High Perceptual Quality Image Denoising with a Posterior Sampling CGAN

Guy Ohayon, Technion
Theo Adrai, Technion
Gregory Vaksman, Technion
Michael Elad, Google Research
Peyman Milanfar, Google Research

Abstract

The vast work in Deep Learning (DL) has led to a leap in image denoising research. Most DL solutions for this task have chosen to put their efforts on the denoiser's architecture while maximizing distortion performance. However, distortion driven solutions lead to blurry results with sub-optimal perceptual quality, especially in immoderate noise levels. In this paper we propose a different perspective, aiming to produce sharp and visually pleasing denoised images that are still faithful to their clean sources. Formally, our goal is to achieve high perceptual quality with acceptable distortion. This is attained by a stochastic denoiser that samples from the posterior distribution, trained as a generator in the framework of conditional generative adversarial networks (CGAN). Contrary to distortion-based regularization terms that conflict with perceptual quality, we introduce to the CGAN objective a theoretically founded penalty term that does not force a distortion requirement on individual samples, but rather on their mean. We showcase our proposed method with a novel denoiser architecture that achieves the reformed denoising goal and produces vivid and diverse outcomes in immoderate noise levels.

1. Introduction

Image denoising is one of the most fundamental problems in image processing, and as such it has been explored quite extensively. As deep learning emerged in the past decade, many neural-network-based attempts were made to solve this task. These led to state-of-the-art (SoTA) performance in commonly used full reference *distortion* measures, such as Mean-Squared-Error (MSE), that quantify the discrepancy between the denoised image and its clean source [26, 27, 29, 33, 34, 35]. While optimizing a distortion measure leads to denoised images that are faithful to their clean sources, the *perceptual quality*, which is the degree to which a denoised image looks natural, is also an important measure to consider. Recent works have tried to achieve higher perceptual quality compared to distortion based solutions by concurrently optimizing both measures [7, 8]. However, these attempts achieve sub-optimal perceptual quality [5], and to the best of our knowledge, there are hardly any deep learning solutions that address the image denoising problem while targeting optimal perceptual performance.

In this work we seek to obtain a denoiser that achieves very high perceptual quality, while accompanied by a guarantee on its distortion performance. As shown in [5], sampling from the posterior distribution achieves such a goal, compromising 3dB on the optimal Peak Signal To Noise Ratio (PSNR) performance, making such a stochastic denoiser an excellent candidate for our needs. The idea of using posterior sampling for solving image restoration tasks has already been suggested in various contexts [1, 25, 28]. Although high dimensional posterior sampling is still considered as a challenging task, recent deep learning methods seem to provide practical tools for handling it.

The success of generative adversarial networks (GAN) has led authors to incorporate sampling (not necessarily from the posterior distribution) to solve various image restoration tasks on certain classes of images [3, 4, 21, 30], and to excellent sampling capabilities from class-specific priors [15, 16]. Most of these were possible due to the improvements in the generative adversarial learning scheme [2, 11, 18] that allowed stable training, contrary to the instabilities of the originally proposed GAN optimization objective [10]. The authors of [1] have shown that the CGAN objective [22] formalized under the Wasserstein-1 metric [2, 11] theoretically drives a conditional generator to sample from the posterior distribution. Therefore, such an optimization framework provides a practical way to approximate the desired sampling. For instance, the Latent Adversarial Generator (LAG) [4] has shown SoTA single image super resolution (SISR) results from an extremely low resolution input, attained by a tweaked version of CGAN.

Rather than seeking a balance between perceptual quality and distortion performance, we aim to sample from the posterior distribution while willing to compromise up to 3dB in PSNR performance. In order to regularize the proposed sampling, we leverage the property that any stochastic denoiser that samples from such a distribution must also agree in expectation with it. We introduce a term to the CGAN objective that penalizes solutions which do

not satisfy such a necessary property. Unlike other related methods, our regularization term does not force a distortion requirement on individual denoised samples, but rather on their mean. Our proposed denoiser's architecture is a novel encoder-decoder, inspired by StyleGAN2 [15] and UNet [23], with a high receptive field and a noise injection scheme generalizing that of StyleGAN [16]. We showcase the capabilities of our proposed method in high noise conditions, basing our experiments on several data sets.

2. Proposed Method: Derivations

Assume an unknown distribution of images $\mathbb{P}_{\mathbf{x}}$ and a known stochastic degradation operator $\mathbf{deg}(\cdot)$ (such as additive Gaussian noise). Our goal is to sample from the posterior distribution $\mathbb{P}_{\mathbf{x}|\mathbf{y}}$ with the help of an independent random vector $\mathbf{z}$ of known distribution. We assume that given $\mathbf{y} = \mathbf{deg}(\mathbf{x})$, a degraded observation of $\mathbf{x}$, there exists a parametric mapping $\mathbf{g}_\theta = G_\theta(\mathbf{z}, \mathbf{y})$ such that $\mathbf{z} \sim \mathbb{P}_{\mathbf{z}}$, $\mathbf{g}_\theta|\mathbf{y} \sim \mathbb{P}_{\mathbf{x}|\mathbf{y}}$, and $\mathbb{P}_{\mathbf{z}}$ is a known latent distribution where $\mathbf{z}$ and $\mathbf{y}$ are mutually independent.

For a given $\mathbf{y} = y$ (denoting y as a realization of the random variable $\mathbf{y}$), the Wasserstein-1 distance [2] between $\mathbb{P}_{\mathbf{x}|\mathbf{y}=y}$ and $\mathbb{P}_{\mathbf{g}_\theta|\mathbf{y}=y}$ can be shown to satisfy the equality

$$\begin{aligned} &W_1(\mathbb{P}_{\mathbf{x}|\mathbf{y}=y}, \mathbb{P}_{\mathbf{g}_\theta|\mathbf{y}=y}) \\ &= \sup_{f \in L_1} \mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[f(\mathbf{x}, y)\right] - \mathbb{E}_{\mathbf{g}_\theta|\mathbf{y}} \left[f(\mathbf{g}_\theta, y)\right], \end{aligned} \tag{1}$$

where L_1 is the set of all functions $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that are 1-Lipschitz on $\mathcal{X}$ for any $y \in \mathcal{Y}$. Observe that (1) is defined for a given realization of $\mathbf{y}$, whereas we seek an optimization objective that considers all possible ones. This can be accomplished by taking an expectation on both sides with respect to $\mathbf{y}$. The work in [1] shows that such an expectation taken on the right hand side in (1) commutes with the supremum, leading to

$$\begin{aligned} &\mathbb{E}_{\mathbf{y}} \left[W_1(\mathbb{P}_{\mathbf{x}|\mathbf{y}}, \mathbb{P}_{\mathbf{g}_\theta|\mathbf{y}})\right] \\ &= \sup_{f \in L_1} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[f(\mathbf{x}, \mathbf{y})\right] - \mathbb{E}_{\mathbf{g}_\theta,\mathbf{y}} \left[f(\mathbf{g}_\theta, \mathbf{y})\right]. \end{aligned} \tag{2}$$

Observe that contrary to (1), the expectations in (2) act on the joint distributions $\mathbb{P}_{\mathbf{x},\mathbf{y}}$ and $\mathbb{P}_{\mathbf{g}_\theta,\mathbf{y}}$. Therefore, assuming that the function f in the supremum of (2) can be found for each y, one could evaluate this distance as follows:

- Draw samples of $\mathbf{x} \sim \mathbb{P}_{\mathbf{x}}$ (e.g., get an image data set).
- Perform $\mathbf{y} = \mathbf{deg}(\mathbf{x})$ on each sample of $\mathbf{x}$, to obtain samples of $\mathbf{y}$ (e.g., contaminate with noise).
- Draw independently samples of $\mathbf{z} \sim \mathbb{P}_{\mathbf{z}}$.
- Compute $G_\theta(\mathbf{z}, \mathbf{y})$ on each sample of $\mathbf{y}$ and $\mathbf{z}$, to obtain samples of $\mathbf{g}_\theta$ (e.g., denoise each noisy image with a stochastic denoiser).
- We now have samples drawn from both $\mathbb{P}_{\mathbf{x},\mathbf{y}}$ and $\mathbb{P}_{\mathbf{g}_\theta,\mathbf{y}}$. Evaluate (2) using the law of large numbers.

Considering $G_\theta(\mathbf{z}, \mathbf{y})$ as a generator and assuming that f is somehow realized for each θ, we could optimize for θ:

$$\begin{aligned} &\min_\theta \mathcal{L}(\mathbb{P}_{\mathbf{x}|\mathbf{y}}, \mathbb{P}_{\mathbf{g}_\theta|\mathbf{y}}) \\ &= \min_\theta \sup_{f \in L_1} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[f(\mathbf{x}, \mathbf{y})\right] - \mathbb{E}_{\mathbf{g}_\theta,\mathbf{y}} \left[f(\mathbf{g}_\theta, \mathbf{y})\right]. \end{aligned} \tag{3}$$

If f is a parametrized critic, this is a game between two adversaries, having a classic GAN structure. While this optimization task may seem appealing and practical, it is in fact ill-posed since we are confined to a finite sized and unbalanced data-set, in which for each $\mathbf{x}$ we have many $\mathbf{y}'s$ but not vice versa. A generator optimized under (3) with such data would try to learn to sample from the posterior distribution $\mathbb{P}_{\mathbf{x}|\mathbf{y}}$ with only one sample from $\mathbf{x}|\mathbf{y}$ for each $\mathbf{y}$. This would most likely lead to mode collapse [10, 13, 20, 31], where $\mathbf{g}_\theta|\mathbf{y}$ becomes a degenerate random variable and the generator ignores $\mathbf{z}$, since for each conditional input $\mathbf{y}$ it is sufficient for the generator to produce only one image that is acceptable by the critic. Hence, the densities $\mathbb{P}_{\mathbf{g}_\theta|\mathbf{y}}$ and $\mathbb{P}_{\mathbf{x}|\mathbf{y}}$ might be equal only on this finite number of points, while allowing a deviation in the remaining domain. To alleviate this weakness, we add a constraint to (3) as follows:

$$\begin{aligned} &\min_\theta \mathcal{L}(\mathbb{P}_{\mathbf{x}|\mathbf{y}}, \mathbb{P}_{\mathbf{g}_\theta|\mathbf{y}}) \\ &s.t.\ \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\|\mathbf{x} - \mathbb{E}\left[\mathbf{g}_\theta|\mathbf{y}\right]\|_2^2\right] = \mathbb{E}_{\mathbf{y}} \left[Var(\mathbf{x}|\mathbf{y})\right]. \end{aligned} \tag{4}$$

Observe that if there exists θ^* such that $p_{\mathbf{g}_{\theta^*}|\mathbf{y}} = p_{\mathbf{x}|\mathbf{y}}$, then θ^* is a global optimum of task (3), implying that the distance between the two conditional distributions is zero. In addition, θ^* remains the global optimum of task (4) since $p_{\mathbf{g}_{\theta^*}|\mathbf{y}} = p_{\mathbf{x}|\mathbf{y}}$ implies that $\mathbb{E}[\mathbf{g}_{\theta^*}|\mathbf{y}] = \mathbb{E}[\mathbf{x}|\mathbf{y}]$, and thus $\mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\|\mathbf{x} - \mathbb{E}\left[\mathbf{g}_{\theta^*}|\mathbf{y}\right]\|_2^2\right] = \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\|\mathbf{x} - \mathbb{E}\left[\mathbf{x}|\mathbf{y}\right]\|_2^2\right] = \mathbb{E}_{\mathbf{y}} \left[Var(\mathbf{x}|\mathbf{y})\right]$, which is the Minimum Mean Squared Error (MMSE). Thus, the added constraint is also satisfied by θ^*, which means that it is a necessary condition on any solution that yields $p_{\mathbf{g}_{\theta^*}|\mathbf{y}} = p_{\mathbf{x}|\mathbf{y}}$. In other words, instead of having distortion requirements on specific samples (as in LAG [4] for instance), we require an agreement with the expectation of the posterior, i.e., the constraint enforces many samples of $\mathbf{g}_\theta|\mathbf{y}$ to agree with $\mathbf{x}|\mathbf{y}$ (in expectation). As we will see in Section 4, this revision leads to a stochastic variation and therefore circumvents mode collapse.

Since $\mathbb{E}_{\mathbf{y}} \left[Var(\mathbf{x}|\mathbf{y})\right]$ is the global minimum of $\mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\|\mathbf{x} - \mathbb{E}\left[\mathbf{g}_\theta|\mathbf{y}\right]\|_2^2\right]$, we can reformulate the optimization of task (4) by adding a penalty term to task (3):

$$\min_\theta \mathcal{L}(\mathbb{P}_{\mathbf{x}|\mathbf{y}}, \mathbb{P}_{\mathbf{g}_\theta|\mathbf{y}}) + \lambda \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\|\mathbf{x} - \mathbb{E}\left[\mathbf{g}_\theta|\mathbf{y}\right]\|_2^2\right]. \tag{5}$$

The posterior distribution is still a globally optimal solution to this problem, since both expressions admit their minimum for the same θ^*. This way, the proposed scheme eliminates many possible solutions that might minimize the first term but are far from the true posterior.

Algorithm 1: Training of the Posterior Sampling CGAN (PSCGAN).

Require: The gradient penalty coefficient λ_{GP}, the expected distance coefficient λ_{MM}, the number of critic iterations per generator iteration n_{critic}, the batch size B, the number of sampled realizations from the generator M, the penalty batch size PB, Adam hyperparameters α, β_1, β_2, initial critic and generator parameters ω_0 and θ_0.

Default Settings:

$\lambda_{GP} = 10, \lambda_{MM} = 10^{-3}, n_{critic} = 1, B = 32, M = 8, PB = 8, \alpha = 2.5 \cdot 10^{-4}, \beta_1 = 0, \beta_2 = 0.99.$

while θ *has not converged* **do**

  for $t = 1, \ldots, n_{critic}$ **do**

    for $i = 1, \ldots, B$ **do**

      Sample $x \sim \mathbb{P}_{\mathbf{x}}$, $z \sim \mathbb{P}_{\mathbf{z}}$, $\epsilon \sim U[0,1]$

      $y \leftarrow \mathbf{deg}(x)$ `// A stochastic degradation operator.`

      $\tilde{x} \leftarrow G_\theta(z, y)$

      $\hat{x} \leftarrow \epsilon x + (1-\epsilon)\tilde{x}$

      $L_C^{(i)} \leftarrow \lambda_{MM}(C_\omega(\tilde{x}, y) - C_\omega(x, y)) + \lambda_{GP}(\|\nabla_{\hat{x}} C_\omega(\hat{x}, y)\|_2 - 1)^2$

    $\omega \leftarrow \text{Adam}(\nabla_\omega \frac{1}{B} \sum_{i=1}^{B} L_C^{(i)}, \omega, \alpha, \beta_1, \beta_2)$

  for $i = 1, \ldots, B$ **do**

    Sample $x \sim \mathbb{P}_{\mathbf{x}}$, $z^{(0)} \sim \mathbb{P}_{\mathbf{z}}$

    $y \leftarrow \mathbf{deg}(x)$

    $L_{G_{MM}}^{(i)} \leftarrow -\lambda_{MM} C_\omega(G_\theta(z^{(0)}, y), y)$

    if $i \leq PB$ **then**

      Sample a batch $\{z^{(j)}\}_{j=1}^{M}$, each from $\mathbb{P}_{\mathbf{z}}$

      $L_{G_A}^{(i)} \leftarrow \|x - \frac{1}{M} \sum_{j=1}^{M} G_\theta(z^{(j)}, y)\|_2^2$

  $\theta \leftarrow \text{Adam}(\nabla_\theta \left[\frac{1}{B} \sum_{i=1}^{B} L_{G_{MM}}^{(i)} + \frac{1}{PB} \sum_{i=1}^{PB} L_{G_A}^{(i)} \right], \theta, \alpha, \beta_1, \beta_2)$

At first glance, LAG [4] could also seem like a method that directly aims for the perceptual quality goal. LAG's objective is almost identical to that of CGAN, with an additional generator regularization term:

$$\min_\theta \mathcal{L}(\mathbb{P}_{\mathbf{x}|\mathbf{y}}, \mathbb{P}_{\mathbf{g}_\theta|\mathbf{y}}) + \lambda \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\|P(\mathbf{x}, \mathbf{y}) - P(G_\theta(0, \mathbf{y}), \mathbf{y})\|_2^2 \right]. \tag{6}$$

In LAG, the function $P(\cdot, \cdot)$ in the above expression represents a part of the critic that extracts features from the image pair fed to it. This function could be considered as a variant or part of the function f we have used above. Referring to the second term, its rationale is the belief that the intermediate representations of $(\mathbf{x}, \mathbf{y})$ and $(G_\theta(0, \mathbf{y}), \mathbf{y})$ are necessarily close-by. Observe that this penalty is substantially different from the expectation requirement we have posed in equation (5), since we do not assume that a given sample (i.e., the one attained at $\mathbf{z} = 0$) and $\mathbf{x}$ are matched in distortion. When $G_\theta(\cdot, \cdot)$ and $P(\cdot, \cdot)$ are continuous mappings, this assumption poses a distortion requirement not only on $G_\theta(0, \mathbf{y})$, but also on its neighbourhood. Thus, the penalty term in (6) conflicts with the perceptual quality goal [5], whereas the penalty term we propose in (5) does not.

3. Proposed Method: Details

3.1. Training Method

Our training method is directly derived from optimization task (5). To enforce the 1-Liphshitz constraint on the critic (denoted as f in equation (3)), we use the gradient penalty version of WGAN [11]. That is, we train a generator G_θ and a critic C_ω (replacing f, to align with common WGAN notations) via the min-max optimization game

$$\min_\theta \max_\omega \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\|\mathbf{x} - \mathbb{E}_{\mathbf{z}} \left[G_\theta(\mathbf{z}, \mathbf{y}) | \mathbf{y} \right]\|_2^2 \right] + \lambda_{MM} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[C_\omega(\mathbf{x}, \mathbf{y}) \right] - \mathbb{E}_{\mathbf{z},\mathbf{y}} \left[C_\omega(G_\theta(\mathbf{z}, \mathbf{y}), \mathbf{y}) \right] + \lambda_{GP} \mathbb{E}_{\hat{\mathbf{x}},\mathbf{y}} \left[(\|\nabla_{\hat{\mathbf{x}}} C_\omega(\hat{\mathbf{x}}, \mathbf{y})\|_2 - 1)^2 \right], \tag{7}$$

where for a given $\mathbf{y} = y$, the last expectation is taken with respect to $\mathbb{P}_{\hat{\mathbf{x}}}$, the distribution of uniform samples along straight lines between pairs of points sampled from $\mathbb{P}_{\mathbf{x}|\mathbf{y}=y}$ and $\mathbb{P}_{\mathbf{g}_\theta|\mathbf{y}=y}$. Our proposed training method is described in Algorithm 1 and our proposed generator architecture is in Appendix A.

3.2. A Denoiser with Two Distinct Capabilities

Recall that in our training method we drive our generator towards the production of samples from the posterior distribution while constraining the average denoised image to

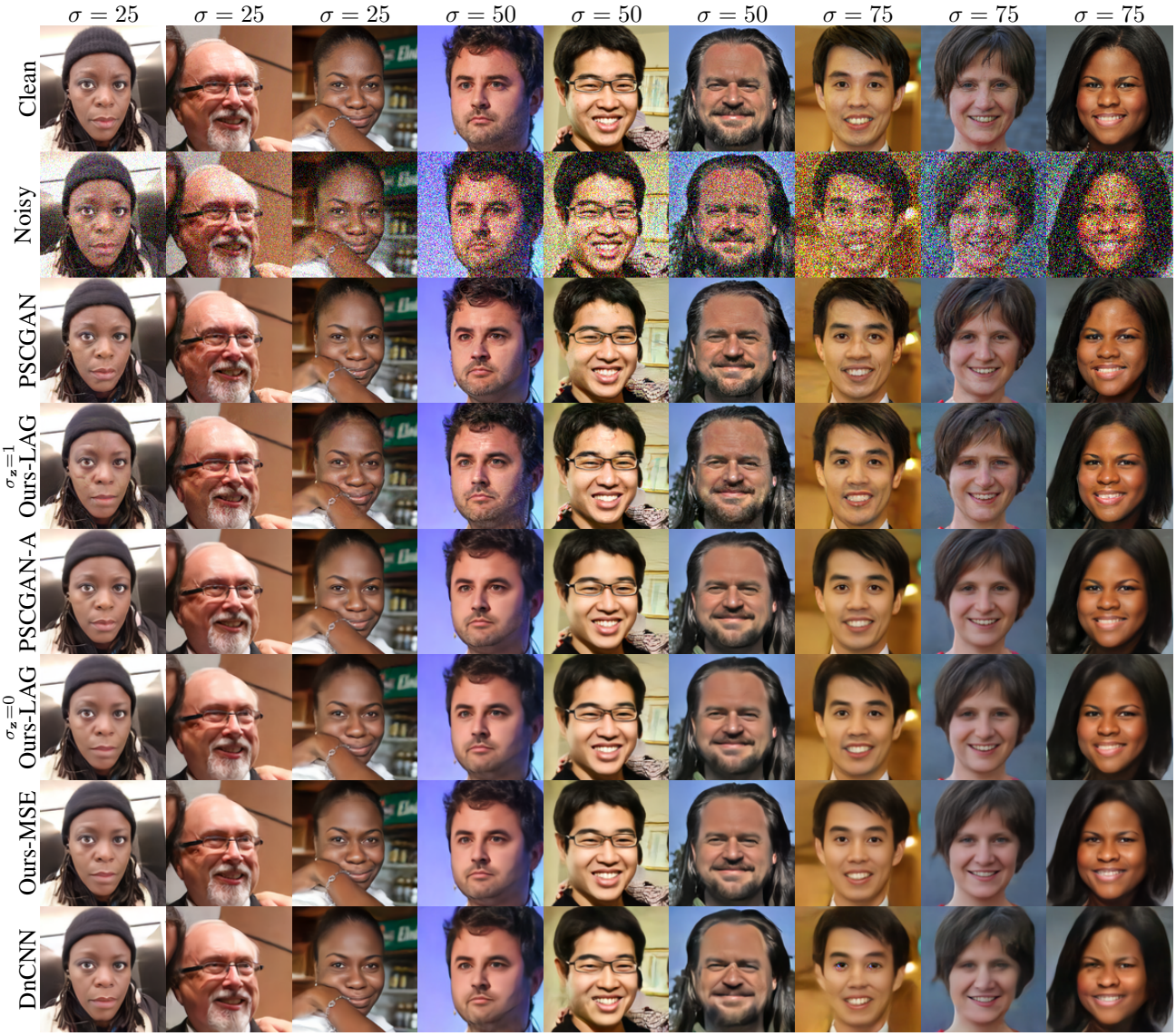

Figure 1: Denoising results on the FFHQ test set produced by several methods. PSCGAN is a sampled denoised image produced by our proposed method, attained by injecting noise with standard deviation of $\sigma_{\mathbf{z}} = 1$ (at both training and inference time). Ours-LAG($\sigma_{\mathbf{z}} = 0$) and Ours-LAG($\sigma_{\mathbf{z}} = 1$) are the same models, while the former is with $\sigma_{\mathbf{z}} = 0$ and the latter is with $\sigma_{\mathbf{z}} = 1$ at inference time. In this case, PSCGAN-A averages 64 instances of PSCGAN. Each model was trained on the FFHQ training set to denoise a specific noise level ($25, 50$ or 75).

minimize the MSE. Thus, given such a trained model with enough capacity, the average denoised image of a given noisy input should approximately achieve the MMSE. This allows the optimized model to go beyond sampling from the posterior, producing an MMSE approximation result by averaging many generated samples. Even if the trained model does not accurately capture the true posterior, an average denoised image should produce low MSE, while a sampled denoised image should produce high perceptual quality. Our training method therefore allows one to obtain two denoisers at the same time: a denoiser that approximately samples from the posterior distribution and achieves high perceptual quality, and a denoiser that approximates the MMSE estimator. In Section 4 we refer to the former as *PSCGAN* and to the latter as *PSCGAN-A*.

4. Experimental Evaluation

We turn to present an evaluation of several methods:

- PSCGAN, our proposed method, which aims to sample from the posterior distribution.

Figure 2: Stochastic variation of denoised images attained by 3 different generators, each trained with PSCGAN to denoise images contaminated with noise levels of $\sigma = 25, 50, 75$. Two clean images are presented to the left, and their corresponding noisy versions to their right. Alongside each noisy input we show 4 examples of possible denoising outcomes, as well as the 4^{th} root of the per-pixel standard deviation image calculated on 32 samples. For convenience, a gray-scale color map is added to the right (white and black correspond to low and high standard deviations, respectively). All denoised image samples were obtained by injecting noise with $\sigma_{\mathbf{z}} = 1$ at inference time.

- PSCGAN-A, which averages instances of PSCGAN.
- Ours-MSE, which is our proposed generator trained to solely optimize the MSE loss (without noise injections). Comparing its performance with PSCGAN allows us to better evaluate our proposed training method to an MSE based optimization procedure since we use the same architecture in both cases.
- DnCNN [34], a commonly accepted baseline.

Additionally, we evaluate *Ours-LAG*, a variant of LAG [4], only to illustrate several tendencies regarding the perception-distortion tradeoff in subsection 4.2. In Appendix B we describe our implementation choices.

In all experiments of PSCGAN the noise injected to the generator is of Gaussian distribution with zero mean. We vary at inference time the standard deviation of all noisy maps injected to the generator, which we denote as $\sigma_{\mathbf{z}}$ ($\sigma_{\mathbf{z}} = 0$ means $\mathbf{z} = 0$). To clarify, PSCGAN is always trained with $\sigma_{\mathbf{z}} = 1$. We also vary the number of instances produced by PSCGAN that are being averaged to compute PSCGAN-A, which we denote by N. We base our evaluations on the FFHQ [16] thumbnails, LSUN Bedroom and LSUN Church outdoor [32] data sets, and assess the performance on images contaminated with different levels of additive white Gaussian noise with $\sigma \in \{25, 50, 75\}$. To clarify, we train a separate denoiser for each configuration of data set and noise level. Supplementary training details are in Appendix C.

4.1. Perceptual Quality and Distortion Evaluation

In Figure 1 we demonstrate the perceptual quality of all evaluated methods on the FFHQ test set, including Ours-LAG. The visual results produced on both LSUN test sets are in Appendix D. PSCGAN and Ours-LAG (at $\sigma_{\mathbf{z}} = 1$) produce sharp and real looking results and outperform the

Data set	σ	PSCGAN		PSCGAN-A		Ours-MSE		DnCNN	
		PSNR	FID	PSNR	FID	PSNR	FID	PSNR	FID
FFHQ	25	29.19	**12.66** $\pm$ 0.07	31.46	27.48	**31.65**	32.38	30.15	38.63
	50	25.83	**15.18** $\pm$ 0.15	28.28	31.81	**28.44**	41.56	28.30	42.97
	75	24.09	**15.78** $\pm$ 0.13	26.57	34.64	**26.81**	46.31	26.46	47.69
LSUN Church	25	29.03	**7.66** $\pm$ 0.04	30.78	9.33	**31.20**	9.69	31.16	10.25
	50	25.50	**9.02** $\pm$ 0.06	27.54	10.86	**27.77**	12.93	27.69	15.66
	75	23.75	**9.12** $\pm$ 0.09	25.84	12.39	**26.00**	14.94	25.78	22.12
LSUN Bedroom	25	30.62	**8.83** $\pm$ 0.05	32.29	9.41	**32.57**	11.86	28.96	12.52
	50	27.30	**9.27** $\pm$ 0.06	29.08	11.13	**29.30**	12.71	29.10	21.38
	75	25.23	**11.56** $\pm$ 0.08	27.26	13.74	**27.43**	15.57	27.14	31.69

Table 1: The PSNR (dB) and FID results obtained by several evaluated methods, each trained to denoise images contaminated with a specific noise level (higher PSNR and lower FID correspond to better performance). Notice that the reported PSNR is not the average one, but rather the PSNR calculated on the average MSE of the entire test set. PSCGAN is our sampler from the learned distribution, where we use $\sigma_{\mathbf{z}} = 1$ for the FFHQ test set and $\sigma_{\mathbf{z}} = 0.75$ for both LSUN test sets during inference. In this case, PSCGAN-A averages $N = 64$ instances of PSCGAN (obtained with $\sigma_{\mathbf{z}} = 1$ on all data sets). Ours-MSE is our proposed generator trained to solely optimize the MSE loss (without noise injections). The FID reports of PSCGAN contain both the mean and the standard deviation (denoted with $\pm$).

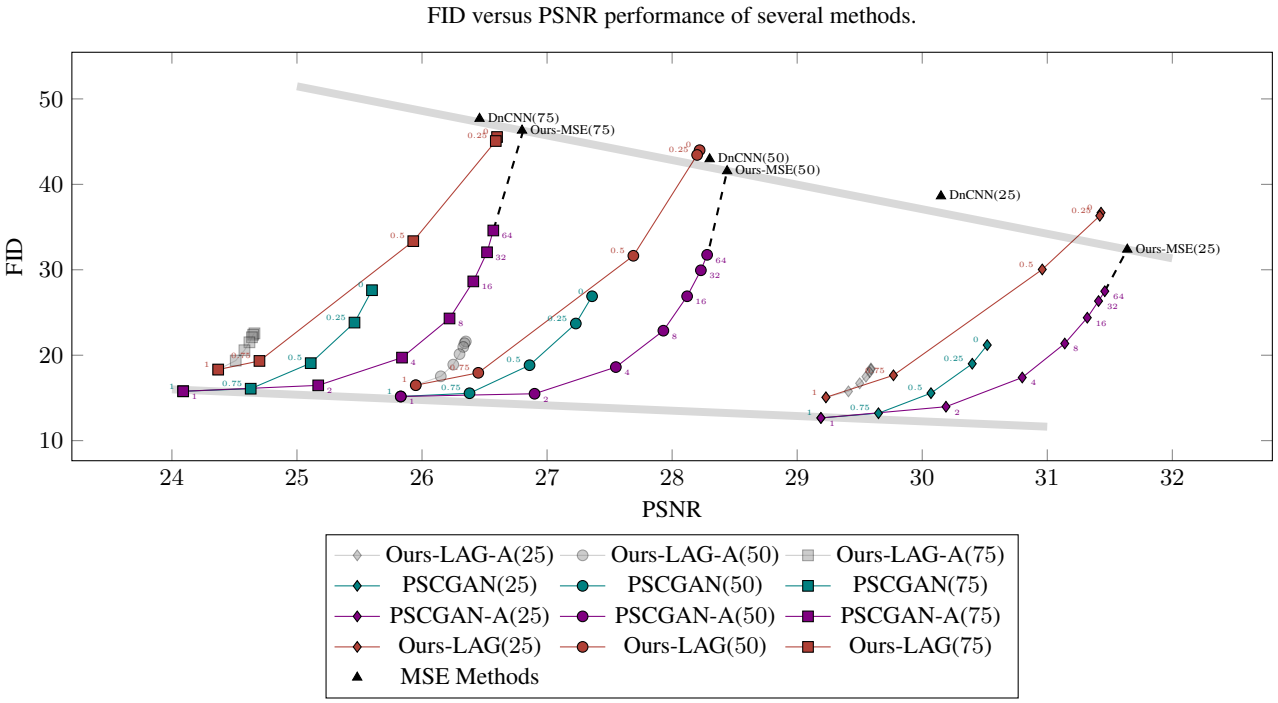

Figure 3: FID versus PSNR results for PSCGAN, PSCGAN-A that averages N PSCGAN instances, Ours-LAG, Ours-LAG-A that averages N Ours-LAG instances, Ours-MSE and DnCNN. The noise contamination level ($\sigma = 25, 50, 75$) is given with parentheses next to the name of each method. PSCGAN and PSCGAN-A are evaluated on different choices of $\sigma_{\mathbf{z}}$ and N during inference, while $\sigma_{\mathbf{z}}$ is fixed to 1 when varying N. Ours-LAG and Ours-LAG-A are evaluated in the same fashion. For PSCGAN and Ours-LAG the values of $\sigma_{\mathbf{z}}$ are given next to each marked point. Similarly, the values of N are given for PSCGAN-A. The performance results of the MSE based methods are also plotted.

MSE methods in terms of perceptual quality, as the latter produce unnaturally smooth images.

Recall that PSCGAN is a stochastic denoiser able to produce many denoised outputs, and such variability is demonstrated in Figure 2. Even though the overall appearance of the varying samples is similar, we observe a rich stochastic variation on fine details such as wrinkles, hair, eyes and more. In the same figure we also show the 4^{th} root of the

per-pixel standard deviation calculated on 32 denoised samples of the same noisy input. These results suggest that the penalty term in optimization objective (5) indeed circumvents the aforementioned mode collapse issue. In addition, it appears that our model does not suffer from inherent bias when handling skin tones, possibly due to the richness of the FFHQ data set [16].

To quantitatively evaluate our proposed method we use the Fréchet Inception Distance (FID) [12], which is known to correlate well with human opinion scores. Reliably computing FID requires a large amount of "real" samples (typically, at least 50,000), but evidently does not require many "fake" samples to remain consistent [19]. To confirm this, we measure the FID (by using [24]) between each of the training sets and 100 randomly chosen subsets of 500 images taken from its corresponding test set, and see negligible variability in the scores. Thus, our procedure to measure the FID of each denoising algorithm is to consider its outputs on each test set to its entirety as the fake samples, and use all the clean images of the corresponding training set as the real ones. Since PSCGAN produces stochastic denoised images, we evaluate its average FID by repeating this procedure 32 times, where in each the FID is calculated by producing one realization of denoised image for each noisy input.

We report in Table 1 the PSNR and FID scores obtained by all evaluated methods (except for Ours-LAG, which we evaluate only on the FFHQ test set in subsection 4.2). Several tendencies should be highlighted:

- As expected, in terms of PSNR, the best performing methods are the MMSE ones, with a gap of less than 3dB between these and PSCGAN, just as anticipated in [5].
- PSCGAN-A that averages 64 PSCGAN instances provides a very good approximation for the MMSE denoiser.
- In FID terms, PSCGAN outperforms all other methods in all configurations, achieving superior perceptual quality.

It is important to note that the reported FID results are not necessarily the optimal ones, and thus we do not claim optimality. We use this measure to provide some quantitative evaluation of the perceptual quality and to illustrate the perception-distortion tradeoff [5] in the next section, but do not perform hyperparameter tuning to achieve the best FID.

4.2. Traversing The Perception-Distortion Tradeoff

Both PSCGAN and Ours-LAG allow traversing the perception-distortion tradeoff in two ways: by varying $\sigma_{\mathbf{z}}$ or by varying N. We vary $\sigma_{\mathbf{z}}$ with values taken from $\{0, 0.25, 0.5, 0.75, 1\}$, and vary N with values taken from $\{1, 2, 4, 8, 16, 32, 64\}$ (while fixing $\sigma_{\mathbf{z}} = 1$). We demonstrate the above traversals on the FFHQ test set in Figure 3, along with the FID and PSNR scores obtained by all evaluated MSE based methods. For PSCGAN and Ours-LAG we report the average FID scores and omit their standard deviations since they are negligible. We observe that:

- As theoretically expected, PSCGAN-A approaches Ours-MSE as N increases, which suggests that our training procedure was successful in satisfying the penalty term, since with the same architecture we see a comparable PSNR performance when solely optimizing the MSE.
- For PSCGAN, varying N (while fixing $\sigma_{\mathbf{z}} = 1$) is more effective than varying $\sigma_{\mathbf{z}}$ (while fixing $N = 1$), since each choice of $\sigma_{\mathbf{z}}$ is *dominated* [5] by some choice of N. In contrast, varying $\sigma_{\mathbf{z}}$ is more effective for Ours-LAG. We leave the explanation of these for future research.
- The FID performance of PSCGAN is only slightly affected by the noise level, suggesting that PSCGAN leads to high perceptual quality regardless of the noise contamination severity. This aligns with the posterior sampler's property to always produce images with perfect perceptual quality [5]. In contrast, the PSNR performance of PSCGAN decreases as the noise level increases, which makes the perception-distortion tradeoff more significant in higher noise levels (emphasized by the two linear lines that diverge as the noise level increases). This evidence suggests that the gap in the perceptual quality of images produced by the posterior sampler and the MMSE estimator does not remain constant with the noise level, unlike the constant 3dB gap in PSNR [5].
- Averaging instances of Ours-LAG leads to a mild effect on the FID and PSNR scores. We leave the explanation of this phenomenon for future research.
- The results of Ours-LAG at $\sigma_{\mathbf{z}} = 0$ and 0.25 are almost identical, emphasizing that the low distortion requirement on $G_\theta(\mathbf{z} = 0, \mathbf{y})$ constrains its neighborhood with a similar requirement, when $G(\cdot, \cdot)$ is a continuous mapping. Indeed, when $\sigma_{\mathbf{z}} = 1$, even though both Ours-LAG and PSCGAN use the same generator and critic architectures, the former slightly outperforms the latter in PSNR, while the opposite is true in FID. As claimed in Section 2, this shows a conflict with the perceptual quality goal [5] at $\sigma_{\mathbf{z}} = 1$. Consequently, we hypothesize that this leads PSCGAN-A to dominate Ours-LAG, since the latter finds a middle ground between the perceptual quality at $\sigma_{\mathbf{z}} = 1$ and the distortion performance at $\sigma_{\mathbf{z}} = 0$.
- The traversal curves of Ours-LAG are more "stretched" than those of PSCGAN (when varying $\sigma_{\mathbf{z}}$). Both "pull" the $\sigma_{\mathbf{z}} = 1$ points towards high perceptual quality (and therefore towards high distortion [5]), while only Ours-LAG "pulls" the $\sigma_{\mathbf{z}} = 0$ points towards low distortion (and therefore towards low perceptual quality [5]).

4.3. Noise Reduction Evaluation

Image denoising is the process of recovering a clean signal $\mathbf{x}$ from a noisy observation $\mathbf{y}$, where in our case, $\mathbf{y} = \mathbf{x} + \mathbf{n}$ and $\mathbf{n}$ is white Gaussian noise. This is an ill-posed inverse problem and usually $\mathbf{x}$ can not be fully re-

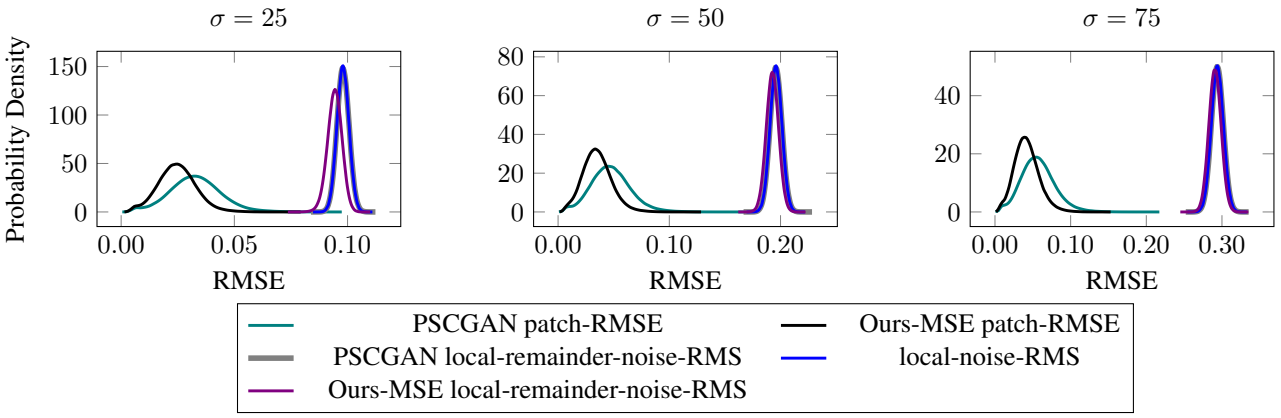

Figure 4: The approximated p.d.f of the patch-RMSE and of the local-remainder-noise-RMS obtained by PSCGAN and by Ours-MSE, and the approximated p.d.f of the local-noise-RMS obtained by Gaussian noise.

trieved. Thus, a *denoising algorithm* should find an approximation of $\mathbf{x}$, denoted as $\hat{\mathbf{x}}$, such that $\mathbf{x}$ and $\hat{\mathbf{x}}$ are close-by, and the *remainder noise* $\hat{\mathbf{n}} = \mathbf{y} - \hat{\mathbf{x}}$ is normally distributed. We assess whether PSCGAN, the algorithm which aims to sample from the posterior, satisfies such criteria.

The theoretical 3dB PSNR gap between the denoising results of a posterior sampler and of the MMSE estimator [5] guarantees that a well trained model that aims to sample from the posterior distribution would produce denoised images that are close to their clean sources, when closeness is measured with MSE (as in many denoising algorithms). Although the PSNR results obtained by PSCGAN (Table 1) are indeed high, it clearly generates local content that is absent from the source image. Thus, we question whether the local patches of the reconstructed samples are still faithful to their clean counterparts. We measure the *patch-RMSE*, the square root of the MSE between all overlapping clean and denoised patches of size 15×15, for all images in the FFHQ test set and for both PSCGAN and Ours-MSE. Finally, we create a histogram of the patch-RMSE values for each algorithm, and present the results in Figure 4. These give an approximation of the probability density function (p.d.f) of the patch-RMSE values obtained by each method. In the same figure we also show the approximated p.d.f of the *local-noise-RMS*, the root mean squared (RMS) value of all 15×15 patches of the noise $\mathbf{n}$ added to each clean image. Likewise, we also show the approximated p.d.f of the *local-remainder-noise-RMS*, referring to $\hat{\mathbf{n}}$. Observe that the patch-RMSE obtained by Ours-MSE and by PSCGAN approximately follow the same p.d.f shape but with different mean and standard deviation. Moreover, the p.d.fs of the local-noise-RMS and of the patch-RMSE obtained by PSCGAN are distant, the mean of the former being much larger. Lastly, the p.d.f of the local-remainder-noise-RMS obtained by PSCGAN cannot be distinguished from that of the local-noise-RMS, while the one obtained by Ours-MSE can, especially in lower noise levels. These results suggest that noise elimination is attained by PSCGAN even locally, which means that it is stable in the sense that it generally does not produce improper local details.

Next, we question whether the remainder noise is normally distributed. We use PSCGAN to denoise each image in the FFHQ test set and perform D'Agostino and Pearson's normality test[1] [6] on all of the resulting remainder noise images (2000 noise images, each of size 128×128). In addition, for each remainder noise image we extract randomly chosen 15×15 patches and also patches that correspond to the largest patch-RMSE values (20 of each, for a total of $20 \cdot 20 \cdot 2000$ patches), and assess if they are normally distributed as well. We find that PSCGAN successfully passes all tests in all configurations, with a p-value > 0.05 with high confidence. This shows that PSCGAN's remainder noise is normally distributed both locally and globally.

5. Summary

In this work we revisit the image denoising task and focus on producing visually pleasing images, as opposed to distortion based methods that target best PSNR. Our strategy relies on the perceptual quality and distortion guarantees of posterior sampling, and a novel design of a CGAN to meet these needs. We introduce a new constraint to the CGAN framework that alleviates its difficulty to train in the case of high dimensional distributions, where each input has only one corresponding source example. We propose novel encoder-decoder denoiser architecture and training method, leading to denoised images with high perceptual quality and acceptable distortion.

[1] A normally distributed random variable should have a p-value greater than a threshold α. We use $\alpha = 0.05$, and in this case a realization of such a variable should pass the test with 95% confidence.

6. Acknowledgements

We thank Bahjat Kawar for his contribution to the development of this paper. We also thank Gennadi Zaidsher and the IT team of Technion's CS Department for providing the resources conducting this research.

References

[1] Jonas Adler and Ozan Öktem. Deep bayesian inversion. *arXiv preprint arXiv:1811.05910*, 2018. 1, 2

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223. PMLR, 2017. 1, 2, 11

[3] Yuval Bahat and Tomer Michaeli. Explorable super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 1

[4] David Berthelot, Peyman Milanfar, and Ian Goodfellow. Creating high resolution images with a latent adversarial generator. *arXiv preprint arXiv:2003.02365*, 2020. 1, 2, 3, 5, 11

[5] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. 1, 3, 7, 8

[6] Ralph D'Agostino and E. S. Pearson. Tests for departure from normality. empirical results for the distributions of $b2$ and $\sqrt{b1}$. *Biometrika*, 60(3):613–622, 1973. 8

[7] Ratnadeep Dey, Debotosh Bhattacharjee, and Mita Nasipuri. Image denoising using generative adversarial network. In J. K. Mandal and Soumen Banerjee, editors, *Intelligent Computing: Image Processing Based Applications*, volume 1157, pages 73–90. Springer Singapore, 2020. 1

[8] Nithish Divakar and R. Venkatesh Babu. Image denoising via cnns: an adversarial approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, July 2017. 1

[9] WA Falcon et al. Pytorch lightning. https://github.com/PyTorchLightning/pytorch-lightning, 2019. 13

[10] Ian Goodfellow et al. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014. 1, 2

[11] Ishaan Gulrajani et al. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5767–5777. Curran Associates, Inc., 2017. 1, 3

[12] 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. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. 7

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. 2

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2018. 11

[15] Tero Karras et al. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 1, 2, 11

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 1, 2, 5, 7, 11

[17] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017. 13

[18] Mario Lucic et al. Are gans created equal? a large-scale study. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 700–709. Curran Associates, Inc., 2018. 1

[19] Alexander Mathiasen and Frederik Hvilshøj. Fast fréchet inception distance. *arXiv preprint arXiv:2009.14075*, 2020. 7

[20] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2016. 2

[21] Sachit Menon et al. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 1

[22] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1

[23] O. Ronneberger, P.Fischer, and T. Brox. U-net: convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. 2, 11

[24] Maximilian Seitzer. pytorch-fid: fid score for pytorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.1.1. 7

[25] Yang Song et al. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1

[26] Rini Thakur, R.N. Yadav, and Lalita Gupta. State-of-art analysis of image denoising methods using convolutional neural networks. *IET Image Processing*, 13, 08 2019. 1

[27] Chunwei Tian et al. Deep learning on image denoising: an overview. *Neural Networks*, 131:251 – 275, 2020. 1

[28] Francesco Tonolini et al. Variational inference for computational imaging inverse problems. *arXiv preprint arXiv:1904.06264*, 2020. 1

[29] Gregory Vaksman, Michael Elad, and Peyman Milanfar. Lidia: lightweight learned image denoising with instance adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2020. 1

[30] Jay Whang, Erik M. Lindgren, and Alexandros G. Dimakis. Approximate probabilistic inference with composed flows. *arXiv preprint arXiv:2002.11743*, 2020. 1

[31] Dingdong Yang et al. Diversity-sensitive conditional gener-

ative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2019. 2

[32] Fisher Yu et al. Lsun: construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5, 12

[33] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 1

[34] Kai Zhang et al. Beyond a gaussian denoiser: residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 1, 5

[35] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 1

Appendices

A. Generator (Denoiser) Architecture

Inspired by StyleGAN2 [15] and UNet [23], our conditional generator (denoiser) is an encoder-decoder deep neural network, as shown in Figure 5. The decoder builds the output image scale by scale by adding residual information at each stage to the up-sampled RGB output of the previous one. This approach was proposed in StyleGAN2 to sidestep the shortcomings of progressive growing [14, 15], a training methodology in which the number of layers in the generator and the discriminator increases during training. In the newly proposed scheme, both are trained end-to-end with all resolutions included, which significantly eases the training procedure but still enforces the decoder to progressively synthesize the output image stage-by-stage by adding resolution-specific details at each level. The encoder is analogous to a drip irrigation system. It consists of a main pipeline that has several exits to independent convolutional neural networks (CNNs), denoted as *Drips*, each of which reinforces its neighboring decoder block with low receptive field information. The main pipeline is a deep CNN with high receptive field that progressively encodes the input image. This method alleviates the task of the decoder, especially at higher scales where original pixel locations are crucial for distortion performance.

Since our denoiser is stochastic and meant to sample from the posterior distribution, it can be considered as a mapping from the latent distribution of the random noise to the posterior, as in all GAN based sampling solutions. Instead of injecting a single noisy tensor to the first layer of our model, we inject noise at each scale of the decoder. These convolutional layers operate as follows: for a given layer with c input channels $\{x_i\}_{i=1}^{c}$ and a random noise input z of the same size, the resulting output of the layer (before the activation function is applied) is

$$\sum_{i=1}^{c} h_i * x_i + h_{c+1} * z, \tag{8}$$

where $\{h_i\}_{i=1}^{c+1}$ are the convolutional kernels of the layer (considering only one block of kernels that leads to one output feature map). If one further assumes that $h_{c+1} = \alpha$ (a 1-by-1 kernel) for some scaling factor α, this boils down to the noise injection scheme of StyleGAN [16] (each resulting feature map corresponds to a different scaling factor). In addition, if one forces the scaling factors of all feature maps to be equal, this becomes the noise injection scheme of StyleGAN2 [15]. Thus, our scheme enlarges the hypothesis set of the convolution operating on z. We incorporate this idea by concatenating each noise injection to the next convolutional layer's input. Consult Figure 5 for clarifications.

B. Latent Adversarial Generator Implementation Details

Expression (6) is the originally proposed optimization task of LAG [4]. Yet, the SISR results presented in LAG's paper are achieved with a tweaked version,

$$\min_{\theta} \sup_{f \in L_1} \mathbb{E}_{\mathbf{x}} \left[f(\mathbf{x}, 0) \right] - \mathbb{E}_{\mathbf{g}_\theta, \mathbf{y}} \left[f(\mathbf{g}_\theta, R(\mathbf{g}_\theta, \mathbf{y})) \right] \tag{9}$$
$$+\lambda \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\| P(\mathbf{x}, 0) - P(G_\theta(0, \mathbf{y}), R(G_\theta(0, \mathbf{y}), \mathbf{y})) \|_2^2 \right],$$

in which, instead of $\mathbf{y}$, the critic receives $R(\mathbf{g}_\theta, \mathbf{y})$ as a second input, the pixel-wise absolute difference between the degraded image $\mathbf{y}$ and the corresponding degraded version of the generated image $\mathbf{g}_\theta|\mathbf{y}$. Such a tweak could also be adopted in the case of image denoising, for instance by defining $R(\mathbf{g}_\theta, \mathbf{y})$ to be the absolute difference between $\mathbf{x}$, the clean source of $\mathbf{y}$, and the corresponding denoised image $\mathbf{g}_\theta|\mathbf{y}$. However, such a revision deviates the posterior sampling goal, since the Wasserstein-1 distance [2] would consequently measure the deviation between the distributions $\mathbb{P}_{\mathbf{x}}$ and $\mathbb{P}_{\mathbf{x}|R(\mathbf{g}_\theta,\mathbf{y})}$. We leave this, possibly beneficial, approach for future research.

In PSCGAN the critic receives $\mathbf{y}$ as its second input, and thus, for a fair comparison, we implement LAG in the same fashion instead of applying the aforementioned tweak. Our choice also aligns with optimization task (6) (since the critic receives $\mathbf{y}$) and therefore leads to a more direct evaluation of it.

It is important to note that expression (6) is highly dependent on the choice of $P(\cdot, \cdot)$. While many choices are possible, we find that in our case choosing $P(x, y) = x$ leads to superior results, both in the FID and the PSNR performance measures. $P(x, y) = x$ means that we measure the distortion between $\mathbf{x}$ and $G_\theta(0, \mathbf{y})$ in expression (6) with the MSE loss, operating in the RGB pixel space of the image instead of operating in an intermediate feature space. While the sampled images attained at $\sigma_{\mathbf{z}} = 1$ should achieve higher perceptual quality (regardless of the choice of $P(\cdot, \cdot)$), this choice, quite conveniently, also allows for a fair PSNR comparison between PSCGAN and LAG, since the images produced by LAG at $\mathbf{z} = 0$ are now directly aimed to optimize the MSE. We refer to this version of LAG as *Ours-LAG* in the experimental evaluations, so as to emphasize that our choices deviate quite significantly from the original implementation of LAG (a different inverse problem, different architectures, and different loss).

C. Training

C.1. Data Splits

- FFHQ [16] thumbnails contains 70,000 images. We use images 3000-4999 for testing and the rest for training.

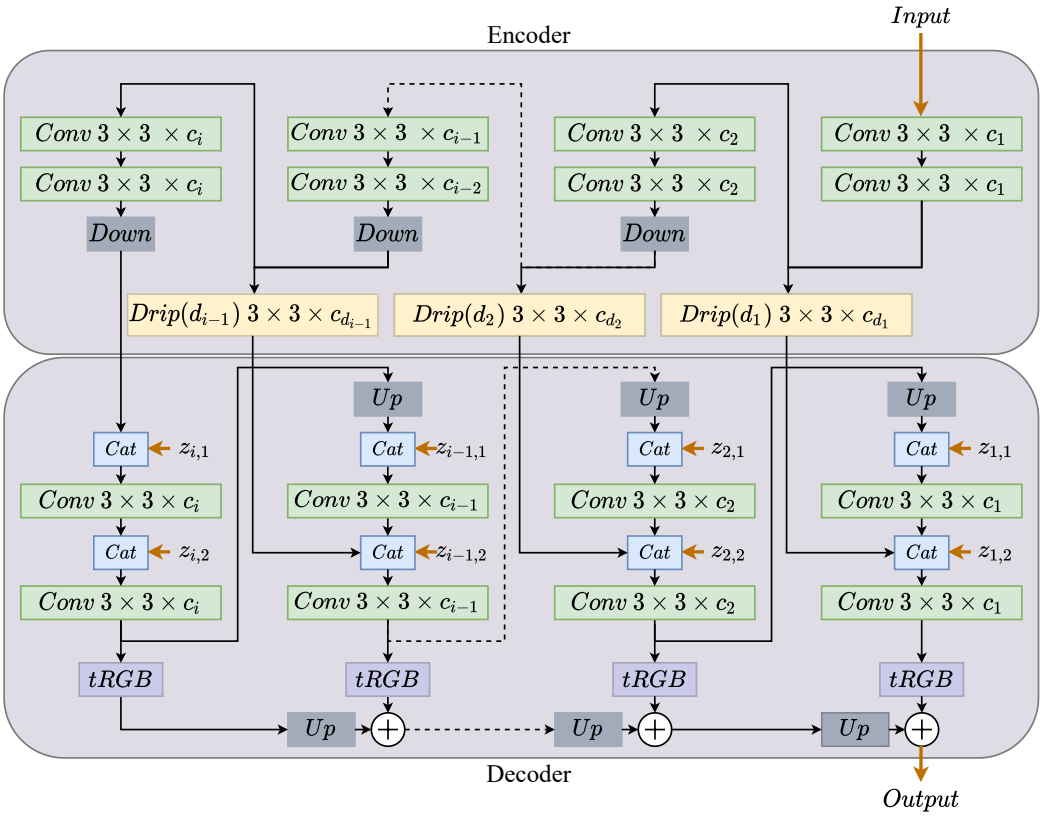

Figure 5: Our proposed generator architecture. An input noisy image is passed through an encoder of i doubly-blocked convolutional layers and downsampled after each (except for the first block). The downsampling operation is performed by a stride of 2 in the preceding convolution layer. The result of each doubly-blocked layer is then passed through a Drip, which is a feed-forward CNN (in the figure, d_k and c_{d_k} denote the number of layers and the number of output channels of each layer in drip k, respectively). Each of these drips extracts features that are limited to a certain receptive field, which are then passed to the neighboring decoder block through concatenation. This further assists in reconstructing the RGB result of the corresponding scale, especially at higher scales. The decoder builds the reconstructed image scale by scale, using features aggregated from previous layers of the decoder and from the drip injections. Noise injections are performed in the decoder's pipeline, where a noisy "image", denoted as $z_{k,1}$ and $z_{k,2}$ for each $1 \leq k \leq i$, is concatenated as another feature map of the next layer's input. All convolutional layers, except for $tRGB$, are coupled with Leaky ReLU activation functions with a slope of $\alpha = 0.2$ for negative values. $tRGB$ is a simple convolution operation with output channels being equal to the number of channels of the input image (3 for RGB images). All up-sampling operations are performed with nearest-neighbor interpolation.

- LSUN Bedroom [32] contains 3,037,042 images. We randomly pick 100,000 and 4,000 non-overlapping images for training and testing, respectively.
- LSUN Church [32] contains 126,227 images. We randomly pick 100,000 and 4,000 non-overlapping images for training and testing, respectively.

C.2. Preprocessing

Our model assumes an input image of size 128×128, and since the images in both LSUN data sets are of larger size in both axes, we first center crop each image while keeping the smaller dimension fixed, and then resize the resulting square image to the desired size through interpolation. All images in the FFHQ thumbnails data set are already of size 128×128, and therefore do not require augmentation. Finally, we use random horizontal flip during training in all data sets.

C.3. Hyperparameters

PSCGAN (and consequently PSCGAN-A) is trained with the default hyperparameters given in Algorithm 1. Note that we evaluate the penalty term on the first $PB = 8$ samples of each mini-batch of $B = 32$ samples, and ap-

proximate $\mathbb{E}\left[G_\theta(\mathbf{z}, \mathbf{y})|\mathbf{y}\right]$ by averaging $M = 8$ generated samples for a given noisy image $\mathbf{y}$. While this choice of M may seem too small to evaluate the expectation of the posterior, it is sufficient to demonstrate the novelty of PSCGAN.

All other algorithms are also trained with a batch size of 32 and the Adam optimizer [17]. LAG is trained with a learning rate of $2.5 \cdot 10^{-4}$, and the Adam hyperparameters are $\beta_1 = 0, \beta_2 = 0.99$ (similar to PSCGAN). DnCNN and Ours-MSE are trained to optimize the MSE loss, the former with a learning rate of 10^{-3} and the latter with a learning rate of $5 \cdot 10^{-4}$. The Adam hyperparameters for both methods are $\beta_1 = 0.9, \beta_2 = 0.99$.

The full implementation of all methods and the checkpoints that reproduce the results reported in this paper are publicly available at `https://github.com/theoad/pscgan.git`. Our implementations are based on PyTorch and PyTorch Lightning [9].

D. LSUN Data Sets' Visual Results

In Figure 6 and Figure 7 we illustrate the visual quality of several denoised images produced by our method and by other MSE based methods on the LSUN Church outdoor and the LSUN Bedroom test sets. As can be seen, our model produces denoised images with high perceptual quality, although in these data sets it is harder to notice the perceptual quality difference with the naked eye (since the images are compressed).

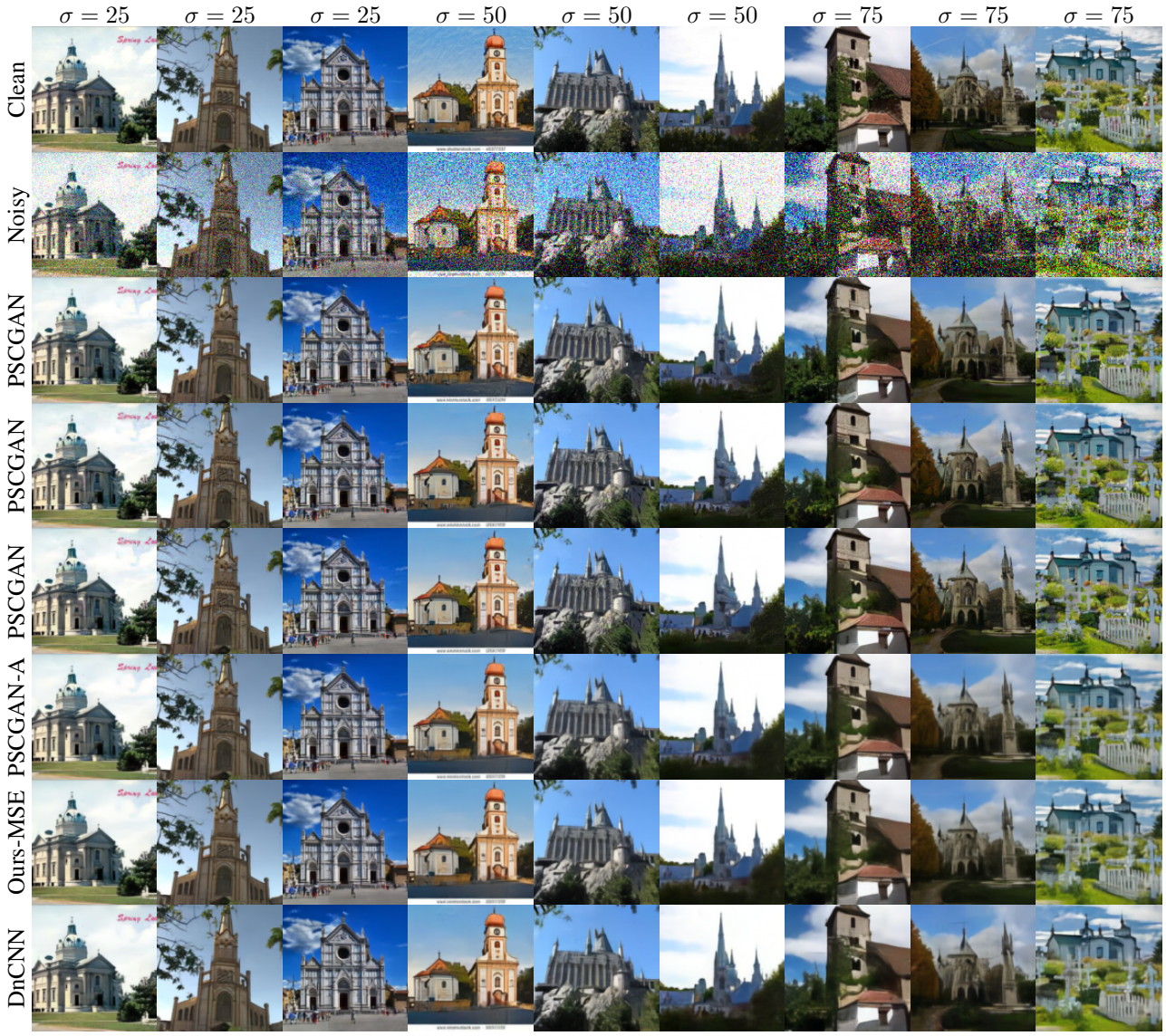

Figure 6: Denoising results on the LSUN Church outdoor test set produced by several methods. For each image we show three different outcomes of PSCGAN, each attained by injecting noise with standard deviation of $\sigma_{\mathbf{z}} = 0.75$. In this case, PSCGAN-A averages 64 instances of PSCGAN, where each instance is attained with $\sigma_{\mathbf{z}} = 1$ at inference time. Each model is trained on the LSUN Church outdoor training set to denoise a specific noise level ($25, 50$ or 75).

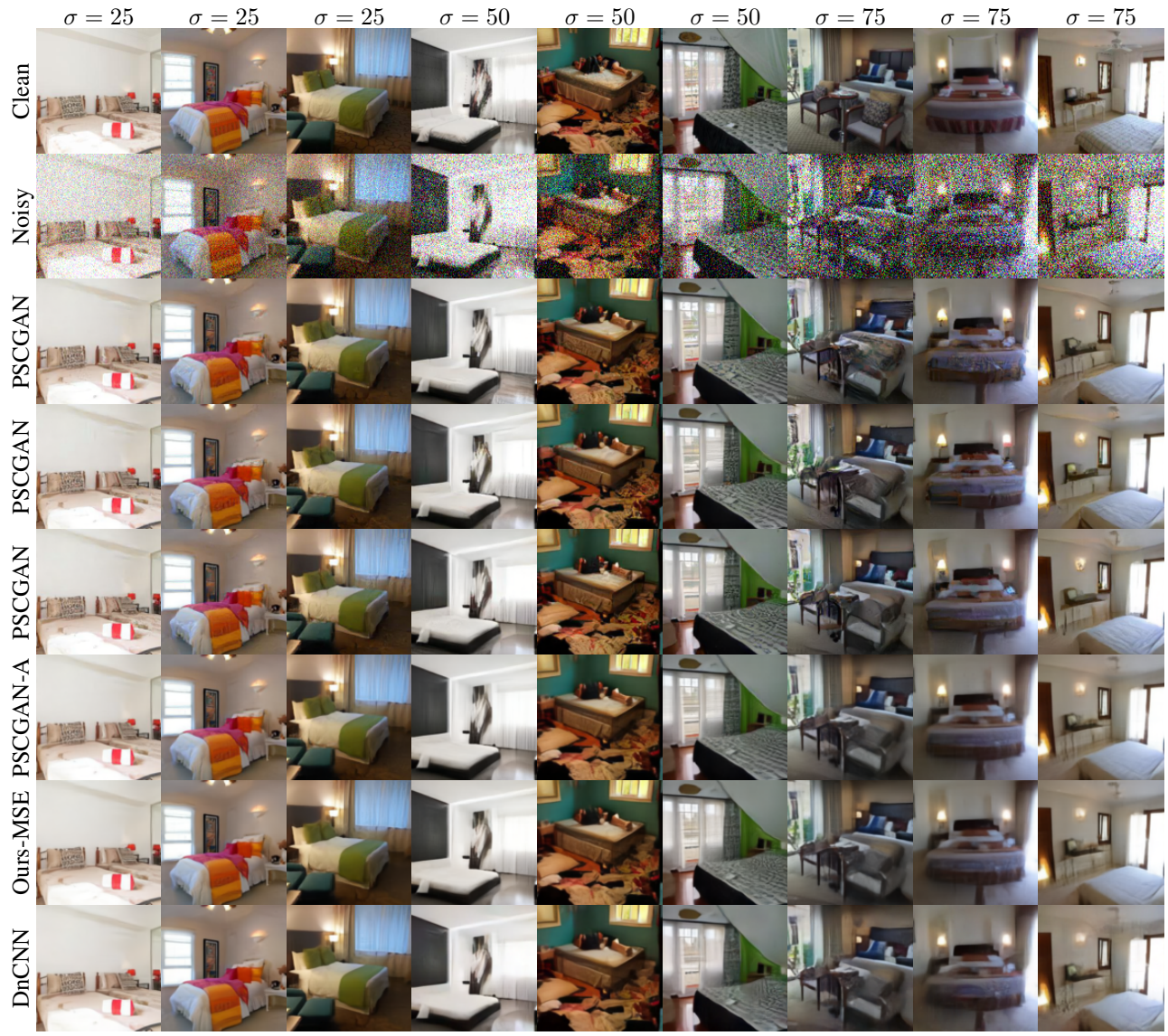

Figure 7: Denoising results on the LSUN Bedroom test set produced by several methods. For each image we show three different outcomes of PSCGAN, each attained by injecting noise with standard deviation of $\sigma_{\mathbf{z}} = 0.75$. In this case, PSCGAN-A averages 64 instances of PSCGAN, where each instance is attained with $\sigma_{\mathbf{z}} = 1$ at inference time. Each model is trained on the LSUN Bedroom training set to denoise a specific noise level ($25, 50$ or 75).