss and concern problems with the same noise-level. We call the first term L^{DEQ}. It is scaled using σ_n to keep it in line with $L_{\sigma_n}^{\text{DSM}}$ and to ensure that larger noise-levels are not overly favored. L^{DEQ} allows training on the actual reconstruction error using the FP estimate $\boldsymbol{x}_g$ (found using Adam [15], following [8], [9]). To optimize $s_{\boldsymbol{\theta}}$, we employ Jacobian-free backpropagation. The resulting contribution of L^{DEQ} to the direction of descent is

$$\boldsymbol{p} := -2\alpha\eta(\boldsymbol{x}_g - \hat{\boldsymbol{x}})^T \frac{\delta s_{\boldsymbol{\theta}}(\boldsymbol{x}', \sigma_n)}{\delta \boldsymbol{\theta}}\Big|_{\boldsymbol{x}'=\boldsymbol{x}_g}. \tag{10}$$

We picked the squared 2-norm for L^{DEQ} to keep it in line with L^{DSM}, as the loss function that aligns best with MAP estimation in the sense of Bayes-optimality is the 0-1 loss and is ineffective for gradient-based training. The training pipeline can be seen in Algorithm 1.

```
Data: Training data $D \subset \mathbb{R}^d$, noise-levels $\Sigma \subset \mathbb{R}_{>0}$
      and set of degradations $\mathcal{A}$;
Result: Fine-tuned $s_{\boldsymbol{\theta}}$ ;
for a number of epochs do
    for all $\hat{x}$ from D do
        $(\sigma_n, \boldsymbol{A}) \leftarrow$ randomly picked from $\Sigma \times \mathcal{A}$;
        $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \sim \mathcal{N}(0, \boldsymbol{I})$;
        $\boldsymbol{y} \leftarrow A\hat{\boldsymbol{x}} + \sigma_n \boldsymbol{\varepsilon}_1$;
        Find FP $\boldsymbol{x}_g$ of GD procedure $g_{\boldsymbol{\theta}}$ ;
        $L^{\text{DEQ}} \leftarrow \frac{1}{\sigma_n^2} \|\boldsymbol{x}_g - \hat{\boldsymbol{x}}\|_2^2$;
        $L^{\text{DSM}} \leftarrow \|\sigma_n \cdot s_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}} + \sigma_n \boldsymbol{\varepsilon}_2, \sigma_n) + \boldsymbol{\varepsilon}_2\|_2^2$;
        $L \leftarrow L^{\text{DEQ}} + L^{\text{DSM}}$ ;
        Update parameters $\boldsymbol{\theta}$ via loss $L$ ;
    end
end
```

Algorithm 1: Training pipeline for MaximUm a PostEriori Training (MUPET).

By training $s_{\boldsymbol{\theta}}$ on a range of different noise-levels and degradations, MUPET steers $s_{\boldsymbol{\theta}}$ to be a general score network that is not tied to any single task. At the same time, it is updated on the actual reconstruction, meaning it is trained on more meaningful inputs than only images perturbed by Gaussian noise.

IV. Experiments

In this section, we evaluate MUPET by comparing it to the DSM-trained score function from [1]. We conduct experiments on CIFAR-10 [16] and LSUN Bedroom [17], assessing performance across several inverse problems. Here, we detail the training setup, inference procedure, and evaluation metrics.

A. Training

The DSM baseline is a noise-conditional score network (NCSN++) taken from [1]. MUPET uses the same architecture, initialized with the pre-trained DSM score function. The FPs of the PnP GD $g_{\boldsymbol{\theta}}$ (in image-space) were computed using the Adam optimizer [15], following [8] (default parameters, learning rate set to $\eta = 0.05$). Finding a more accurate FP takes longer but provides a more accurate gradient for backpropagation. For this work, iterations were terminated when the mean squared distance between two iterates fell below $5 \cdot 10^{-5}$, at the latest after 500 iterations. The latter only came into play in less than 10% of training iterations, meaning convergence was reached consistently. Each iteration started from an $\boldsymbol{x}_0$, where pixel values were randomly drawn from a Gaussian distribution with mean 0.5 and standard deviation 1. For both datasets, the pre-processing was taken from [1]. The loaded images were normalized to lie in $[0, 1]$, but outputs were left as is and not clipped.

Training Setup:

- **Epochs:** 100
- **Optimizer for network parameters $\boldsymbol{\theta}$:** Adam [15] (default parameters, learning rate 10^{-5})
- **CIFAR-10 [16]:** 50K images, in batches of 128 images
- **LSUN Bedroom [17]:** 500 iterations of 8-image batches per epoch
- **Hardware:** A single Nvidia A40 GPU
- **Runtime:** ∼1.3–1.35h/epoch

At each training iteration, there was an equal chance of picking an inverse problem from:

- **Super-resolution:** The scale factor was uniformly chosen from $\{1, 2, 3, 4, 5, 6, 7, 8\}$.
- **Deblurring:** The standard deviation for the Gaussian blur kernel was uniformly sampled from $[0, 8]$.
- **Pixel-wise completion:** A masking factor was drawn from $[0, 1]$, determining the probability of each pixel being masked.
- **Block-wise completion:** The block's width and height were independently sampled from $[0, 32]$, with a randomly placed mask.
- **Denoising:** No additional degradation was applied.

Also, at each iteration, we chose a different σ_n, by first choosing an r uniformly from $[0, 1]$, leading to

$$\sigma_n = (\sigma_{\max} - \sigma_{\min} + 1)^r - 1 + \sigma_{\min}. \tag{11}$$

The sampling strategy as well as $\sigma_{\min} = 0.01$, $\sigma_{\max}^{\text{CIFAR}} = 50$ and $\sigma_{\max}^{\text{LSUN}} = 378$ (for CIFAR-10 and LSUN Bedroom, respectively) were taken from [1]. To maintain comparability, we did not deviate from this selection process.

B. Inference

After training, we assessed the models on posterior sampling and prior generation, using an exponential moving average of the model's parameters, following [1]. The experiments compare a score function obtained via MUPET with the baseline DSM, which is the pre-trained NCSN++ from [1].

We evaluated MUPET's ability to recover images from degraded observations using a VE version of DiffPIR (see Table II and Fig. 1 for results). To isolate the differences between MUPET and the pre-trained DSM from [1], we plug them into DiffPIR, a well-established diffusion-based reconstruction method with strong performance across various tasks. DiffPIR reports excellent perceptual quality, heavily outperforming PnP in LPIPS [2]. The use of the same DiffPIR procedure and identical network structure allows for a direct evaluation of how MUPET affects diffusion priors for posterior sampling. DiffPIR's strong perceptual performance and principled approach make it a natural choice for evaluating the effects of MUPET.

The sampling process on CIFAR-10 involved 50,000 images on a single Nvidia A40 GPU (30 hours runtime). The inverse problems all considered $\sigma_n = 0.01$ and included:

- **Pixel-wise Completion (pix):** 80% or 90% missing pixels
- **Block-wise completion (block):** 24×24 and 64×64 masked regions
- **Super-resolution (SR):** scaling factors: 2, 3, and 4
- **Deblurring (deblur):** Gaussian blur with $\sigma_{\text{blur}} = 2$

The tests for image restoration were performed on the LSUN Bedroom test-set (300 images). The DiffPIR hyper-parameters (ζ, σ_T) for DSM and MUPET were selected via grid search selecting the values that lead to the best LPIPS, respectively. We chose regularization weight $\lambda = 1$ for its alignment with the original inverse problem formulation. The exact hyper-parameters and code can be found in the repository.

For the evaluation of reconstruction quality, we used **Peak Signal-to-Noise Ratio (PSNR)** and **Structural Similarity Index Measure (SSIM)** [18] for pixel-based accuracy and structural similarity. For perceptual quality, we used **LPIPS** [11]: A deep-learning-based metric that aligns with human perception and is thus often preferred over pixel-based metrics.

We evaluated prior sampling using the predictor-corrector framework for the VE framework from [1]. We generated 50,000 images from CIFAR-10's prior distribution and compared their **Fréchet Inception Distance (FID)** [19] with respect to both the CIFAR-10 training set (50,000 images) and test set (10,000 images). FID estimates distributional similarity between two datasets and was calculated using [20].

V. Results and Discussion

A. Performance Evaluation

We evaluated MUPET by comparing its score functions to those trained with standard DSM [1] on both prior sampling and various inverse problems. DiffPIR was used as a controlled framework to assess the two score-functions in Table II, permitting a direct evaluation of MUPET's effect.

Table I shows a slight improvement in FID when sampling from the prior and highlights that the addition of L^{DEQ} represents a refinement of the score network rather than a conflicting training objective.

TABLE I
Sampling on CIFAR-10; evaluated using FID↓.

Network	Training Set	Test Set
DSM	2.49	4.54
MUPET	**2.45**	**4.50**

TABLE II
Evaluation of DSM and MUPET using DiffPIR: Problems considered are completion (pixel-wise and block-wise), Gaussian deblurring, and super-resolution with $\sigma_n = 0.01$.

Task	Network	PSNR↑	SSIM↑	LPIPS↓
2xSR	DSM	28.53	0.862	0.152
	MUPET	**29.34**	**0.871**	**0.111**
3xSR	DSM	26.21	0.791	0.280
	MUPET	**26.75**	**0.792**	**0.220**
4xSR	DSM	24.54	0.712	0.390
	MUPET	**24.99**	**0.721**	**0.312**
80% pix	DSM	**27.73**	**0.855**	0.088
	MUPET	27.54	0.851	**0.085**
90% pix	DSM	**25.19**	**0.782**	0.167
	MUPET	25.08	0.779	**0.159**
24 block	DSM	40.74	0.985	0.005
	MUPET	**40.97**	**0.987**	0.005
64 block	DSM	**29.68**	0.957	0.031
	MUPET	29.67	**0.959**	0.031
deblur	DSM	26.92	0.790	0.257
	MUPET	**27.05**	**0.802**	**0.201**

Table II demonstrates MUPET's superior perceptual quality in posterior sampling compared to DSM, particularly for super-resolution tasks, where reconstructions can be sharper with fewer artifacts (see Fig. 1). While there are small drops in PSNR by up to 0.23 dB for completion, there is no increase in LPIPS. Specifically, LPIPS improves notably by 20%–27% in super-resolution and deblurring tasks. The absence of LPIPS improvements in block-wise completion aligns with expectations, as this scenario closely resembles prior sampling and thus can be solved well by a network trained on Gaussian-noise perturbations, only.

Fig. 1 highlights that MUPET, while sometimes leading to smoother surfaces, reduces blur and artifacts compared to DSM. The improvement can be ascribed to MUPET having been DEQ-trained in a MAP GD algorithm rather than on images only perturbed by Gaussian noise, exposing it to different influences of the data-fidelity term for different degradations. As a result, MUPET improves posterior sampling while maintaining prior sampling quality, demonstrating the effectiveness of fine-tuning diffusion models within a MAP framework.

B. Limitations and Future Work

While this work is theoretical, we acknowledge that generative models may be misused or produce fabricated content. To mitigate such risks, we focus on datasets not centered around human faces. However, as generative models continue to advance, ethical concerns regarding their deployment such as data bias and malicious use should be considered. Ensuring

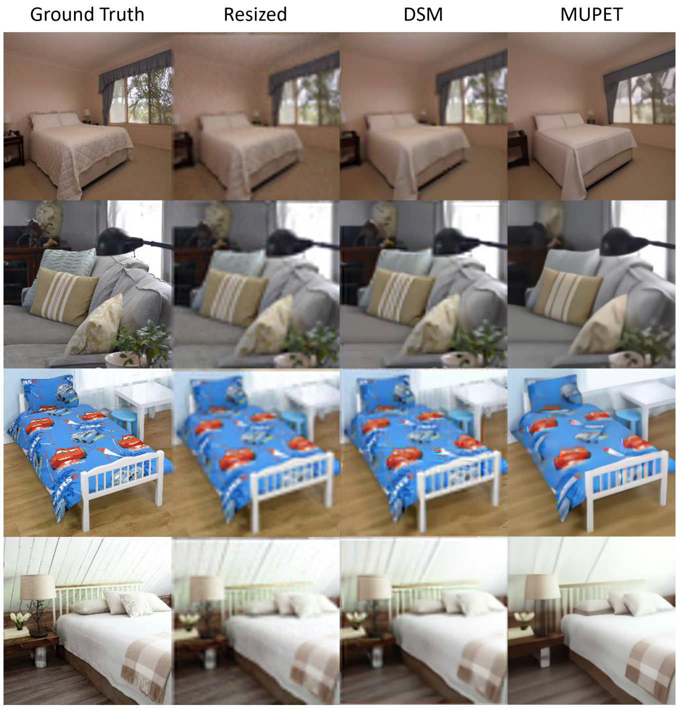

Fig. 1. DiffPIR: 4× super-resolution on LSUN Bedroom

that models like MUPET are used for responsible applications remains an important challenge for the field.

Although DEQ models offer advantages for fine-tuning, they introduce additional computational overhead in training. In our experiments, over 90% of iterations converged before reaching the maximum 500 iterations, suggesting reliable convergence in most cases. Nonetheless, future work could explore adaptive stopping criteria or more advanced root solvers to further reduce training-time.

For this work, we set $\alpha = 1$ in (9) to avoid extensive hyperparameter tuning, but exploring other choices may improve performance. Beyond weighting, the choice of loss function for L^{DEQ} is another avenue for future improvements, as the current loss function is not Bayes-optimal. Since the Bayes-optimal choice is not feasible, investigating alternative loss functions may lead to significant improvements for MAP estimation as a whole and, as a consequence, MUPET.

Future research could extend MUPET to applications like medical imaging (e.g., MRI or CT reconstruction), potentially leading to tangible benefits for society. Additionally, exploring its impact on other inverse problems, such as hyperspectral imaging or remote sensing, may open new avenues for practical deployment.

VI. Conclusion

This work introduced MUPET, a framework that bridges the gap between MAP estimation and generative modeling by fine-tuning diffusion models directly on inverse problems, beyond traditional training restricted to Gaussian-noise perturbations. By directly refining the score network on actual inverse problems, MUPET improves perceptual reconstruction quality (particularly for super-resolution) without compromising generative capability or increasing inference cost. Thus, MUPET is a principled and practical enhancement of existing diffusion-based methods for image restoration.

References

[1] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[2] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *CVPR*, pages 1219–1229, 2023.

[3] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *NeurIPS*, 35:23593–23606, 2022.

[4] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *NeurIPS*, 32, 2019.

[5] Samy Wu Fung, Howard Heaton, Qiuwei Li, Daniel McKenzie, Stanley Osher, and Wotao Yin. Jfb: Jacobian-free backpropagation for implicit networks. In *AAAI Conf. Artif. Intell.*, volume 36, pages 6648–6656, 2022.

[6] Davis Gilton, Gregory Ongie, and Rebecca Willett. Deep equilibrium architectures for inverse problems in imaging. *IEEE TCI*, 7:1123–1133, 2021.

[7] Samuel Willingham, Mårten Sjöström, and Christine Guillemot. Prior for multi-task inverse problems in image reconstruction using deep equilibrium models. In *EUSIPCO*, pages 1220–1224. IEEE, 2023.

[8] Samuel Willingham, Mårten Sjöström, and Christine Guillemot. Training methods for regularizing gradients on multi-task image restoration problems. In *Volume 3: VISAPP*, pages 145–153. INSTICC, SciTePress, 2024.

[9] Rita Fermanian, Mikael Le Pendu, and Christine Guillemot. Pnp-reg: Learned regularizing gradient for plug-and-play gradient descent. *SIIMS*, 16(2):585–613, 2023.

[10] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *GlobalSIP*, pages 945–948. IEEE, 2013.

[11] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[12] Jiezhang Cao, Yue Shi, Kai Zhang, Yulun Zhang, Radu Timofte, and Luc Van Gool. Deep equilibrium diffusion restoration with parallel sampling. In *CVPR*, pages 2824–2834, 2024.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

[14] Bradley Efron. Tweedie's formula and selection bias. *JASA*, 106(496):1602–1614, 2011.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[17] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.

[18] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017.

[20] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.3.0.