Reasons for the Superiority of Stochastic Estimators over Deterministic Ones: Robustness, Consistency and Perceptual Quality

Guy Ohayon [1] **Theo Adrai** [1] **Michael Elad** [1] **Tomer Michaeli** [2]

Abstract

Stochastic restoration algorithms allow to explore the space of solutions that correspond to the degraded input. In this paper we reveal additional fundamental advantages of stochastic methods over deterministic ones, which further motivate their use. First, we prove that any restoration algorithm that attains perfect perceptual quality and whose outputs are consistent with the input must be a posterior sampler, and is thus required to be stochastic. Second, we illustrate that while deterministic restoration algorithms may attain high perceptual quality, this can be achieved only by filling up the space of all possible source images using an extremely sensitive mapping, which makes them highly vulnerable to adversarial attacks. Indeed, we show that enforcing deterministic models to be robust to such attacks profoundly hinders their perceptual quality, while robustifying stochastic models hardly influences their perceptual quality, and improves their output variability. These findings provide a motivation to foster progress in stochastic restoration methods, paving the way to better recovery algorithms.

1. Introduction

Image restoration has a central role in imaging sciences, enabling much of the imaging revolution we see today. Modern restoration methods allow to squeeze out ever fantastic quality from ever smaller sensors (e.g. on mobile devices), which are prone to severe noise, blur, limited resolution, and color degradations. Traditionally, most image restoration algorithms were deterministic: providing one restored image for a given degraded input. Stochastic restoration algorithms, which started to receive notable attention in recent years (Lugmayr et al., 2021; 2022; Bahat & Michaeli, 2020; Ohayon et al., 2021; Kawar et al., 2022; 2021b;a; Saharia et al., 2021; Kadkhodaie & Simoncelli, 2021; Song & Ermon, 2019), instead sample from a distribution conditioned on the degraded input. Yet, beyond their core ability to explore the space of possible solutions, the advantages of stochastic methods over deterministic ones are still unclear. For instance, a fundamental mystery is whether stochastic restoration algorithms are theoretically superior to deterministic ones in terms of the achievable perceptual quality, the robustness to adversarial attacks, and the faithfulness of the results to the measurements (a property we refer to as consistency).

In this paper we provide novel justifications for using stochastic restoration methods. First, we show that any estimator that generates consistent restorations and attains perfect perceptual quality must be sampling from the posterior distribution. An immediate, yet significant, consequence is that if a deterministic restoration algorithm is consistent then it cannot attain perfect perceptual quality[1]. Somewhat strangely, despite this theoretical limitation, deterministic restoration algorithms are known to be able to attain high perceptual quality, or at least produce images with high precision (Wang et al., 2018; Blau & Michaeli, 2018; Ledig et al., 2017; Yu et al., 2018b;b; Zhao et al., 2021; Yi et al., 2020). While this may seem as a contradiction, we argue that it is not. We show empirically that, in order to cope with the lack of output diversity, such deterministic estimators adopt an erratic[2] output behavior in an attempt to "fill up" the space of all possible natural images, so as to minimize the distance between the distribution of their outputs and the distribution of natural images. As a result, such estimators are highly sensitive to small changes in their input. Indeed,

[1]Faculty of Computer Science, Technion, Haifa, Israel [2]Faculty of Electrical and Computer Engineering, Technion, Haifa, Israel. Correspondence to: Guy Ohayon <ohayonguy@campus.technion.ac.il>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1]Unless the posterior assigns nonzero probability to only one reconstruction for every input, in which case the inverse problem is non-ambiguous and it is possible to restore images with zero error.

[2]By erratic we mean that the restoration algorithm is not robust to an input adversarial attack: a small, unnoticeable change in the input degraded image may lead to an unreasonably large change in the output estimated image. I.e., the value of Equation (5) is too high.

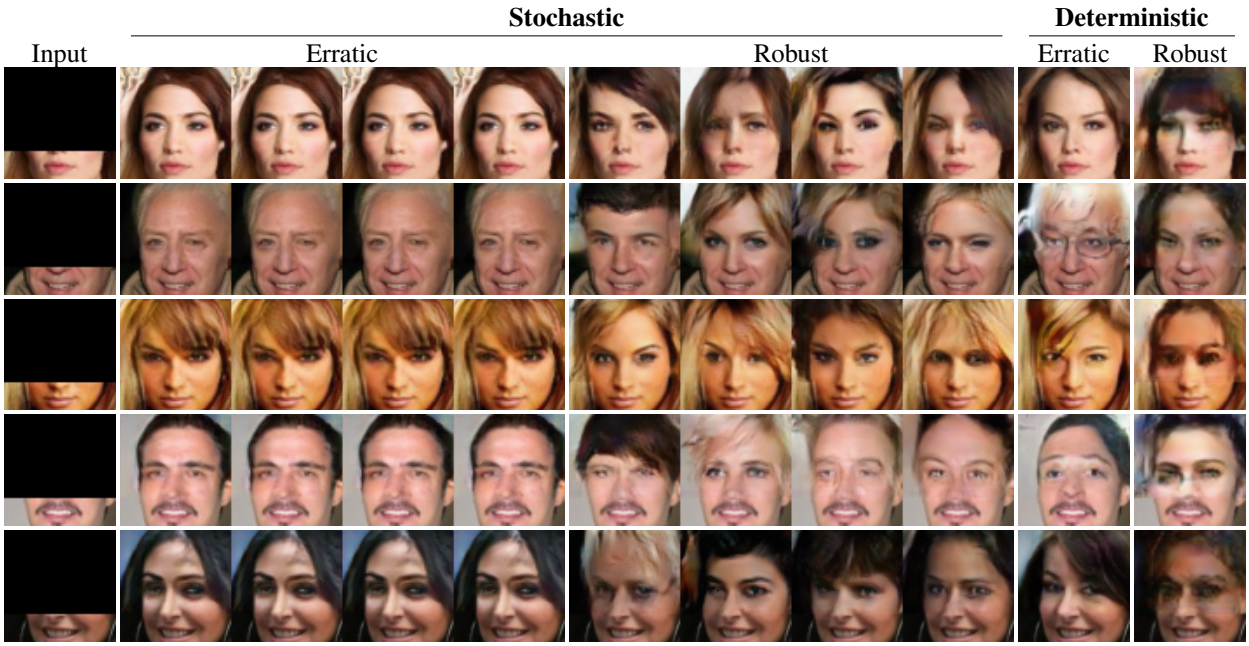


Figure 1. Output samples from several consistent restoration algorithms that solve the image inpainting task on the CelebA data set. The erratic stochastic and deterministic algorithms are trained solely with a GAN loss. As can be seen, the erratic stochastic estimator barely produces output variability per input, which reveals a tendency of mode collapse of CGANs (Goodfellow et al., 2014; Isola et al., 2017; Mathieu et al., 2016; Yang et al., 2019; Ohayon et al., 2021). The robust algorithms are trained to also defend against adversarial attacks by adding Equation (7) to the GAN objective. Robustifying the deterministic algorithm deteriorates its perceptual quality, while doing so for the stochastic algorithm preserves this quality and significantly improves its output variability. Refer to Table 1 for quantitative evaluation.

it has been experimentally observed that a visually unnoticeable input perturbation leads to unreasonable changes in the output in this kind of estimators (Choi et al., 2019; 2021; Antun et al., 2020; Yan et al., 2022; Raj et al., 2020; Gandikota et al., 2022; Yu et al., 2022; Choi et al., 2020; Yue et al., 2021). Furthermore, we illustrate that making deterministic methods more robust to adversarial attacks causes the quality of their outputs to deteriorate (see Figure 1). Hence, our findings provide a novel explanation for the increased vulnerability of high perceptual quality deterministic estimators to adversarial attacks.

Since robustness for deterministic estimators comes at the expense of perceptual quality, it is appealing to resort to stochastic estimators. Such estimators can produce many restorations for any given input, so that they seemingly do not need to be highly sensitive to their inputs. Indeed, we show that as opposed to deterministic methods, robustifying stochastic algorithms hardly impairs their perceptual quality. Interestingly, some stochastic methods do exhibit a relatively high sensitivity to their inputs, but this is indicative of mode-collapse (they effectively behave like deterministic methods). As we illustrate in Figure 1, in such cases robustification improves the output diversity without hampering perceptual quality. All at all, unlike deterministic algorithms, stochastic methods allow to attain consistent, high perceptual quality restorations, while remaining robust to adversarial attacks.

The contributions of this paper are the following: 1) We prove that a posterior sampler is the only consistent restoration algorithm that attains perfect perceptual quality; 2) While deterministic restoration algorithms could still be consistent and attain very high perceptual quality, we reveal that they behave erratically in order to do so, i.e., we empirically show that for deterministic estimators, high perceptual quality comes with the cost of vulnerability to input adversarial attacks; 3) We develop a novel notion of robustness for stochastic estimators; and 4) We show that robustifying deterministic algorithms deteriorates their perceptual quality, while doing so for stochastic algorithms only improves their output diversity. Hence, we empirically confirm that stochastic restoration algorithms can be robust and attain high perceptual quality at the same time. We thus suggest robustness as an effective regularization for promoting meaningful diversity in stochastic restoration methods. Please refer to Figure 2 for a flow chart that clarifies the main conclusions of this paper.

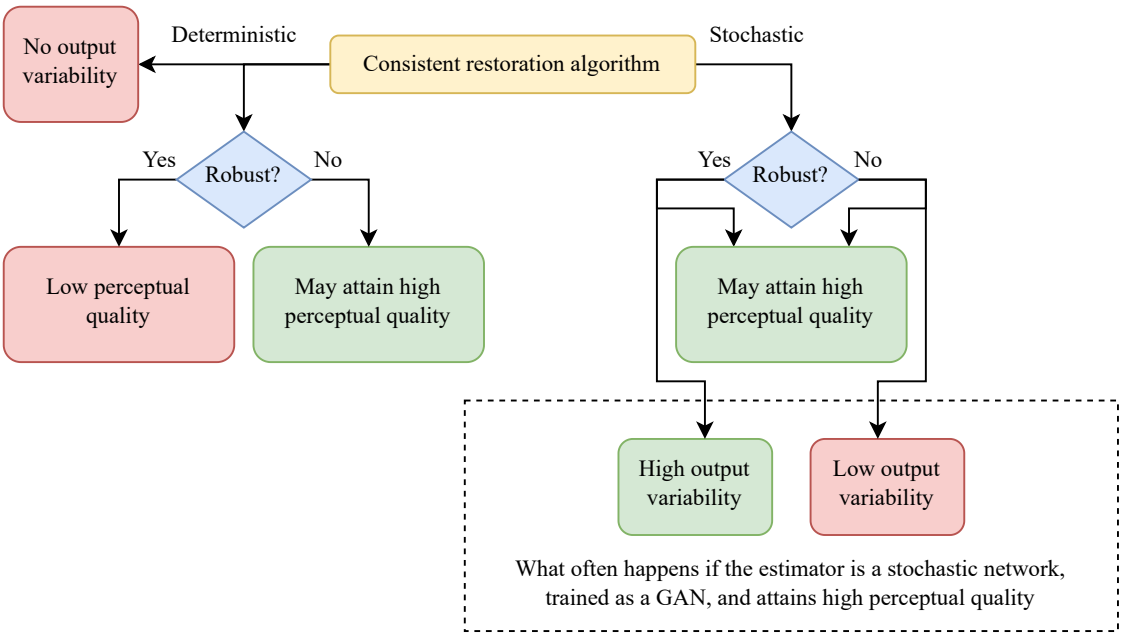


Figure 2. The main conclusions of this paper, summarized in a flow chart. We only consider perfectly consistent restoration algorithms. A deterministic restoration algorithm has, by definition, no output variability. Such an algorithm attains low perceptual quality if it is robust, and may attain high perceptual quality otherwise. A stochastic restoration algorithm may always attain high perceptual quality, whether it is robust or not. In the case where the stochastic algorithm is a neural network trained as a GAN, robustifying such an algorithm improves (increases) its output variation per input. Without robustification, such an algorithm often attains low output variability, and essentially behaves as a non-robust (erratic), deterministic algorithm.

2. Problem setting and preliminaries

We consider a natural image $x \in \mathbb{R}^{n_x}$ as a realization of a multivariate random variable X with probability density function p_X. A degraded version $y \in \mathbb{R}^{n_y}$ is also a realization of a random vector Y which is related to X via some conditional distribution $p_{Y|X}$. In this paper we assume that the degradation is deterministic, i.e., $y = D(x)$, which implies that $p_{Y|X}(y|x) = \delta(y - D(x))$, where $D : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ is a non-invertible function. Problems such as image colorization, inpainting, demosaicing, single-image super-resolution, JPEG-deblocking, and more all follow this assumed structure. Here we focus on ill-posed inverse problems in which X cannot be retrieved from Y with zero error, i.e., $p_{X|Y}(\cdot|y)$ is not a delta function for almost every y.

Given a particular input $Y = y$, an image restoration algorithm produces an estimate $\hat{X}$ according to some distribution $p_{\hat{X}|Y}(\cdot|y)$ such that the estimate $\hat{X}$ is statistically independent of X given Y. For deterministic algorithms, $p_{\hat{X}|Y}(\cdot|y)$ is a delta function for every y, while for stochastic algorithms it is a non-degenerate distribution.

2.1. Perceptual quality

Conceptually, the perceptual quality of a restoration algorithm is a quantification of its ability to produce images that appear natural. While there are several notions of perceptual quality, we measure it as the deviation of the restorations from natural image statistics (Blau & Michaeli, 2018). Formally, we quantify the perceptual quality of an estimator $\hat{X}$ using the *perceptual index* (lower is better),

$$d(p_X, p_{\hat{X}}), \tag{1}$$

where $d(p, q)$ is a divergence between distributions (e.g., Kullback-Leibler, Wasserstein). i.e., an estimator attains perfect perceptual quality if $p_{\hat{X}} = p_X$. As discussed in (Sajjadi et al., 2018; Kynkäänniemi et al., 2019), one can measure the deviation between $p_{\hat{X}}$ and p_X via *precision* (the probability that a random sample from $p_{\hat{X}}$ falls within the support of p_X) and *recall* (the probability that a random sample from p_X falls within the support of $p_{\hat{X}}$). Such an approach to measure statistical deviation is useful since it provides a meaning to the difference between $p_{\hat{X}}$ and p_X, if exists. That is, low precision means that a sample of $\hat{X}$ may seem unnatural, while low recall implies that part of the support of p_X (possible natural images) can never be the output of $\hat{X}$. Note that $p_{\hat{X}} = p_X$ if and only if both the precision and recall are perfect (Sajjadi et al., 2018).

Importantly, the reader should not confuse high precision with high perceptual quality. A restoration algorithm might produce images that appear natural (with high precision), but this does not mean that the algorithm attains sufficient perceptual quality, since its recall might be compromised.

2.2. Consistency

Another important quality measure of a restoration algorithm is its ability to produce results that are consistent with the low quality input. Such an ability is important to consider since only perfectly consistent restorations could potentially be the high quality source image. In our deterministic degradation scenario, a restoration $\hat{x}$ is consistent if it satisfies $D(\hat{x}) = y$ (which in turn equals $D(x)$), which can be equivalently written as $p_{Y|X}(\cdot|x) = p_{Y|\hat{X}}(\cdot|x)$. We regard an algorithm as perfectly consistent (or consistent, in short) if it satisfies this property for every x. Note that a restoration algorithm that is not perfectly consistent could still be considered as such for any practical use. For instance, (Lugmayr et al., 2022; 2021) consider super-resolution (SR) methods as consistent if they satisfy $\text{PSNR}(D(\hat{X}), D(X)) \geq 45\text{dB}$.

While being an important property for restoration algorithms, apparently only recent publications provide a report of consistency (Lugmayr et al., 2022; Jo et al., 2021; Lugmayr et al., 2020; 2021; Bahat & Michaeli, 2020; Kim et al., 2015), while many previous ones do not (Wang et al., 2018; Chen et al., 2018; Ledig et al., 2017; Shi et al., 2016; Dong et al., 2015). Moreover, works that do report consistency typically deal only with stochastic image restoration. It therefore seems that consistency has been overlooked, or taken for granted by deterministic restoration algorithms.

3. Can deterministic estimators be consistent and achieve high perceptual quality?

While deterministic algorithms are being extensively used to produce high perceptual quality restorations (e.g., (Wang et al., 2018; Ledig et al., 2017; Yu et al., 2018b;b; Zhao et al., 2021; Yi et al., 2020; Galteri et al., 2017)), the following important result must be taken into account.

Theorem 3.1. *For a deterministic degradation, $y = D(x)$, an estimator $\hat{X}$ is perfectly consistent ($p_{Y|X} = p_{Y|\hat{X}}$) and achieves perfect perceptual quality ($p_{\hat{X}} = p_X$) if and only if it is the posterior sampler $p_{\hat{X}|Y} = p_{X|Y}$.*

The proof of the theorem makes simple use of Bayes' rule (see Appendix A). But despite its simplicity, this result has several important implications. (i) Since $p_{X|Y}(\cdot|y)$ is not a delta function for almost every y (the degradation is not invertible), an immediate corollary is that *a consistent deterministic algorithm can never attain perfect perceptual quality*. (ii) It is a known fact that there are cases in which the posterior sampler does not attain the lowest possible MSE (or any other distortion measure) among the estimators that achieve perfect perceptual quality (Blau & Michaeli, 2018; Freirich et al., 2021). But from Theorem 3.1 we see that *the posterior sampler is the only perfect perceptual quality estimator that is also perfectly consistent*. Thus, for perfect perceptual quality estimators, aiming for low distortion might come on the expense of inconsistency. (iii) Finally, we can conclude from Theorem 3.1 that if perfect consistency is enforced, then one can train a restoration algorithm to become a posterior sampler by *solely encouraging high perceptual quality* (e.g. with a GAN loss), without any additional loss.

4. Erratic behavior of deterministic estimators

Earlier work (Choi et al., 2019; 2021) has already identified the tendency of deterministic GAN-based restoration methods to be vulnerable to adversarial attacks. In this section we shed light on this phenomenon. The authors of both of these works hypothesize that, since GANs usually produce perceptually appealing and sharp textures, small input perturbations tend to be severely intensified in the output image. While such a claim is valid, it does not tell the full story. We argue that this phenomenon is indeed a result of attempting to attain high perceptual quality, but more specifically, doing so *with a deterministic estimator*. To attain high perceptual quality, such an estimator must adopt an erratic output behavior in order to fill up the space of possible source signals, and consequently be highly sensitive to small input perturbations (i.e., vulnerable to adversarial attacks). In other words, the problem is not related to the use of GANs, but rather to the lack of randomness. Using any non-GAN-based *deterministic* method with high perceptual quality would yield the same issue; and a GAN-based *stochastic* method with high perceptual quality can avoid this phenomenon. Refer to Appendix B for further discussion.

Let us demonstrate this phenomenon in more detail through a toy experiment. Suppose that $X = (X^{(1)}, X^{(2)})$ is uniformly distributed on a disk with radius 1.0 centered at $(0, 0)$, and let $Y = X^{(1)}$. Our goal is to estimate X based on Y (i.e., we need to estimate only $X^{(2)}$, as $X^{(1)}$ is known). In Figure 3 (the two leftmost columns in the top row) we present two *consistent* deterministic restoration algorithms of the form $\hat{X}_\alpha = (Y, \sqrt{1-Y^2}\sin(\alpha Y))$. These estimators provide a mental demonstration of the anticipated behavior of robust and non-robust deterministic restoration algorithms that strive for high perceptual quality. For an estimator of this form, notice that α controls the tradeoff between its perceptual quality and robustness: increasing α improves its perceptual quality but hinders its robustness. For instance, $\hat{X}_{\text{Erratic}}$ (with $\alpha = 50$) represents an erratic (non-robust) estimator with high perceptual quality. It "zigzags" and fills the support of p_X, which intuitively leads

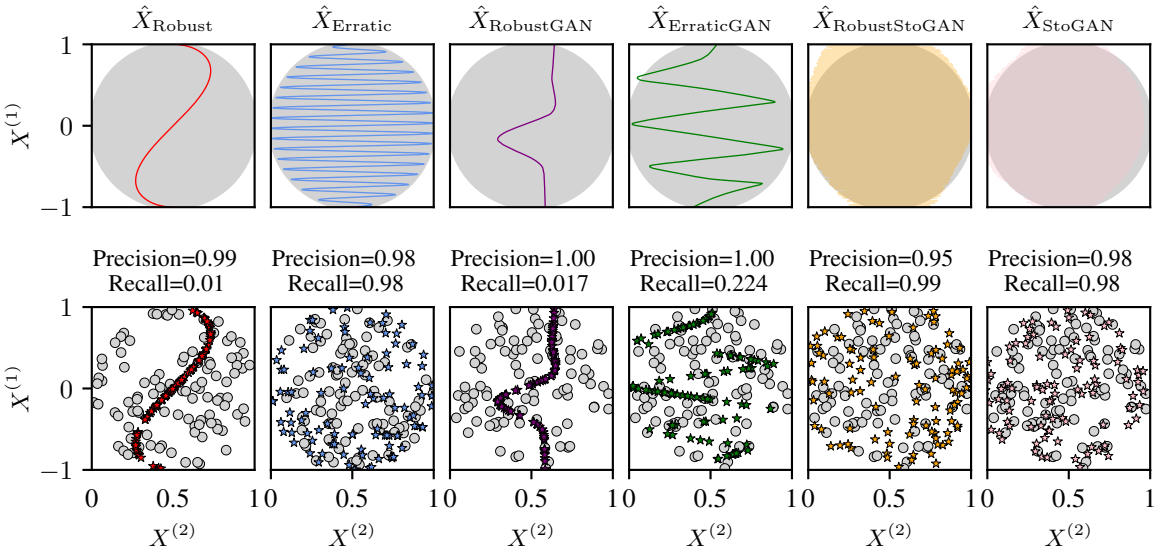


Figure 3. An illustration on a toy restoration problem of the tradeoff between robustness and perceptual quality for deterministic restoration algorithms, as well as as a demonstration that such a tradeoff does not exist for stochastic algorithms. Each column corresponds to a different estimator, with the estimator's name indicated in the title of the column. Top row: the support of the data distribution (gray disc), and the support of the output distribution of each estimator (a continuous colored line of 10,000 random samples from the output distribution of each estimator). Bottom row: 100 random samples from the data distribution (gray dots), and 100 random samples from the output distribution of each estimator (colored stars). The precision and recall (Kynkäänniemi et al., 2019) of each estimator are computed between 1000 random output samples from the output distribution of the estimator and 1000 random samples from the data distribution. Refer to Section 4 for more details.

to a smaller distance between $p_{\hat{X}}$ and p_X. $\hat{X}_{\text{Robust}}$ (with $\alpha = 1$) represents a robust estimator with low perceptual quality. Its output mildly changes when perturbing its input, so it fills a smaller region from the support of p_X. To numerically confirm that an increased erratic behavior results in higher perceptual quality, we approximate the precision and recall (Kynkäänniemi et al., 2019) of these estimators, showing that such practical statistical distance measures can be fooled by highly erratic deterministic algorithms. For an analytical confirmation, refer to Appendix C.

4.1. Perceptual quality approximation

In real world problems p_X and $p_{\hat{X}}$ are typically unknown, and their statistical discrepancy is commonly approximated by using finite sized sets of independently drawn samples from both distributions (e.g., (Heusel et al., 2017; Kynkäänniemi et al., 2019; Sajjadi et al., 2018)). In some scenarios it would be difficult or even impossible for such a statistical distance approximation algorithm to distinguish between the output distribution of a highly erratic deterministic algorithm and the distribution of the source data. To illustrate this, we randomly and independently sample 1000 points from the outputs of $\hat{X}_{\text{Robust}}$ and $\hat{X}_{\text{Erratic}}$ and approximate their perceptual quality via precision and recall (Kynkäänniemi et al., 2019) (with $k = 5$) and without using feature projection. In the bottom row of Figure 3 we present 100 random output samples of each algorithm (colored stars), 100 random samples from p_X (gray dots), as well as the precision and recall results. As expected, all the algorithms attain almost perfect precision. For the erratic estimator $\hat{X}_{\text{Erratic}}$, a human observer might be fooled to believe that its samples are actually drawn from p_X. Indeed, it attains almost perfect recall as well, and consequently almost perfect perceptual quality according to these metrics. However, the robust algorithm $\hat{X}_{\text{Robust}}$ attains a much lower recall, and one can easily distinguish between the distribution of its output samples and the samples from the data distribution. This illustrates what one might expect when robustifying a deterministic estimator: its perceptual quality would be limited by a lower recall.

These toy restoration algorithms are hand-crafted to visually demonstrate our hypothesis, so let us confirm that this behavior occurs for practical estimators as well.

4.2. Deterministic GAN estimators

We trained a neural network $\hat{X}_{\text{ErraticGAN}} = G_\theta(Y)$ as a GAN (Goodfellow et al., 2014) to solve the aforementioned toy problem (we omit the use of θ from now on). We evaluated the precision and recall in the same fashion as before, and present the results in Figure 3. As shown, $\hat{X}_{\text{ErraticGAN}}$

is indeed erratic, which aligns with our argument that this would be the behavior of a deterministic algorithm when attempting to attain high perceptual quality. I.e., $\hat{X}_{\text{ErraticGAN}}$ becomes highly sensitive to small input perturbations.

Let us observe the effect of robustifying such an estimator. The sensitivity of a deterministic algorithm $G(Y)$ at $Y = y$ can be measured by

$$r_{G,\epsilon}(y) = \max_{\delta:\|\delta\|_2\leq\epsilon} \|G(y) - G(y+\delta)\|_2^2. \tag{2}$$

That is, $r_{G,\epsilon}(y)$ is the extent to which a small input perturbation of y leads to a change in the output. From here, we can measure the *robustness* of a deterministic estimator by averaging $r_{G,\epsilon}(y)$ for all y's, leading to

$$R_{G,\epsilon} = \mathbb{E}\left[r_{G,\epsilon}(Y)\right]. \tag{3}$$

In Figure 3 we present an additional GAN based algorithm $\hat{X}_{\text{RobustGAN}}$, which was trained with $R_{G,\epsilon}$ as a regularizer (refer to Appendix E.1 for full training details of these estimators). As expected, $\hat{X}_{\text{RobustGAN}}$ is a much more stable algorithm, but has a lower perceptual quality (lower recall). As hypothesized, $\hat{X}_{\text{ErraticGAN}}$ attains a higher recall than $\hat{X}_{\text{RobustGAN}}$, which can also be confirmed visually as it results in a more erratic behavior and passes through more regions in $\text{Supp}(p_X)$. We therefore confirm again that the robustness of a deterministic estimator should hinder its perceptual quality (and more specifically, its recall). These experiments demonstrate again that robustness of deterministic estimators comes at the cost of lower perceptual quality.

5. Robustness of stochastic estimators

Deep-learning based recovery algorithms are known to be sensitive to miniature and visually unnoticeable input perturbations, e.g. in single-image super-resolution (Choi et al., 2019; 2021; 2020; Yue et al., 2021), deblurring (Gandikota et al., 2022), rain removal (Yu et al., 2022), and denoising (Yan et al., 2022). Studying the stability of image restoration methods is important and sometimes critical (e.g. in medical imaging for diagnosis), as only robust algorithms are able to provide trustworthy estimations. Interestingly, to the best of our knowledge, all the existing publications that discuss robustness in image restoration only consider deterministic solutions.

To discuss the robustness of an arbitrary (not necessarily deterministic) estimator, we first need to generalize the notion of robustness to the stochastic case. We extend Equation (3) by thinking of robustness as the maximal deviation in $p_{\hat{X}|Y}(\cdot|y)$ that can occur due to a small change in y. Formally, we define the sensitivity of a possibly stochastic estimator $\hat{X}$ at $Y = y$ by

$$r_{\hat{X},\epsilon}(y) = \max_{\delta:\|\delta\|_2\leq\epsilon} W_2^2(p_{\hat{X}|Y}(\cdot|y), p_{\hat{X}|Y}(\cdot|y+\delta)), \tag{4}$$

where W_2 is the Wasserstein-2 distance between distributions. As before, we define the overall robustness of $\hat{X}$ by

$$R_{\hat{X},\epsilon} = \mathbb{E}\left[r_{\hat{X},\epsilon}(Y)\right]. \tag{5}$$

To see why Equation (5) serves as a natural extension of Equation (3), we show that both of these definitions are equivalent when $\hat{X}$ is a deterministic estimator. Indeed, if $\hat{X} = G(Y)$ for some deterministic function G, $p_{\hat{X}|Y}(\cdot|y)$ is a delta function for any y, so the W_2 distance in $r_{\hat{X}}(y)$ measures the deviation between two Dirac measures and therefore becomes a simple L_2 norm between two vectors, i.e., $W_2(p_{\hat{X}|Y}(\cdot|y), p_{\hat{X}|Y}(\cdot|y+\delta)) = \|G(y) - G(y+\delta)\|_2$.

Practically, measuring the Wasserstein-2 distance between distributions, and not to mention minimizing it, is a highly difficult task. We therefore offer an alternative in the case where the estimator is a neural network and the stochasticity is acquired by using an input random seed (as done in all GANs). Let $\hat{X} = G(Y, Z)$ be a neural network, where Y is the input and Z is some random seed that follows any type of distribution. We propose to measure the sensitivity of $\hat{X}$ at $Y = y$ by considering its average sensitivity over random draws of Z, i.e.,

$$\tilde{r}_{\hat{X},\epsilon}(y) = \max_{\delta:\|\delta\|_2\leq\epsilon} \mathbb{E}\left[\|G(y,Z) - G(y+\delta,Z)\|_2^2\right]. \tag{6}$$

In words, Equation (6) measures the extent to which a small input perturbation changes the output with *the same random seed*, averaged over many seeds. Finally, we approximate the overall robustness of $\hat{X}$ via

$$\tilde{R}_{\hat{X},\epsilon} = \mathbb{E}\left[\tilde{r}_{\hat{X},\epsilon}(Y)\right]. \tag{7}$$

Observe that here as well we naturally extend the definition of robustness for deterministic estimators. As we move from a stochastic to a deterministic algorithm, the variance of the seed Z drops to zero, and the two definitions coincide. Moreover, this is a rational approximation method for the stochastic case since Equation (7) upper bounds Equation (5) (see proof in Appendix D). Hence, minimizing Equation (7) forces Equation (5) to also minimize.

To illustrate that stochastic algorithms can simultaneously be robust and attain high perceptual quality, we trained a neural network $G(Y, Z)$ to solve the toy problem presented in Section 4, where $Z \sim \mathcal{N}(0, I)$ is an input random seed that allows the outputs to vary for each input Y. As before, we trained two consistent algorithms: $\hat{X}_{\text{StoGAN}}$ and $\hat{X}_{\text{RobustStoGAN}}$, where the former is trained solely with a GAN loss, and the latter is also regularized to be robust using Equation (7) (see Appendix E.1 for full training details). Following a similar procedure to Section 4.2, we randomly sample 1000 outputs from each estimator (by sampling 1000

random inputs and coupling each input with one random seed) and evaluate precision and recall. Unsurprisingly, both $\hat{X}_{\text{StoGAN}}$ and $\hat{X}_{\text{RobustStoGAN}}$ lead to precision and recall scores above 0.95, which confirms our hypothesis that a consistent stochastic algorithm can maintain high perceptual quality even when it is robust to input adversarial attacks.

It is important to note that the whole discussion on robust restoration algorithms with high perceptual quality is relevant only when assuming that a minor perturbation in y does not lead to a large change in the posterior distribution $p_{X|Y}(\cdot|y)$ when dealing with natural images. As discussed in Section 4, a consistent deterministic estimator with high perceptual quality attempts to become a posterior sampler by behaving erratically, and thus we propose to use stochastic estimators instead (such as a posterior sampler). See Appendix B for more details on this subject.

Table 1. Quantitative evaluation of the CelebA image inpainting algorithms described in Section 6.1.1. Robstifying the deterministic restoration algorithm significantly deteriorates its recall and FID performance. Doing so for the stochastic algorithm only slightly hinders its performance (with the same AI-PSNR), while also improving its output variability (high per-pixel std). Refer to Section 6.1.2 for further analysis of these results.

	Stochastic		**Deterministic**	
Metric	Erratic	Robust	Erratic	Robust
FID$^{\downarrow}$	14.10	19.27	14.37	39.09
Precision$^{\uparrow}$	0.670	0.634	0.670	0.533
Recall$^{\uparrow}$	0.388	0.268	0.382	0.022
Per-pixel std$^{\uparrow}$	0.007	0.105	0	0
Robustness$^{\uparrow}$	6.884	26.19	3.544	23.69
AI-PSNR	34.47	32.95	34.18	32.98

6. Experiments on natural images

6.1. Demonstrations with GANs

We train several types of GAN-based restoration algorithms to solve the image inpainting and super resolution tasks (see Appendix F for a discussion on our choice of using GANs for the following demonstrations). In all the experiments we use a U-Net architecture (Ronneberger et al., 2015) as our estimator, which we denote by $G(Y, Z)$. For the stochastic algorithms $Z \sim \mathcal{N}(0, I)$ is a random input seed that allows their outputs to vary for each Y, while for the deterministic ones we fix $Z = 0$. We use the same architecture for the stochastic and deterministic algorithms to ensure that it does not impair the evaluation. All the algorithms are enforced to produce perfectly consistent restorations. The optimization task in all of the experiments is a weighted sum of two objectives: $\mathcal{L}_{GAN}$ and $\mathcal{L}_R$. Here, $\mathcal{L}_{GAN}$ is a non-saturating generative adversarial training loss (Goodfellow et al., 2014) combined with R_1 critic regularization (Mescheder et al., 2018), where the critic is the same ResNet architecture as in (Mescheder et al., 2018). Optimizing such a loss would drive $p_{\hat{X}}$ to be as close as possible to p_X, and due to the consistency enforcement, a posterior sampler is the only optimal solution for this task (Theorem 3.1). Note that the critic's objective is to distinguish between samples from $p_{\hat{X}}$ and p_X without taking Y into account. The term $\mathcal{L}_R$ is a loss that drives the restoration algorithm to be robust to input adversarial attacks. It is equal to Equation (7), where Z is zero for the deterministic algorithms. In practice, we perform 5 Adam (Kingma & Ba, 2014) optimization steps to approximate the attack δ^* on each input y, and for the stochastic algorithms we perform the average in Equation (6) over 10 random seeds for each y. The final training objective is

$$\mathcal{L}_{GAN} + \lambda_R \mathcal{L}_R, \tag{8}$$

where λ_R controls the level of the algorithm's robustness. Complementary training details are provided in the following sections and in Appendix E.2.

6.1.1. Experimental setup

Inpainting: We perform several experiments on the image inpainting task, in which some pixels of a high quality image are masked and a restoration algorithm estimates their values. We assume that the locations of the masked pixels are fixed and known. In all of our experiments, we mask the upper $\frac{3}{4}$ part of the image, and leave the remaining bottom pixels untouched. We use the CelebA data set (Liu et al., 2015) for the experiments in this task. Consistency is enforced by replacing the bottom $\frac{1}{4}$ part of the output with that of the input, a typical way to enforce consistency in image inpainting (Saharia et al., 2021; Yu et al., 2018a;b; Yi et al., 2020; Zhao et al., 2021). We train several consistent restoration algorithms by optimizing Equation (8): two deterministic and two stochastic models, where one of each is trained with $\lambda_R = 0$ and the other with $\lambda_R = 500$. We refer to the algorithms trained with $\lambda_R = 0$ as *erratic*, and to the others as *robust*. We choose ϵ in Equation (7) so that $\text{PSNR}(y + \delta^*, y) \geq 32.9\text{dB}$ for any y (so that δ^* leads to a barely noticeable change in y). We refer to this quantity as the *adversarial input PSNR* (AI-PSNR).

Super resolution: We train the same types of algorithms as before (erratic & robust, deterministic & stochastic), but this time we solve the super resolution task. The degradation we consider is a plain averaging with scaling factors of $4\times$, $8\times$, and $16\times$. The training objective and the used architectures remain the same. Consistency is enforced with CEM (Bahat & Michaeli, 2020). Unlike in the inpainting task, this time the dimensionality of the input is different than that of the output. Since our estimator is a neural network with input

and output of the same size, we feed to it an upsampled version of the low-resolution image, using nearest-neighbor interpolation. Moreover, we train the models on CelebA, as well as on the 64×64 version of ImageNet (Chrabaszcz et al., 2017).

Performance metrics: For the inpainting algorithms, we report in Table 1 the perceptual quality performance according to FID (Heusel et al., 2017), precision, and recall (Kynkäänniemi et al., 2019) (with $k = 3$), and present several outputs produced by the algorithms in Figure 1. We also demonstrate the quality and the extent of the output variation of the stochastic models by showing four output samples for each input. The perceptual quality metrics are computed by considering the training set as the real samples, and the algorithms' outputs on the validation set as the fake samples. Moreover, we report the robustness performance of each algorithm according to

$$\mathbb{E}\left[\text{PSNR}(G(Y,Z), G(Y+\delta^*(Y),Z)\right], \tag{9}$$

where the average is computed over the entire validation set and $\delta^*(y)$ is the solution of Equation (6) for each y, and is acquired in the same fashion as in the training stage. Lastly, we report the per-pixel standard deviation of each algorithm over 32 restored samples (for the same input), averaged across all the output pixels and over the entire validation set. This measures the average variation of the output pixels for all the inputs in the validation data, i.e., a higher value corresponds to a more diverse output space, per input, on average. In Figure 4 we report these performance metrics for the super resolution algorithms as well. Please refer to Appendix E.2 for complementary training details.

6.1.2. Analysis of the Results

Inpainting: The erratic deterministic algorithm is competitive with the stochastic methods in terms of FID (Table 1). However, while both robust algorithms attain the same level of robustness (and for the same AI-PSNR), the deterministic one suffers from a significant deterioration in FID and the stochastic one does not. This can be confirmed visually in Figure 1 as well. As the only difference between the robust algorithms is the random noise injection (same model, same loss, etc.), this result supports our hypothesis that stochastic models can overcome the trade-off between robustness and perceptual quality of deterministic mappings.

Another interesting outcome is that the erratic stochastic algorithm produces virtually zero output diversity per input, while the robust stochastic algorithm maintains meaningful output variation. This can be confirmed by the naked eye when observing the randomly sampled outputs in Figure 1 (the robust algorithm creates several types of hair, genders, etc. for the same input), and by the higher per-pixel standard deviation. These results yield several insights: 1) Since the

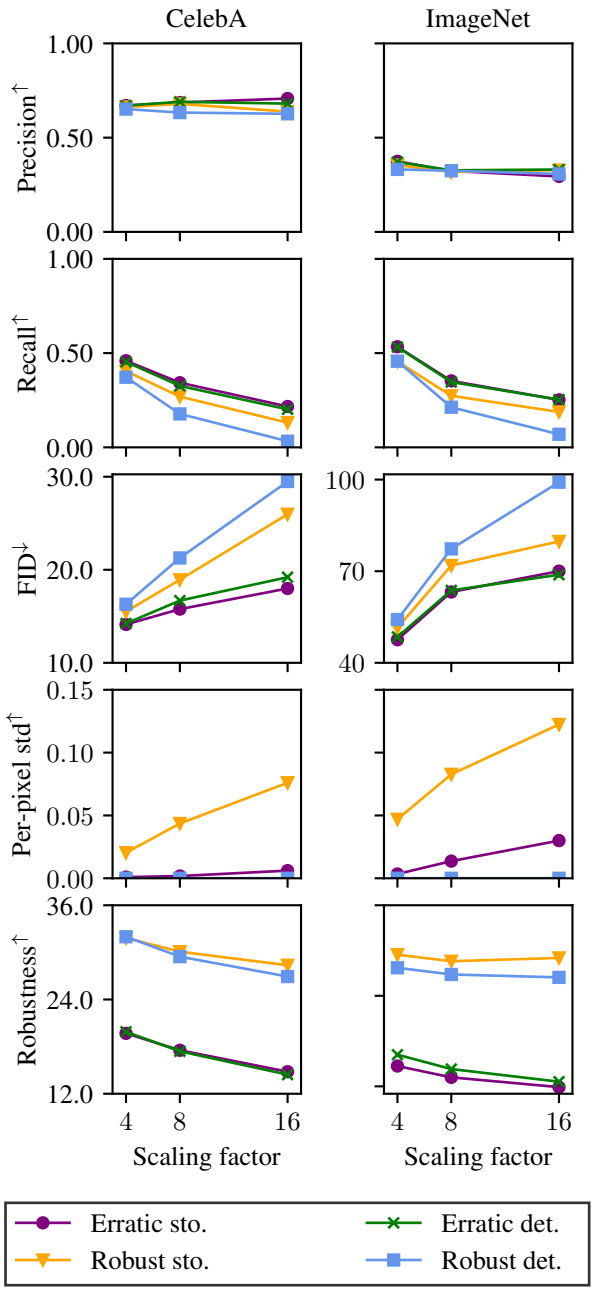


Figure 4. Quantitative evaluation of the super resolution algorithms described in Section 6.1.1. AI-PSNR is equal to 34dB for all algorithms. The anticipated link between robustness and perceptual quality is revealed again: The robust deterministic models achieve the lowest perceptual quality for all scaling factors. As the scaling factor (degradation severity) increases, we observe higher output variability only for the robust stochastic models, and a larger perceptual quality gap for the robust deterministic models.

erratic stochastic algorithm barely produces output variation, it effectively behaves like a deterministic estimator. This shows that the instability of a stochastic model is indicative of conditional mode-collapse (Goodfellow et al., 2014; Isola et al., 2017; Mathieu et al., 2016; Yang et al., 2019). 2) Apparently, in order to attain high perceptual quality, it is less challenging to learn an erratic output curvature rather than to map a random input seed to meaningful, high quality output variation. 3) Enforcing robustness on a stochastic model promotes meaningful output variation and effectively alleviates the conditional mode-collapse issue. These results further support our link between the robustness and stochasticity of a restoration algorithm, as anticipated. Indeed, we observe that the robust deterministic model suffers from a very low recall compared to the robust stochastic one.

Super resolution: The first observation we make from Figure 4 is that, as the scaling factor increases, the FID of all algorithms deteriorate. This is supposedly a result of attempting to solve a more difficult inverse problem: higher scaling factors preserve less information (the problem is "more" ill-posed). Yet, we see that for the same degradation severity, the robust deterministic algorithm always attains the worst FID performance. Moreover, its FID gap with the other algorithms increases with the scaling factor, suggesting that the tradeoff between robustness and perceptual quality (for deterministic estimators) is more severe for a higher scaling factor. Intuitively, a higher scaling factor is expected to increase the size of the support of $p_{X|Y}(\cdot|y)$ for any y, so it covers a larger portion of the support of p_X. We therefore hypothesize that, for high scaling factors, a consistent, robust deterministic estimator would "miss" a larger portion in the support of p_X, and would therefore attain lower recall, just as the results show. Interestingly, we see that the precision of all algorithms and across all scaling factors remain roughly the same, showing that a restoration algorithm can indeed output images that appear natural but still attain low perceptual quality (due to low recall). Lastly, we again see that the erratic stochastic model barely produces output variability, while the robust one does.

6.2. Robustness of SRFlow

Previous work (Choi et al., 2019; 2021) hypothesize that restoration algorithms with high perceptual quality are more vulnerable to adversarial attacks. This hypothesis stems from the low robustness levels of ESRGAN (Wang et al., 2018), a GAN-based, deterministic, high perceptual quality algorithm. However, such a hypothesis is limited as it is based on the evaluation of a single high perceptual quality restoration algorithm. Here we provide additional support for our claim that such an hypothesis is incomplete: it is valid only for deterministic estimators, while stochastic ones can be robust and still attain high perceptual quality.

In Appendix G we evaluate the robustness of SRFlow (Lugmayr et al., 2020), a stochastic super-resolution algorithm that produces highly consistent outputs and comparable perceptual quality to ESRGAN (Wang et al., 2018). As reported in Figure 8, SRFlow exhibits substantially higher robustness than ESRGAN, and comparable robustness to distortion-optimized models. These findings further demonstrate the potential of high perceptual quality stochastic estimators to attain superior robustness compared to deterministic estimators with comparable perceptual quality performance.

7. Summary

In this work we ask whether stochastic estimators are better than deterministic ones, and our short answer is *yes*. Indeed, we proved that a posterior sampler is the only restoration algorithm with perfect consistency and perceptual quality, which shows that a consistent deterministic algorithm can never attain perfect perceptual quality. While a deterministic estimator can still attain high perceptual quality, it must adopt an erratic output behavior in order to do so, which makes it vulnerable to adversarial attacks. Hence, our work provides a novel explanation for the existence of adversarial attacks in high perceptual quality deterministic restoration algorithms. We expand the notion of robustness for stochastic algorithms, and then show that such algorithms can significantly alleviate the tradeoff between robustness and perceptual quality that exists in deterministic algorithms. Interestingly, we find that robustness can be used to promote meaningful output variability in stochastic models, which aligns well with the theory and hypothesis developed in this paper. Our conclusion: Robust, stochastic restoration algorithms do provide better recovery.

8. Acknowledgments

This research was partially supported by the Israel Science Foundation (ISF) under Grants 335/18 and 2318/22 and by the Council For Higher Education - Planning & Budgeting Committee.

References

Antun, V., Renna, F., Poon, C., Adcock, B., and Hansen, A. C. On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, 117(48):30088–30095, 2020.

Bahat, Y. and Michaeli, T. Explorable super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Bevilacqua, M., Roumy, A., Guillemot, C., and line Alberi Morel, M. Low-complexity single-image super-

resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference*, pp. 135.1–135.10. BMVA Press, 2012. ISBN 1-901725-46-4.

Blau, Y. and Michaeli, T. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Chen, R., Qu, Y., Zeng, K., Guo, J., Li, C., and Xie, Y. Persistent memory residual network for single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

Choi, J.-H., Zhang, H., Kim, J.-H., Hsieh, C.-J., and Lee, J.-S. Evaluating robustness of deep image super-resolution against adversarial attacks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Choi, J.-H., Zhang, H., Kim, J.-H., Hsieh, C.-J., and Lee, J.-S. Adversarially robust deep image super-resolution using entropy regularization. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

Choi, J.-H., Zhang, H., Kim, J.-H., Hsieh, C.-J., and Lee, J.-S. Deep image destruction: Vulnerability of deep image-to-image models against adversarial attacks. *arXiv preprint arXiv:2104.15022*, 2021.

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Dong, C., Loy, C. C., He, K., and Tang, X. Image super-resolution using deep convolutional networks, 2015.

Freirich, D., Michaeli, T., and Meir, R. A theory of the distortion-perception tradeoff in wasserstein space. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 25661–25672. Curran Associates, Inc., 2021.

Galteri, L., Seidenari, L., Bertini, M., and Del Bimbo, A. Deep generative adversarial compression artifact removal. *arXiv preprint arXiv:1704.02518*, 2017.

Gandikota, K. V., Chandramouli, P., and Moeller, M. On adversarial robustness of deep image deblurring. In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 3161–3165, 2022.

Gatopoulos, I., Stol, M., and Tomczak, J. M. Super-resolution variational auto-encoders. 2020.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

Jo, Y., Yang, S., and Kim, S. J. Srflow-da: Super-resolution using normalizing flow with deep convolutional block. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 364–372, June 2021.

Kadkhodaie, Z. and Simoncelli, E. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 13242–13254. Curran Associates, Inc., 2021.

Kastryulin, S., Zakirov, D., and Prokopenko, D. PyTorch Image Quality: Metrics and measure for image quality assessment, 2019. Open-source software available at https://github.com/photosynthesis-team/piq.

Kawar, B., Vaksman, G., and Elad, M. Stochastic image denoising by sampling from the posterior distribution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 1866–1875, October 2021a.

Kawar, B., Vaksman, G., and Elad, M. Snips: Solving noisy inverse problems stochastically. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 21757–21769. Curran Associates, Inc., 2021b.

Kawar, B., Elad, M., Ermon, S., and Song, J. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022.

Kim, J., Lee, J. K., and Lee, K. M. Accurate image super-resolution using very deep convolutional networks. *arXiv preprint arXiv:1511.04587*, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*, 2017.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved precision and recall metric for assessing generative models. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Liu, Z.-S., Siu, W.-C., and Chan, Y.-L. Photo-realistic image super-resolution via variational autoencoders. *IEEE Transactions on Circuits and Systems for Video Technology*, 31:1351–1365, 4 2021a. ISSN 1051-8215.

Liu, Z.-S., Siu, W.-C., and Wang, L.-W. Variational autoencoder for reference based image super-resolution. pp. 516–525. IEEE, 6 2021b. ISBN 978-1-6654-4899-4.

Lugmayr, A., Danelljan, M., Van Gool, L., and Timofte, R. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, 2020.

Lugmayr, A., Danelljan, M., and Timofte, R. Ntire 2021 learning the super-resolution space challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 596–612, June 2021.

Lugmayr, A., Danelljan, M., Timofte, R., Kim, K.-w., Kim, Y., Lee, J.-y., Li, Z., Pan, J., Shim, D., Song, K.-U., Tang, J., Wang, C., and Zhao, Z. Ntire 2022 challenge on learning the super-resolution space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 786–797, June 2022.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pp. 416–423 vol.2, 2001.

Mathieu, M., Couprie, C., and LeCun, Y. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2016.

Mescheder, L., Nowozin, S., and Geiger, A. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018.

Ohayon, G., Adrai, T., Vaksman, G., Elad, M., and Milanfar, P. High perceptual quality image denoising with a posterior sampling cgan. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pp. 1805–1813, October 2021.

Raj, A., Bresler, Y., and Li, B. Improving robustness of deep-learning-based image reconstruction. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7932–7942. PMLR, 13–18 Jul 2020.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *arXiv preprint arXiv:1505.04597*, 2015.

Saharia, C., Chan, W., Chang, H., Lee, C. A., Ho, J., Salimans, T., Fleet, D. J., and Norouzi, M. Palette: Image-to-image diffusion models. *arXiv preprint arXiv:2111.05826*, 2021.

Sajjadi, M. S. M., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. Assessing generative models via precision and recall. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *arXiv preprint arXiv:1609.05158*, 2016.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

Yan, H., Zhang, J., Feng, J., Sugiyama, M., and Tan, V. Y. F. Towards adversarially robust deep image denoising. In Raedt, L. D. (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 1516–1522. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

Yang, D. et al. Diversity-sensitive conditional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Yi, Z., Tang, Q., Azizi, S., Jang, D., and Xu, Z. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018a.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018b.

Yu, Y., Yang, W., Tan, Y.-P., and Kot, A. C. Towards robust rain removal against adversarial attacks: A comprehensive benchmark analysis and beyond. *arXiv preprint arXiv:2203.16931*, 2022.

Yue, J., Li, H., Wei, P., Li, G., and Lin, L. Robust real-world image super-resolution against adversarial attacks. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, pp. 5148–5157, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386517.

Zeyde, R., Elad, M., and Protter, M. On single image scale-up using sparse-representations. In *Proceedings of the 7th International Conference on Curves and Surfaces*, pp. 711–730, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 9783642274121.

Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E. I., and Xu, Y. Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021.

A. Proof of Theorem 3.1

Theorem 3.1. *For a deterministic degradation, $y = D(x)$, an estimator $\hat{X}$ is perfectly consistent ($p_{Y|X} = p_{Y|\hat{X}}$) and achieves perfect perceptual quality ($p_{\hat{X}} = p_X$) if and only if it is the posterior sampler $p_{\hat{X}|Y} = p_{X|Y}$.*

Proof. First, note that $D(\hat{X}) = D(X)$ if and only if $p_{Y|\hat{X}} = p_{Y|X}$. Assume that $p_{\hat{X}|Y} = p_{X|Y}$. Then,

$$p_{\hat{X}}(x) = \int_y p_{\hat{X}|Y}(x|y)p_Y(y)dy = \int_y p_{X|Y}(x|y)p_Y(y)dy = p_X(x),$$

and

$$p_{Y|\hat{X}}(y|x) = \frac{p_{\hat{X}|Y}(x|y)p_Y(y)}{p_{\hat{X}}(x)} = \frac{p_{X|Y}(x|y)p_Y(y)}{p_X(x)} = p_{Y|X}(y|x).$$

In the other direction, assume that $p_{\hat{X}} = p_X$ and $p_{Y|\hat{X}} = p_{Y|X}$. Then,

$$p_{\hat{X}|Y}(x|y) = \frac{p_{Y|\hat{X}}(y|x)p_{\hat{X}}(x)}{p_Y(y)} = \frac{p_{Y|X}(y|x)p_X(x)}{p_Y(y)} = p_{X|Y}(x|y),$$

concluding the proof. □

B. Further discussion on deterministic estimators

In this section we expand on the topics discussed in Section 4 and Section 5.

B.1. A tradeoff between precision and recall for continuous, consistent deterministic estimators

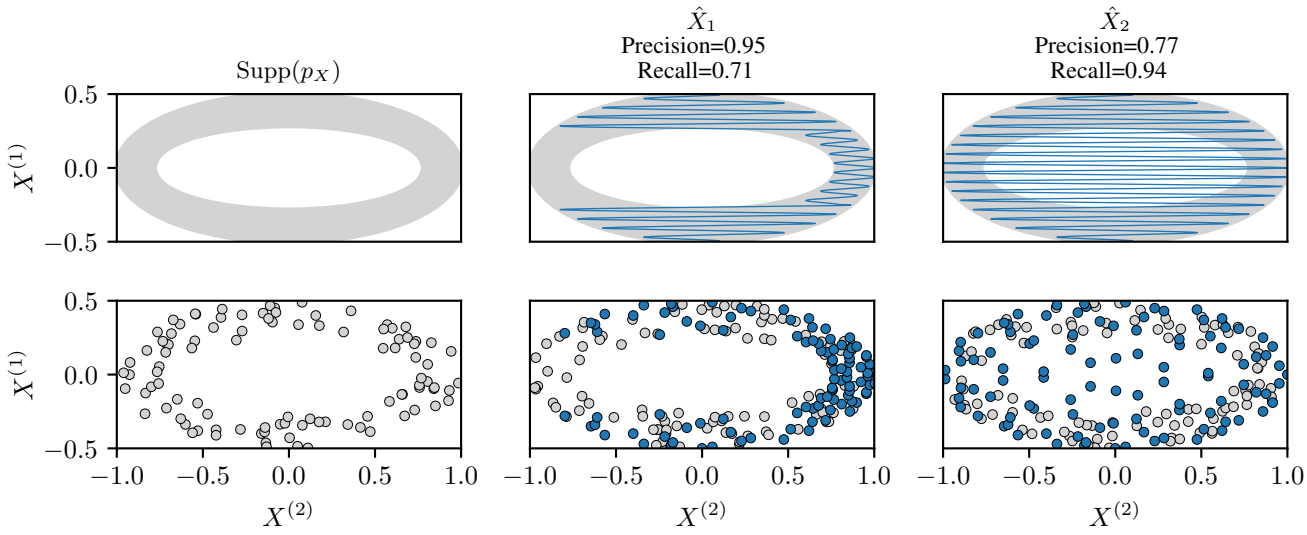


Figure 5. Mental illustration of a tradeoff between precision and recall for consistent, high perceptual quality, continuous deterministic estimators. In the top row we show the support of the data distribution and of the distribution of each estimator's outputs. In the bottom row we show 100 random samples from these distributions. $\hat{X}_1$ avoids the forbidden region (the middle empty ellipse which is not part of Supp(p_X)), and therefore attains almost perfect precision with impaired recall. $\hat{X}_2$ passes through the forbidden region in order to attain higher recall, but as a result compromises on precision since it generates outputs that are not in the data distribution.

Let $X = (X^{(1)}, X^{(2)})$ be a two dimensional random variable supported on the set of all points between two concentric ellipses, as shown in Figure 5. The task is to estimate X given $Y = X^{(1)}$ with perfect consistency. Notice that, different

from the example in Section 4, now there is a "hole" in the support of p_X. In Figure 5 we present two hypothetical estimators that are consistent and continuous (these estimators are handcrafted for the purpose of this demonstration). Observe that due to the hole in p_X, a continuous, deterministic estimator, regardless of how erratic it is, can never approach having perfect perceptual quality (unlike the two examples in Appendix C and Section 4). Such an estimator would always have to pass through the central empty ellipse region, which is not part of $\text{Supp}(p_X)$, in order to produce outputs from both sides of $\text{Supp}(p_X)$. Interestingly, this example reveals a tradeoff between precision and recall for such continuous deterministic estimators. To demonstrate this, let us assume that the data distribution is uniform over its support (the set of points between the two concentric ellipses). We randomly sample 1000 points from each distribution, and compute precision and recall as in Section 4.1. As the results show (Figure 5), $\hat{X}_1$, the estimator that does not pass through the forbidden region, compromises on recall in order to attain higher precision. $\hat{X}_2$ decides to pass through the forbidden region in order to generate samples from the left side of the ellipse, which results in lower precision and higher recall.

Notice that the situation described above has nothing to do with robustness, and occurs even for erratic, non-robust deterministic estimators. Moreover, this tradeoff does not exist for stochastic estimators (Theorem 3.1).

B.2. A note on the MMSE and MAP estimators

In Section 5 we note that the discussion on robust estimators with high perceptual quality is only relevant when assuming that a posterior sampler is robust, i.e., when a small change in y does not lead to an unreasonable change in the distribution $p_{X|Y}(\cdot|y)$. In such a case, we expect that both the MMSE estimator $\mathbb{E}[X|Y]$ and the Maximum A Posteriori (MAP) estimator $\max_x p_{X|Y}(x|Y)$ would also be robust, since the mean and the maximum of $p_{X|Y}(\cdot|y)$ cannot significantly deviate when a small perturbation is added to y.

C. Concrete mathematical example

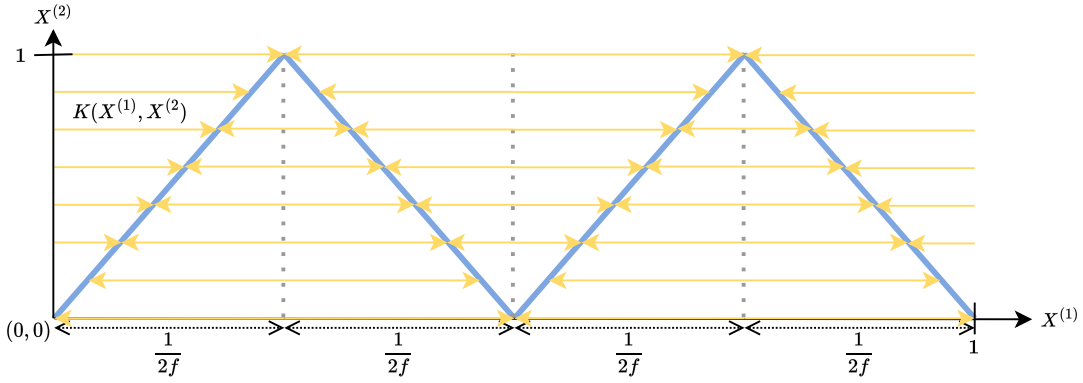


Figure 6. An illustration of $\text{Supp}(p_{\hat{X}})$ (zigzag line) and the mapping $K_f(X^{(1)}, X^{(2)})$ (arrow lines), where $\hat{X} = (X^{(1)}, G_f(X^{(1)}))$ is the estimator from Appendix C and the mapping $K_f(X^{(1)}, X^{(2)})$ maps each point $(X^{(1)}, X^{(2)})$ onto the zigzag in a horizontal fashion, as illustrated in the figure.

We provide a concrete toy example which shows that the perceptual index (according to the W_2 distance) of a consistent deterministic estimator can be arbitrarily minimized by making the estimator more erratic. Let $X = (X^{(1)}, X^{(2)})$ be a two dimensional random variable, where $X^{(1)}, X^{(2)} \sim U(0,1)$, and $X^{(1)}, X^{(2)}$ are statistically independent. Let $Y = X^{(1)}$, and $\hat{X} = (Y, G_f(Y))$ be a consistent, deterministic estimator, where $G_f(y) = \frac{1}{\pi}\arccos\left(\cos\left(2\pi f y\right)\right)$ and $f \in \mathbb{N}$. Our goal is to show that

$$W_2(p_X, p_{\hat{X}}) \underset{f\to\infty}{\longrightarrow} 0. \tag{10}$$

We prove this by considering the Monge formulation of the W_2 distance, i.e.,

$$W_2^2(p_X, p_{\hat{X}}) = \inf_T \mathbb{E}[\|X - T(X)\|_2^2] \; s.t. \; T(X) \sim p_{\hat{X}}, \tag{11}$$

which is true since p_X is absolutely continuous. Let

$$K_f(x_1, x_2) = \frac{1}{2f}\left(\lfloor 2fx_1 \rfloor + \begin{cases} 1 - x_2 & mod(\lfloor 2fx_1 \rfloor, 2) = 1 \\ x_2 & mod(\lfloor 2fx_1 \rfloor, 2) = 0 \end{cases}\right), \tag{12}$$

$$T_0(X) = (K_f(X^{(1)}, X^{(2)}), X^{(2)}). \tag{13}$$

One can show that $T_0(X) \sim p_{\hat{X}}$ (see Figure 6 for intuition), so T_0 satisfies the constraint in Equation (11) and therefore

$$W_2^2(p_X, p_{\hat{X}}) \leq \mathbb{E}[\|X - T_0(X)\|_2^2]. \tag{14}$$

Moreover, $\forall x_1, x_2 \in [0, 1]$ we have

$$\lim_{f\to\infty}(x_1 - \frac{1}{2f}(\lfloor 2fx_1 \rfloor - x_2))^2 = 0, \tag{15}$$

$$\lim_{f\to\infty}(x_1 - \frac{1}{2f}(\lfloor 2fx_1 \rfloor - 1 + x_2))^2 = 0, \tag{16}$$

so

$$\lim_{f\to\infty}(x_1 - K_f(x_1, x_2))^2 = 0. \tag{17}$$

Since both $X^{(1)}, K_f(X^{(1)}, X^{(2)}) \sim U[0, 1]$, we have that $\mathbb{P}((X^{(1)} - K_f(X^{(1)}, X^{(2)}))^2 \leq 4) = 1$. Hence, from the dominated convergence theorem (DCT) we have

$$W_2^2(p_X, p_{\hat{X}}) \leq \lim_{f\to\infty}\mathbb{E}[\|X - T_0(X)\|_2^2] = \lim_{f\to\infty}\int_{[0,1]^2}(x_1 - K_f(x_1, x_2))^2 p_X(x_1, x_2)dx_1dx_2 \tag{18}$$

$$= \int_{[0,1]^2}\lim_{f\to\infty}(x_1 - K_f(x_1, x_2))^2 p_X(x_1, x_2)dx_1dx_2 = 0. \tag{19}$$

We have shown that the perceptual index of the estimator $\hat{X}$ can be arbitrarily minimized (by taking larger values of f).

D. Proof that Equation (7) upper bounds Equation (5)

Our goal is to show that

$$R_{\hat{X},\epsilon} \leq \tilde{R}_{\hat{X},\epsilon}. \tag{20}$$

Let $\hat{X} = G(Y, Z)$ for some random variables Y, Z, and let

$$\delta^* = \arg\max_{\delta:\|\delta\|_2\leq\epsilon} W_2^2(p_{\hat{X}|Y}(\cdot|y), p_{\hat{X}|Y}(\cdot|y + \delta)). \tag{21}$$

Then, for any y

$$r_{\hat{X},\epsilon}(y) = W_2^2(p_{\hat{X}|Y}(\cdot|y), p_{\hat{X}|Y}(\cdot|y + \delta^*)). \tag{22}$$

Note that the W_2 distance between $p_{\hat{X}|Y}(\cdot|y)$ and $p_{\hat{X}|Y}(\cdot|y + \delta^*)$ is essentially the distance between the distributions of the random variables $G(y, Z)$ and $G(y + \delta^*, Z)$. Since the MSE between two random variables upper bounds their W_2^2 distance, we have

$$r_{\hat{X},\epsilon}(y) \leq \mathbb{E}[\|G(y, Z) - G(y + \delta^*, Z)\|_2^2]. \tag{23}$$

By the definition of $\tilde{r}_{\hat{X},\epsilon}$ we have

$$\mathbb{E}[\|G(y, Z) - G(y + \delta^*, Z)\|_2^2] \leq \tilde{r}_{\hat{X},\epsilon}(y), \tag{24}$$

so $r_{\hat{X},\epsilon}(y) \leq \tilde{r}_{\hat{X},\epsilon}(y)$, and since this is true for any y, we conclude that

$$R_{\hat{X},\epsilon} = \mathbb{E}\left[r_{\hat{X},\epsilon}(Y)\right] \leq \mathbb{E}\left[\tilde{r}_{\hat{X},\epsilon}(Y)\right] = \tilde{R}_{\hat{X},\epsilon}. \tag{25}$$

E. Complementary training details

E.1. Toy GAN experiments

The sizes of the training and validation sets are 100,000 and 10,000, respectively, both of which are random, independent samples from the toy distribution described in Section 4. In all experiments the estimator's and critic's architecture is the simple fully connected network described in Table 2. Notice that the estimator outputs only one value (the estimate of $X^{(2)}$) as the value of $X^{(1)}$ is known. The output and the input are being concatenated to result in an output of size 2, which is the final estimate. For the stochastic estimators $\hat{X}_{\text{StoGAN}}$ and $\hat{X}_{\text{ErraticStoGAN}}$ one of the inputs is Y and the other is $Z \sim U[0,1]$. For the deterministic estimators we simply fix $Z = 0$, like we do in the experiments on natural images. The loss we use is the exact same one we use for natural images, but with a gradient penalty coefficient of 10.0, and $\lambda_R = 0.1$ for the robust estimators (both the deterministic and the stochastic). During training, we alternate between one training step for the estimator and one step for the critic, where each is trained for a total of $20,000$ steps. To optimize the parameters of the networks we use the Adam optimizer with $(\beta_1, \beta_2) = (0, 0.9)$ and a learning rate of 10^{-4}. ϵ in the computation of Equation (7) is set to 10^{-3} in all experiments. For the stochastic estimator, we perform the average in Equation (6) over 50 realizations of Z. Finding the adversarial attack on each input in done by using the Adam optimizer for 4 steps, with $(\beta_1, \beta_2) = (0.9, 0.999)$ and a learning rate of 10^{-4}. In Table 3 we report the robustness values of all toy GAN estimators.

Table 2. The architecture of both the estimator and critic in all toy GAN experiments.

Layer	Output size	Filter
Fully Connected	512	$2 \to 512$
ReLU	512	-
Fully Connected	512	$512 \to 512$
ReLU	512	-
Fully Connected	512	$512 \to 512$
ReLU	512	-
Fully Connected	512	$512 \to 512$
ReLU	512	-
Fully Connected	512	$512 \to 1$

Table 3. Robustness of the toy GAN estimators from Section 4.2 and Section 5.

Estimator	Robustness: $\sqrt{\tilde{R}_{\hat{X},\epsilon}}(\cdot 10^{-3})$ (higher is better)	$\epsilon(\cdot 10^{-3})$
$\hat{X}_{\text{ErraticGAN}}$	3.62	1
$\hat{X}_{\text{RobustGAN}}$	0.77	1
$\hat{X}_{\text{StoErraticGAN}}$	0.13	1
$\hat{X}_{\text{StoRobustGAN}}$	0.51	1

E.2. Experiments on natural images

E.2.1. Neural network architectures

In all the experiments the estimator is a U-Net architecture (Ronneberger et al., 2015) (receiving Y and Z as inputs), and the critic is a ResNet model (He et al., 2015) with a similar structure to the one described in (Mescheder et al., 2018). Full details of the architectures are disclosed in Figure 7 and Table 4. While the chosen GAN architecture for the deterministic and stochastic estimators could have been further improved, we chose the structure described herein, as it provides good quality outcomes and is relatively easy to train. We believe that, while improving the GAN training and architecture may boost performance, it would not change the interplay we exposed between deterministic and stochastic restoration methods.

E.2.2. Optimization settings

We alternate between optimizing the estimator and the critic (one step for each model at a time), performing a total of 1.2M steps for each model. At the estimator's optimization step we also perform an "inner" optimization procedure that finds

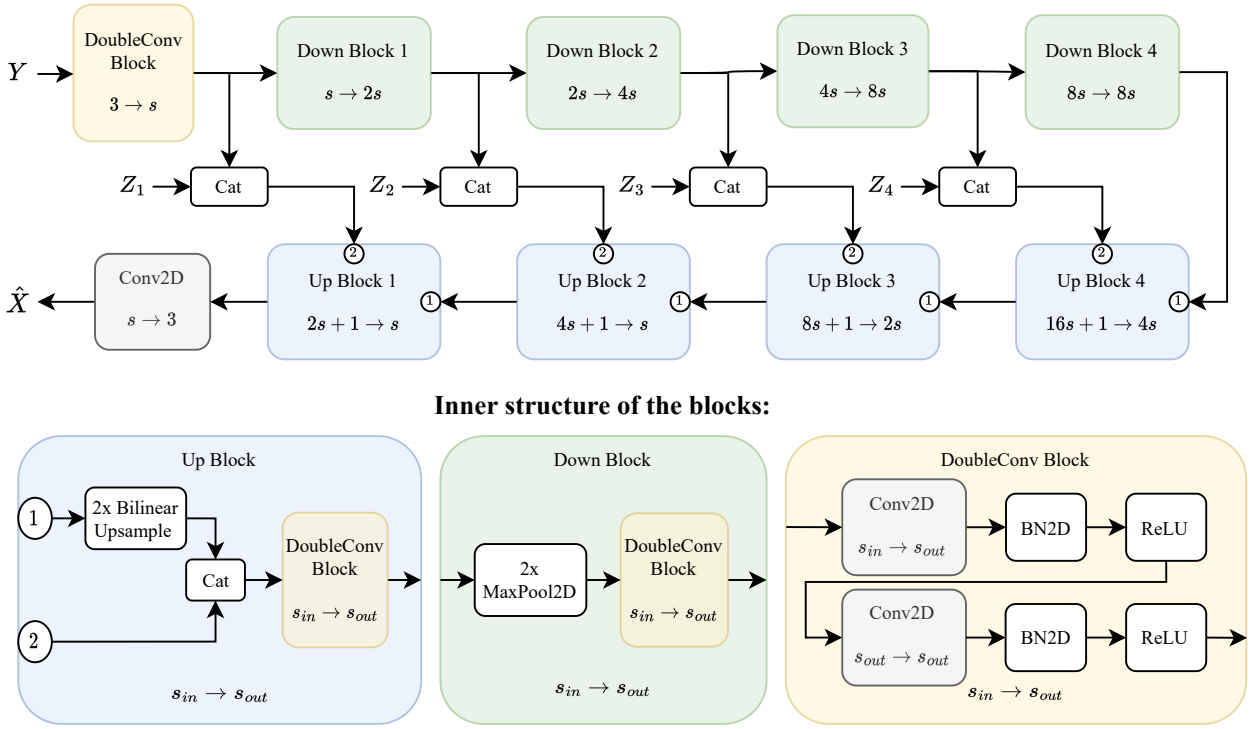


Figure 7. Description of the U-Net architecture (Ronneberger et al., 2015), adopted as the estimator in all the experiments on natural images. We use $Z = (Z_1, Z_2, Z_3, Z_4)$, where for the deterministic estimators $Z_i = 0$, and for the stochastic ones Z_i are statistically independent random vectors, each following a normal distribution with zero mean and identity covariance matrix. Each Z_i is reshaped to match the spatial size of the input to Down Block i. Cat corresponds to concatenation on the channels dimension. BN2D is a two dimensional batch normalization layer (Ioffe & Szegedy, 2015). We use a ReLU activation after each BN2D layer in each DoubleConv block. There is no activation in the last Conv2D layer of the network. The only hyperparameter of the network is s. For the experiments on CelebA we use $s = 24$ (for both the inpainting and super resolution tasks), and for the experiments on ImageNet we use $s = 32$.

Table 4. Full description of the critic's architecture used in all the experiments on natural images. We use pre-activation ResNet blocks and Leaky ReLU activations with a slope of 0.2 everywhere (except for the last layer, where there is no activation). The output of each ResNet block is multiplied by 0.1. The only hyperparameter of the network is f, and the spatial extent of the input image is always 64×64 in our experiments. $f = 16$ in all the CelebA experiments, and $f = 32$ in all ImageNet experiments. This architecture follows a similar structure to the one described in (Mescheder et al., 2018).

Layer	Output size	Filter
Conv2D	$f \times 64 \times 64$	$3 \to f$
ResNet Block	$f \times 64 \times 64$	$f \to f \to f$
ResNet Block	$2f \times 64 \times 64$	$f \to f \to 2f$
Average Pooling 2D	$2f \times 32 \times 32$	-
ResNet Block	$2f \times 32 \times 32$	$2f \to 2f \to 2f$
ResNet Block	$4f \times 32 \times 32$	$2f \to 2f \to 4f$
Average Pooling 2D	$4f \times 16 \times 16$	-
ResNet Block	$4f \times 16 \times 16$	$4f \to 4f \to 4f$
ResNet Block	$8f \times 16 \times 16$	$4f \to 4f \to 8f$
Average Pooling 2D	$8f \times 8 \times 8$	-
ResNet Block	$8f \times 8 \times 8$	$8f \to 8f \to 8f$
ResNet Block	$16f \times 8 \times 8$	$8f \to 8f \to 16f$
Average Pooling 2D	$16f \times 4 \times 4$	-
ResNet Block	$16f \times 4 \times 4$	$16f \to 16f \to 16f$
ResNet Block	$16f \times 4 \times 4$	$16f \to 16f \to 16f$
Average Pooling 2D	$16f \times 2 \times 2$	-
ResNet Block	$16f \times 2 \times 2$	$16f \to 16f \to 16f$
ResNet Block	$16f \times 2 \times 2$	$16f \to 16f \to 16f$
Fully Connected	1	$16f \cdot 2 \cdot 2 \to 1$

the adversarial attack of each input, as described in Appendix E.2.3. The robustness loss (Equation (7)) is included in the estimator's objective once in every three optimization steps, i.e., we always perform two estimator training steps solely with a GAN loss, and the robustness loss is added at the third step when training a robust model. For the optimization of the estimator' and critic's parameters, we always use the Adam optimizer with $(\beta_1, \beta_2) = (0.5, 0.99)$ and a learning rate of 10^{-4}. For both models we also perform a multi-step learning rate scheduling with a decay of $\gamma = 0.6$, scheduled at the steps $400k, 500k, 600k, 700k$ and $750k$. The R_1 gradient penalty coefficient of the critic is set to 1.

E.2.3. Computation of the robustness loss (Equation (7))

Note that measuring the robustness of an estimator with Equation (7) involves a maximization procedure for each y (Equation (6)). We approximate this maximization with 5 optimization steps using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 1 and $(\beta_1, \beta_2) = (0.9, 0.999)$. The optimizer's parameters are re-initialized at the beginning of each training step (so its inner parameters are not transferred from step to step). The same procedure with the same hyper-parameters is done both for training and validation.

Another thing to note is that computing Equation (6) for each y requires averaging over samples of Z. For the stochastic estimators we sample 10 instances of Z to compute this average, and for the deterministic estimators averaging is not required since $Z = 0$.

Lastly, in the inpainting task we use $\epsilon = 2.5$, and in the super resolution task we use $\epsilon = 0.553, 0.276, 0.138$ for the scaling factors $\times 4, \times 8, \times 16$, respectively. These values of ϵ ensure the same bound on the adversarial input PSNR across all scaling factors. To ensure that the attack δ holds $\|\delta\|_2 \leq \epsilon$, we project δ on the ϵ ball after each update step of the attack.

E.2.4. DATA SETS

CelebA: We pre-process each source image by first resizing it to 102x102 via bilinear-interpolation and then cropping the center 64x64 pixels, so the source images we use are of size 64x64. We use 80% of the original CelebA training data to train all the algorithms (the images in the resulting subset are being chosen at random), and the remaining 20% is added to the original CelebA validation data. We do so to have enough images for perceptual quality evaluation using metrics such as FID (Heusel et al., 2017).

ImageNet: We use the 64×64 version of the ImageNet data set (Chrabaszcz et al., 2017) with no pre-processing, and with the original training and validation sets splits.

E.2.5. COMPUTATION OF PERCEPTUAL QUALITY METRICS

We compute FID (Heusel et al., 2017), precision and recall (Kynkäänniemi et al., 2019) (with $k = 3$) using the PyTorch package (Kastryulin et al., 2019), and with the default feature extraction network of that package (InceptionV3 (Szegedy et al., 2015)). We randomly choose 50,000 real and fake samples to perform the evaluation (the real samples are the high quality training images, and the fake samples are the outputs of the algorithm on the validation inputs). For the stochastic algorithms each fake sample corresponds to a different input, i.e., we sample 50,000 random seeds and each seed is combined with a different input to provide one output, effectively sampling one instance from $p_{\hat{X}|Y}(\cdot|y)$ for each y.

F. What is the rationale of utilizing GANs to compare stochastic and deterministic restoration models?

Among image restoration methods, GAN-based (Wang et al., 2018; Ledig et al., 2017; Bahat & Michaeli, 2020; Ohayon et al., 2021), flow-based (Lugmayr et al., 2020; Jo et al., 2021), diffusion-based (Song & Ermon, 2019; Kadkhodaie & Simoncelli, 2021; Kawar et al., 2021a;b; 2022), and VAE-based (Gatopoulos et al., 2020; Liu et al., 2021a;b) techniques are currently the most widely adopted and effective in terms of achieving high levels of perceptual quality. However, in order to provide sufficient evidence for our hypothesis in this paper, which posits that stochastic models can achieve high levels of perceptual quality with higher levels of robustness compared to deterministic models, it is essential to conduct a fair and unbiased comparison. For example, a stochastic flow-based method that achieves high levels of perceptual quality may exhibit greater robustness compared to a deterministic GAN-based technique with similar levels of perceptual quality (as demonstrated in Appendix G). However, this does not necessarily imply that the superior performance of the flow-based method is solely attributed to its stochasticity. Rather, it shows that a restoration algorithm with high levels of perceptual quality can also exhibit robustness, which refutes the hypothesis proposed in (Choi et al., 2019). Hence, to conduct a fair evaluation, it is crucial to isolate the impact of stochasticity on the model's performance. This can be achieved by using GANs, as they naturally allow to perform a comparison "on the same grounds" (same data sets, same loss, same architecture, same AI-PSNR, same hyper-parameters, etc.) by manipulating the input random seed from a degenerate distribution (to obtain a deterministic model) to a non-degenerate distribution (to obtain a stochastic model). Such an effective way to isolate the impact of output stochasticiy on the model's performance is currently much less trivial with flow-based, diffusion-based, and VAE-based methods. We therefore leave the analysis of these methods for future work.

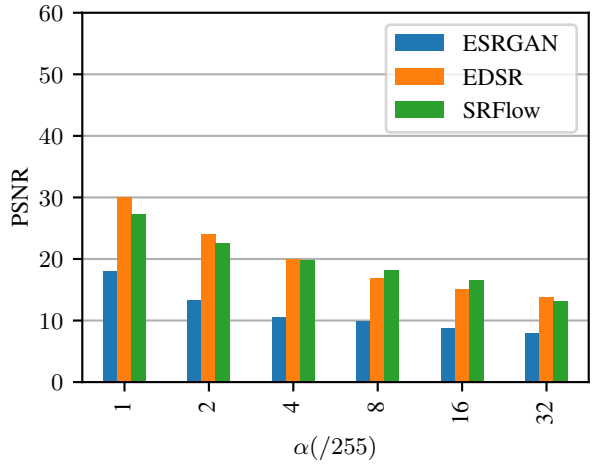
G. Robustness of SRFlow: results

We assess the robustness of SRFlow (Lugmayr et al., 2020) by adapting the I-FGSM (Choi et al., 2019; Kurakin et al., 2017) method to be applicable to any differentiable estimation procedure that utilizes a random seed as input. This adaptation enables a fair comparison of stochastic methods with the I-FGSM attacks performed on other methods (Choi et al., 2019).

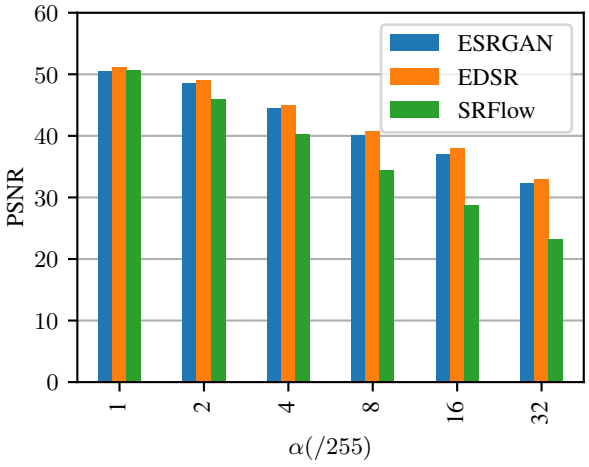
To find the input attack $\tilde{y}$ on the SRFlow model (denoted by $\text{SRF}(Y, Z)$), we compute the loss $\mathbb{E}[\|\text{SRF}(y, Z) - \text{SRF}(\tilde{y}, Z)\|_2]$ for each $Y = y$, averaging over 10 realizations of Z (the random seed of SRFlow), and then continue to perform 50 I-FGSM update steps in the same fashion as described in (Choi et al., 2019), Sec. 3.1. To ensure a fair comparison with other models, we adhere to the official public implementation of the I-FGSM procedure described in (Choi et al., 2019), and use the same initialization for the attack. Additionally, we utilize the official code and pre-trained model checkpoints of SRFlow as provided by the authors. Lastly, as in (Choi et al., 2019), we perform the evaluation on the Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2010), and BSD100 (Martin et al., 2001) data sets.

The robustness evaluation of SRFlow is presented in Figure 8. For ease of comparison, we also include the robustness

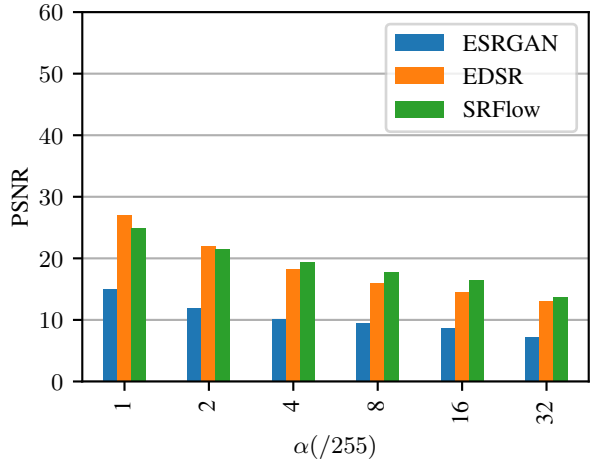
levels of ESRGAN (Wang et al., 2018) and EDSR (Chen et al., 2018) as reported in (Choi et al., 2019) (refer to (Choi et al., 2019) for the analysis of additional distortion-based models). Our results show that the robustness levels of SRFlow are significantly higher than those of ESRGAN across all values of α, the hyper-parameter controlling the severity of the I-FGSM attack. This confirms that even an "off the shelf" stochastic restoration algorithm with high perceptual quality can exhibit much higher robustness than a deterministic algorithm with comparable output perceptual quality. Furthermore, the results show that SRFlow exhibits comparable level of robustness to EDSR and most of the distortion-based models evaluated in (Choi et al., 2019). I.e., despite achieving high perceptual quality, SRFlow maintains a robustness level that is comparable to models that attain much lower levels of perceptual quality. All and all, this experiment further supports our claim that the link between high perceptual quality and low robustness (Choi et al., 2019) is valid only for deterministic mappings.



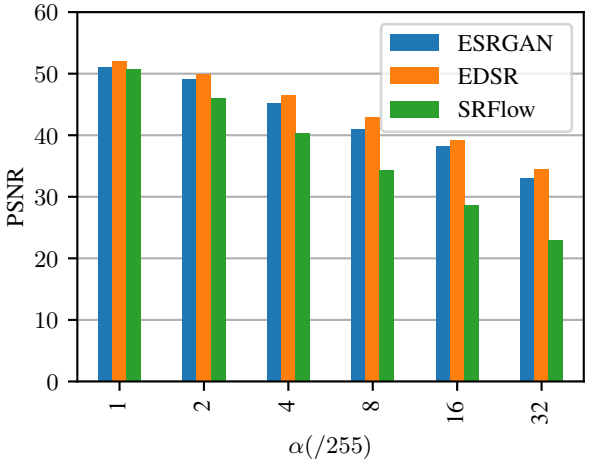
(a) Set5 adversarial output PSNR.



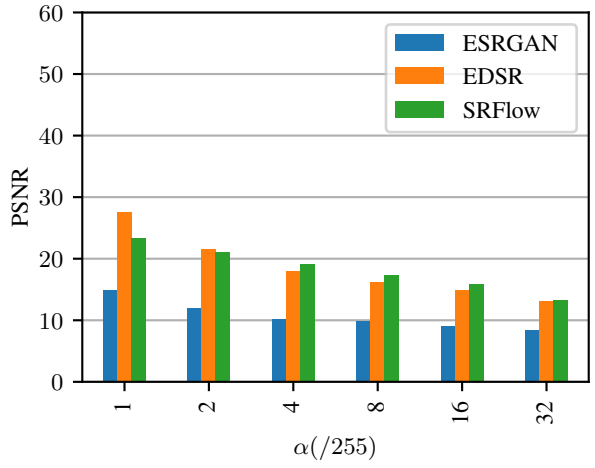
(b) Set5 adversarial input PSNR.



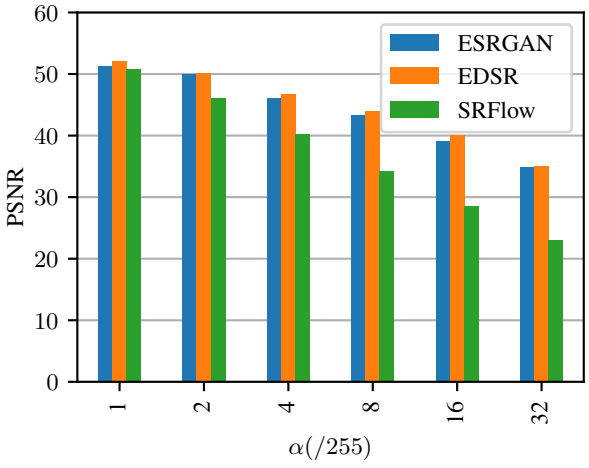
(c) Set14 adversarial output PSNR.



(d) Set14 adversarial input PSNR.



(e) BSD100 adversarial output PSNR.



(f) BSD100 adversarial input PSNR.

Figure 8. Comparison of adversarial output PSNR (robustness) and adversarial input PSNR (AI-PSNR) on Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2010), and BSD100 (Martin et al., 2001), using the I-FGSM attack (Choi et al., 2019; Kurakin et al., 2017) with several values of α. The evaluated methods are SRFlow (Lugmayr et al., 2020), EDSR (Chen et al., 2018), and ESRGAN (Wang et al., 2018). The results of EDSR and ESRGAN are copied from (Choi et al., 2019), whereas SRFlow is evaluated as described in Appendix G.