Noise Robust Generative Adversarial Networks

Takuihiro Kaneko\textsuperscript{1} Tatsuya Harada\textsuperscript{1,2}

\textsuperscript{1}The University of Tokyo \textsuperscript{2}RIKEN

Abstract

Generative adversarial networks (GANs) are neural networks that learn data distributions through adversarial training. In intensive studies, recent GANs have shown promising results for reproducing training images. However, even when training images are noisy (a)(d), they attempt to reproduce the training images faithfully, as shown in (b)(e). To remedy this, we propose noise robust GANs (NR-GANs), which can learn to generate clean images (c)(f), even when training images are noisy (a)(d). Our NR-GANs are unique in that they solve this problem without full knowledge of the noise (e.g., the noise distribution type, noise amount, or signal-noise relationship). Indeed, in (c) and (f), although the same models (in particular, SI-NR-GAN-II, which is a variant of NR-GANs) are used for different noises (a)(d), they succeed in learning clean image generators adaptively through training.

1. Introduction

In computer vision and machine learning, generative models have been actively studied and used to generate or reproduce an image that is indistinguishable from a real image. Generative adversarial networks (GANs) [21], which learn data distributions through adversarial training, have garnered special attention owing to their ability to produce high-quality images. In particular, with recent advancements [2, 54, 46, 23, 40, 58, 36, 76], the latest GANs (e.g., BigGAN [6] and StyleGAN [37]) have succeeded in generating images indistinguishable for humans.

However, a persistent issue is that recent high-capacity GANs could replicate images faithfully even though the training images were noisy. Indeed, as shown in Figure 1(b)(e), when standard GAN is trained with noisy images, it attempts to recreate them. Although the long-term development of devices has steadily improved image quality, image degradation is unavoidable in real situations. For example, electronic noise is inevitable in digital imaging [60, 1] and estimator variance often appears as noise in graphic rendering [83, 72]. Therefore, susceptibility to noise is practically undesirable for GANs.

The question becomes: “How can we learn a clean image generator even when only noisy images are available for training?” We call this problem noise robust image generation. One solution is to apply a denoiser as pre-
process. However, a limitation is that the generator performance highly relies on the quality of the denoiser, which is relatively difficult to learn when clean images are not available for training. As an alternative, AmbientGAN [5] was recently proposed, which provides a promising solution by simulating the noise corruption on the generated images and learning the discriminator so that it distinguishes a real noisy image from a simulative corrupted generated image. This makes it possible to learn a clean image generator directly from noisy images without relying on a denoiser.

However, a key limitation of AmbientGAN is that it assumes that the noise corruption process is pre-defined. Therefore, to utilize it, we need to have all information about the noise, such as the noise distribution type (e.g., Gaussian), noise amount (e.g., standard deviation), and signal-noise relationship. For instance, to treat 16 noises shown in Figure 2, we need to carefully prepare 16 noise simulation models that depend on the noise.

To deal with this, we propose noise robust GANs (NR-GANs), which can achieve noise robust image generation without having complete noise information. Our main idea is as follows. We first introduce two generators, a clean image generator and noise generator. To make them generate an image and noise, respectively, we impose a distribution or transformation constraint on the noise generator so that it only captures the components that follow the specified distribution or transformation invariance. As such a constraint can take various forms depending on the type of assumptions; we develop five variants: two signal-independent NR-GANs (SI-NR-GANs) and three signal-dependent NR-GANs (SD-NR-GANs). Figure 1(c)/(f) shows examples of images generated using NR-GANs. Here, although the same models are used for different noises (a)/(d), NR-GANs succeed in learning clean image generators adaptively.

As the noise robustness of GANs has not been sufficiently studied, we first perform a comprehensive study on CIFAR-10 [42], where we compare various models in diverse noise settings (in which we test 152 conditions). Furthermore, inspired by the recent large-scale study on GANs [44], we also examin the performance on more complex datasets (LSUN BEDROOM [74] and FFHQ [37]). Finally, we demonstrate the applicability of NR-GANs in image denoising, where we learn a denoiser using generated noisy images and generated clean images (GN2GC), and empirically examine a chicken and egg problem between noise robust image generation and image denoising.

Overall, our contributions are summarized as follows:

- We provide noise robust image generation, the purpose of which is to learn a clean image generator even when training images are noisy. In particular, we solve this problem without full knowledge of the noise.
- To achieve this, we propose a novel family of GANs called NR-GANs that train a clean image generator and noise generator simultaneously with a distribution or transformation constrain on the noise generator.
- We provide a comprehensive study on CIFAR-10 (in which we test 152 conditions) and examine the versatility in more complex datasets (LSUN BEDROOM and FFHQ); finally, we demonstrate the applicability in image denoising. The project page is available at https://takuhirok.github.io/NR-GAN/.

2. Related work

Deep generative models. Image generation is a fundamental problem and has been intensively studied in computer vision and machine learning. Recently, deep generative models have emerged as a promising framework. Among them, three prominent models along with GANs are variational...
autoencoders [39, 64], autoregressive models [70], and flow-based models [14, 15]. Each model has pros and cons. A well-known disadvantage of GANs is training instability; however, it has been steadily improved by recent advancements [2, 54, 46, 4, 65, 23, 40, 58, 36, 76, 6, 9, 37]. In this work, we focus on GANs for their design flexibility, which allows them to incorporate the core of our models, a noise generator and its constraints. Also in other models, image fidelity has improved [71, 61, 56, 38]. Hence, sensitivity to noise can be problematic. Incorporating our ideas into them is a possible direction of future work.

**Image denoising.** Image denoising is also a fundamental problem and several methods have been proposed. They are roughly categorized into two: model-based methods [12, 22, 17, 51, 7, 50, 16, 49, 52] and discriminative learning methods [29, 55, 67, 77, 78, 8, 24, 45, 43, 3]. Recently, discriminative learning methods have shown a better performance; however, a limitation is that most of them (i.e., Noise2Clean (N2C)) require clean images for supervised training of a denoiser. To handle this, self-supervised learning methods (e.g., Noise2Void (N2V) [43] and Noise2Self (N2S) [3]) were proposed. These methods assume the same data setting as ours, i.e., only noisy images are available for training. However, they still have some limitations, e.g., they cannot handle pixel-wise correlated noise, such as shown in Figure 2(G)(H), and their performance is still inferior to supervised learning methods.

Image denoising and our noise robust image generation is a chicken and egg problem and each task can be used as a pre-task for learning the other. In the spirit of AmbientGAN, we aim to learn a clean image generator directly from noisy images. However, examining the performance on (1) learning a generator using denoised images and (2) learning a denoiser using generated clean and noisy images is an interesting research topic. Motivated by this, we empirically examined them through comparative studies. We provide the results in Sections 8.1 and 8.3.

**Noise robust models.** Except for image denoising, noise robust models have been studied in image classification to learn a classifier in practical settings. There are two studies addressing label noise [18, 80, 62, 53, 30, 68, 63, 25, 66, 31, 59, 20] and addressing image noise [82, 13]. For both tasks, the issue is the memorization effect [75], i.e., DNN classifiers can fit labels or images even though they are noisy or fully corrupted. As demonstrated in Figure 1, a similar issue also occurs in image generation.

Pertaining image generation, handling of label noise [35, 34, 69, 32] and image noise [5] has begun to be studied. Our NR-GANs are categorized into the latter. As discussed in Section 1, AmbientGAN [5] is a representative model in the latter category. However, a limitation is that it requires full knowledge of the noise. Therefore, we introduce NR-GANs to solve this problem as they do not have this limitation.

### 3. Notation and problem statement

We first define notation and the problem statement. Hereafter, we use superscripts r and g to denote the real distribution and generative distribution, respectively. Let \( y \) be the observable noisy image and \( x \) and \( n \) be the underlying signal (i.e., clean image) and noise, respectively, where \( y, x, n \in \mathbb{R}^{H \times W \times C} \) (\( H, W, C \) are the height, width, and channels of an image, respectively). In particular, we assume that \( y \) can be decomposed additively: \( y = x + n \).

Our task is to learn a clean image generator that can reproduce clean images, such that \( p^g(x) = p^r(x) \), when trained with noisy images \( y^r \sim p^r(y) \). This is a challenge for standard GAN as it attempts to mimic the observable images including the noise; namely, it learns \( p^g(y) = p^r(y) \).

We assume various types of noise. Figure 2 shows the categorization and examples of the noises that we address in this paper. They include signal-independent noises (A)–(H), signal-dependent noises (I)–(P), pixel-wise correlated noises (G)(H), local noises (C)(D), and their combination (H)(K)(L)(O)(P). We also consider two cases: the noise amount is either fixed or variable across the dataset.

As discussed in Section 1, one solution is AmbientGAN [5]; however, it is limited by the need for prior noise knowledge. We plan a solution that will not require that full prior knowledge. Our central idea is to introduce two generators, i.e., a clean image generator and noise generator, and impose a distribution or transformation constraint on the noise generator so that it captures only the noise-specific components. In particular, we explicate such constraints by relying on the signal-noise dependency. We first review our baseline AmbientGAN [5] (Section 4); then detail NR-GANs for signal-independent noise (Section 5) and signal-dependent noise (Section 6).

### 4. Baseline: AmbientGAN

AmbientGAN [5] (Figures 3(a) and 4(a)) is a variant of GANs, which learns an underlying distribution \( p^r(x) \) only from noisy images \( y^r \sim p^r(y) \). This is a challenging because the desired images \( x^r \sim p^r(x) \) are not observable during training. To overcome this challenge, AmbientGAN introduces a noise simulation model \( y = F^\theta(x) \) under the assumption that it is priory known. The main idea of AmbientGAN is to incorporate this noise simulation model into the adversarial training framework:

\[
\min_{G_x} \max_{D_y} \mathbb{E}_{y^r \sim p^r(y)} [\log D_y(y^r)] + \mathbb{E}_{x^r \sim p^r(x)} [\log(1 - D_y(F^\theta(G_x(z_x))))].
\]

\(^1\)We decompose additively; however, note that this representation includes signal-independent noise \( n \sim p(n) \) and signal-dependent noise \( n \sim p(n|\mathbf{x}) \).

\(^2\)Strictly, AmbientGAN can handle more general *lossy* data, such as missing data. Here, we narrow the target in accordance with our task.
5. Signal-independent noise robust GANs

As described above, a limitation of AmbientGAN is that it requires that a noise simulation model \( F_\theta(x) \) is priorly known. To alleviate this, we introduce a noise generator \( n = G_n(z_n) \) (Figure 3(b)) and train it along with a clean image generator \( G_x \) using the following objective function:

\[
\min_{G_x,G_n} \max_{D_y} \mathbb{E}_{y^r \sim p(y^r)} \left[ \log D_y(y^r) \right]
+ \mathbb{E}_{x \sim p(z_\alpha), z_n \sim p(z_n)} \left[ \log (1 - D_y(G_x(z,x) + G_n(z_n))) \right].
\]

(2)

Nevertheless, without any constraints, there is no incentive to make \( G_x \) and \( G_n \) generate an image and a noise, respectively. Therefore, we provide a constraint on \( G_n \) so that it captures only the noise-specific components. In particular, we develop two variants that have different assumptions: SI-NR-GAN-I (Section 5.1) and SI-NR-GAN-II (Section 5.2).

5.1. SI-NR-GAN-I

In SI-NR-GAN-I, we assume the following:

**Assumption 1** (i) The noise \( n \) is conditionally pixel-wise independent given the signal \( x \). (ii) The noise distribution type (e.g., Gaussian) is priorly known. Note that the noise amount needs not to be known. (iii) The signal \( x \) does not follow the defined noise distribution.

Under this assumption, we develop SI-NR-GAN-I (Figure 3(c)). In this model, we regularize the output distribution of \( G_n \) in a pixel-wise manner using a reparameterization trick [39]. Here, we present the case when the noise distribution type is defined as zero-mean Gaussian:

\[
y = x + n, \text{ where } n \sim \mathcal{N}(0, \text{diag}(\sigma)^2),
\]

(3)

where \( \sigma \in \mathbb{R}^{H \times W \times C} \) is the pixel-wise standard deviation. In this case, we redefine the noise generator as \( \sigma = G_n(z_n) \); and introduce an auxiliary pixel-wise random variable \( \epsilon \sim \mathcal{N}(0, I) \), where \( \epsilon \in \mathbb{R}^{H \times W \times C} \); and then calculate the noise \( n \) by multiplying them: \( n = \sigma \cdot \epsilon \), where \( \cdot \) represents an element-wise product. This formulation allows the noise to be sampled as \( n \sim \mathcal{N}(0, \text{diag}(\sigma)^2) \).

In SI-NR-GAN-I, \( \sigma \) is learned through training in a pixel-wise manner. Therefore, the same model can be applied to various noises (e.g., Figure 2(A)–(D)), in which each pixel’s noise follows a Gaussian distribution, while the noise amount is different in a sample-wise (e.g., (B)) or pixel-wise (e.g., (D)) manner.

5.2. SI-NR-GAN-II

Two limitations of SI-NR-GAN-I are that it assumes that (i) the noise is pixel-wise independent and (ii) the noise distribution type is pre-defined. The first assumption makes it difficult to apply to a pixel-wise correlated noise (e.g., Figure 2(G)(H)). The second assumption could cause difficulty when diverse noises are mixed (e.g., Figure 2(F)) or the noise distribution type is different from the pre-defined (e.g., Figure 2(E)). This motivates us to devise SI-NR-GAN-II, which works under a different assumption:

**Assumption 2** (i) The noise \( n \) is rotation-, channel-shuffle-, or color-inverse-invariant. (ii) The signal \( x \) is rotation-, channel-shuffle-, or color-inverse-invariant.

3Strictly, our approach is applicable as long as a noise follows a differentiable distribution [39].
Among the noises in Figure 2, this assumption holds in all signal-independent noises (A)–(H). This assumption is reasonable when \( \mathbf{n} \) is a zero-mean signal-independent noise and \( \mathbf{x} \) is a natural image. Under this assumption, we establish SI-NR-GAN-II (Figure 3(d)). In this model, we redefine the noise generator as \( \hat{\mathbf{n}} = G_{\mathbf{n}}(\mathbf{z}_n) (\hat{\mathbf{n}} \in \mathbb{R}^{H \times W \times C}) \) and apply transformations to \( \hat{\mathbf{n}} \) by \( \mathbf{n} = T(\hat{\mathbf{n}}) \), where \( T \) is a transformation function. As \( T \), we can use arbitrary transformation as long as it is applicable to \( \mathbf{n} \) but not allowable to \( \mathbf{x} \). In practice, we use three transformations: (i) rotation – rotating \( \hat{\mathbf{n}} \) by \( d \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} \) randomly, (ii) channel shuffle – shuffling RGB channels randomly, and (iii) color inversion – inverting colors randomly in a channel-wise manner. Each one utilizes one of the invariant and variant characteristics mentioned in Assumption 2. In SI-NR-GAN-II, the noise origin \( \hat{\mathbf{n}} \) is acquired in a data-driven manner; therefore, it is applicable to diverse noises (e.g., Figure 2(A)–(H)) without model modifications.

### 6. Signal-dependent noise robust GANs

Just like in the signal-independent noise case, AmbientGAN is applicable to signal-dependent noise by incorporating the pre-defined noise model (Figure 4(a)). However, it requires prior knowledge about the noise distribution type, signal-noise relationship, and noise amount. To deal with these requirements, we establish three variants that have different assumptions: SD-NR-GAN-I (Section 6.1), SD-NR-GAN-II (Section 6.2), and SD-NR-GAN-III (Section 6.3).

#### 6.1. SD-NR-GAN-I

We first consider the case when the following assumption holds in addition to Assumption 1.

**Assumption 3** The signal-noise relationship is priorly known. Note that the noise amount needs not be known.

We only depict the generators. (a) AmbientGAN pre-defines the noise model. (b) SD-NR-GAN-I represents the signal-noise relationship explicitly, while the noise amount is estimated through training. (c) SD-NR-GAN-II expresses the signal-noise relationship implicitly and this relationship and the noise amount are acquired through training. (d) SD-NR-GAN-III only imposes a weak constraint via the transformation and learns the noise distribution type, signal-noise relationship, and noise amount through training.

Under this assumption, we devise SD-NR-GAN-I (Figure 4(b)), which incorporates a signal-noise relational procedure into SI-NR-GAN-I explicitly. In particular, we devise two configurations for two typical signal-dependent noises: multiplicative Gaussian noise (Figure 2(I)(J)) and Poisson noise (Figure 2(M)(N)).

Multiplicative Gaussian noise is defined as

\[
y = \mathbf{x} + \mathbf{n}, \quad \text{where} \quad \mathbf{n} \sim \mathcal{N}(0, \text{diag}(\mathbf{\sigma} \cdot \mathbf{x})^2).
\]

To represent this noise with trainable \( \mathbf{\sigma} \), we redesign the noise generator as \( \mathbf{\sigma} = G_{\mathbf{\sigma}}(\mathbf{z}_n) \). Then, we convert \( \mathbf{\sigma} \) using a signal-noise relational function \( R(\mathbf{x}, \mathbf{\sigma}) = \mathbf{\sigma} \cdot \mathbf{x} = \hat{\mathbf{\sigma}} \).

Finally, we obtain \( \mathbf{n} \sim \mathcal{N}(0, \text{diag}(\hat{\mathbf{\sigma}})^2) \) by using the reparameterization trick described in Section 5.1.

Poisson noise (or shot noise) is sampled by \( \mathbf{y} \sim \text{Poisson}(|\lambda \cdot \mathbf{x}|/\lambda) \), where \( \lambda \) is the total number of events. As this noise is discrete and intractable to construction of a differentiable model, we use a Gaussian approximation [26], which is commonly used for Poisson noise modeling:

\[
y = \mathbf{x} + \mathbf{n}, \quad \text{where} \quad \mathbf{n} \sim \mathcal{N}(0, \text{diag}(\sqrt{\mathbf{\sigma} \cdot \mathbf{x}})^2),
\]

where \( \mathbf{\sigma} = \sqrt{1/\lambda} \). The implementation method is almost the same as that for the multiplicative Gaussian noise except that we redefine \( R(\mathbf{x}, \mathbf{\sigma}) \) as \( R(\mathbf{x}, \mathbf{\sigma}) = \mathbf{\sigma} \cdot \sqrt{\mathbf{x}} = \hat{\mathbf{\sigma}} \).

In both noise cases, the noise amount \( \mathbf{\sigma} \) is trainable; therefore, each configuration of SD-NR-GAN-I is applicable to the noises in Figure 2(I)(J) and those in Figure 2(M)(N), respectively, without model modifications.

#### 6.2. SD-NR-GAN-II

In SD-NR-GAN-II, we consider the case when the noise distribution type is known (i.e., Assumption 1 holds) but the signal-noise relationship is unknown (i.e., Assumption 3 is not required). Under this assumption, we aim to learn \( R(\mathbf{x}, \mathbf{\sigma}) \) implicitly, which is explicitly given in SD-NR-GAN-I. To achieve this, we develop SD-NR-GAN-II (Figure 4(c)), which is an extension of SI-NR-GAN-I incorporating the image latent vector \( \mathbf{z}_n \) into the input of \( G_{\mathbf{n}} \), i.e.,

![Figure 4. Comparison of AmbientGAN (baseline) and SD-NR-GANs (proposed). Because the discriminators are the same, we only depict the generators. (a) AmbientGAN pre-defines the noise model. (b) SD-NR-GAN-I represents the signal-noise relationship explicitly, while the noise amount is estimated through training. (c) SD-NR-GAN-II expresses the signal-noise relationship implicitly and this relationship and the noise amount are acquired through training. (d) SD-NR-GAN-III only imposes a weak constraint via the transformation and learns the noise distribution type, signal-noise relationship, and noise amount through training.](image-url)
\( \sigma = G_n(z_n, z_x) \). Similarly to SI-NR-GAN-I, we sample \( n \sim N(0, \text{diag}(\sigma) I) \) using the reparameterization trick described in Section 5.1. Here, we consider the case when the noise distribution type is defined as zero-mean Gaussian.

As discussed in Section 6.1, multiplicative Gaussian noise and Poisson noise are represented (or approximated) as signal-dependent Gaussian noise; therefore, SD-NR-GAN-II is applicable to these noises (e.g., Figure 2(I)(J)(M)(N)). Furthermore, SD-NR-GAN-II can internally learn \( R(x, \sigma) \); therefore, the same model can also be applied to signal-independent noise (Figure 2(A)–(D)), i.e., \( R(x, \sigma) = x \), and the combination of multiple noises (Figure 2(K)(L)(O)(P)), e.g., \( R(x, \sigma_d, \sigma_t) = \sigma_d \cdot x + \sigma_t \).

6.3. SD-NR-GAN-III

Finally, we deal with the case when both the noise distribution type and signal-noise relationship are not known. In this case, we impose a similar assumption as Assumption 2. However, rotation and channel shuffle collapse the per-pixel signal-noise dependency that is included in typical signal-dependent noise (e.g., Figure 2(I)–(P)). Therefore, we only induce the assumption regarding color inversion. Under this assumption, we devise SD-NR-GAN-III (Figure 4(d)). Similarly to SD-NR-GAN-II, SD-NR-GAN-III learns the signal-noise relationship implicitly by incorporating \( z_x \) into the input of \( G_n \), i.e., \( \hat{n} = G_n(z_n, z_x) \). Similarly to SI-NR-GAN-II, we impose a transformation constraint on \( G_n \) by applying \( n = T(\hat{n}) \), where \( T \) is defined as color inversion. The noise origin \( \hat{n} \) is learned through training; therefore, SD-NR-GAN-III can be adopted to various noises (e.g., all noises in Figure 2) without modifying the model.

7. Advanced techniques for practice

7.1. Alleviation of convergence speed difference

In proposed NR-GANs, \( G_\sigma \) and \( G_n \) are learned simultaneously. Ideally, we expect that \( G_\sigma \) and \( G_n \) would be optimized at the same speed; however, through experiments, we found that \( G_n \) tends to learn faster than \( G_\sigma \) and results in a mode collapse in the early training phase. A possible cause is that the noise distribution is simpler and easier to learn than the image distribution. To address this problem, we apply the diversity-sensitive regularization [73] to \( G_n \). Intuitively, this regularization makes \( G_n \) sensitive to \( z_n \) and has an effect to prevent the mode collapse. In the experiments, we incorporate this technique to all NR-GANs. We discuss the effect of this regularization in our arXiv [33].

7.2. Alleviation of approximation degradation

As described in Section 6.1, we apply a Gaussian approximation to the Poisson noise to make it tractable and differentiable. However, through experiments, we found that this approximation causes the performance degradation even using AmbientGAN, which knows all information about the noise. A possible reason is that powerful \( G_\sigma \) attempts to fill in the discretized gap caused by this approximation. To alleviate the effect, we apply an anti-alias (or low-pass) filter [79] to \( x \) before providing to \( D_y \). In particular, we found that applying vertical and horizontal blur filters respectively and providing both to \( D_y \) works well. In the experiments, we apply this technique to all GANs in the Poisson or Poisson-Gaussian noise setting. We discuss the effect with and without this technique in our arXiv [33].

8. Experiments

8.1. Comprehensive study

To advance the research on noise robust image generation, we first conducted a comprehensive study, where we compared various models in diverse noise settings (in which we tested 152 conditions in total).

Data setting. In this comprehensive study, we used CIFAR-10 [42], which contains 60k 32 \( \times \) 32 natural images, partitioned into 50k training and 10k test images. We selected this dataset because it is commonly used to examine the benchmark performance of generative models (also in the study of AmbientGAN [5]); additionally, the image size is reasonable for a large-scale comparative study. Note that we also conducted experiments using more complex datasets in Section 8.2. With regard to noise, we tested 16 noises, shown in Figure 2. See the caption for their details.

Compared models. In addition to the models in Figures 3 and 4, we tested several baselines. As comparative GAN models, we examined four models: (1) Standard GAN, (2) P-AmbientGAN (parametric AmbientGAN), a straightforward extension of AmbientGAN, which has a single trainable parameter \( \sigma \). As with SI-NR-GAN-I and SD-NR-GAN-I, we construct this model for Gaussian, multiplicative Gaussian, and Poisson noises and generate the noise with \( \sigma \) using a reparameterization trick [39]. (3) SI-NR-GAN-0 (Figure 3(b)), which has the same generators as SI-NR-GANs but has no constraint on \( G_n \). (4) SD-NR-GAN-0, which has the same generators as SD-NR-GAN-II and -III but has no constraint on \( G_n \).

We also examined the performance of learning GANs using denoised images (denoiser+GANs). As a denoiser, we investigated four methods. As typical model-based methods, we used (1) GAT-BM3D [52] and (2) CBM3D [11] for Poisson/Poisson-Gaussian noise (Figure 2(M)–(P)) and the other noises, respectively. As discriminative learning

---

5For further clarification, we conducted a comparative study on transformations in SD-NR-GAN-III. See our arXiv [33] for details.

6Strictly speaking, this strategy goes against the assumptions of SD-NR-GAN-II and -III because they are agnostic to the signal-noise relationship. However, in this main text, we do that to focus on comparison of the generator performance.
methods, we used (3) N2V (Noise2Void) [43] and (4) N2N (Noise2Noise) [45]. N2V can be used in the same data setting as ours (i.e., only noisy images are available for training), while N2N requires noisy image pairs for training. We used N2N because it is commonly used as the upper bound of self-supervised learning methods (e.g., N2V).

**Evaluation metrics.** We used the Fréchet inception distance (FID) [28] as an evaluation metric because its validity has been demonstrated in large-scale studies on GANs [47, 44], and because the sensitivity to the noise has also been shown [28]. The FID measures the distance between real and generative distributions and a smaller value is better.

**Implementation.** We implemented GANs using the ResNet architectures [27] and trained them using a non-saturating GAN loss [21] with a real gradient penalty regularization [57]. In NR-GANs, we used similar architectures for $G_a$ and $G_n$. As our aim is to construct a general model applicable to various noises, we examined the performance when the training settings are fixed regardless of the noise. We provide the implementation details in our arXiv [33].

**Results on signal-independent noises.** The upper part of Table 1 summarizes the results on signal-independent noises. In P-AmbientGAN and SI-NR-GANs, we defined the distribution type as Gaussian for all noise settings and analyzed the effect when the noise is beyond assumption. Our main findings are the following:

1. **Comparison among GAN models.** As expected, AmbientGAN tends to achieve the best score owing to the advantageous training setting, while the best SI-NR-GAN shows a competitive performance (with a difference of 3.3 in the worst case). P-AmbientGAN is defeated by SI-NR-GAN-I in all cases. These results indicate that our two-generator model is reasonable when training a noise generator and image generator simultaneously.

2. **Comparison between SI-NR-GANs and denoiser+GANs.** The best SI-NR-GAN outperforms the best denoiser+GAN in most cases (except for (G)). In particular, pixel-wise correlated noises (G)(H) are intractable for denoiser+GANs except for N2N+GAN, which uses additional supervision, while SI-NR-GAN-II works well and outperforms the baseline models by a large margin (with a difference of over 100).

3. **Comparison among SI-NR-GANs.** SI-NR-GAN-II shows the stable performance across all cases (the difference to the best SI-NR-GAN is within 3.1). SI-NR-GAN-I shows the best or competitive performance in Gaussian (A)–(D) or near Gaussian noise (F); however, the performance degrades when the distribution is beyond assumption (E)(G)(H).

**Results on signal-dependent noises.** The lower part of Table 1 lists the results on signal-dependent noises. In P-AmbientGAN and SD-NR-GAN-I, we defined the distribution type as multiplicative Gaussian and Poisson in (I)–(L) and (M)–(P), respectively. With regard to a comparison among GAN models and comparison between SD-NR-GANs and denoiser+GANs, similar findings (i.e., the best SD-NRGAN is comparable with AmbientGAN and outperforms the best denoiser+GAN) are observed; therefore, herein we discuss a comparison among SD-NR-GANs. SD-NR-GAN-II and -III stability work better than SD-NR-GAN-0. Among the two, SD-NR-GAN-II, which has a stronger assumption, outperforms SD-NR-GAN-III in all cases (with a difference of over 5.4). SD-NR-GAN-I shows the best or competitive performance when noises are within or a little over assumption (I)–(K)(M)–(O); however, when the unexpected noise increases (L)(P), the performance degrades.

**Summary.** Through the comprehensive study, we confirm the following: (1) NR-GANs work reasonably well comparing to other GAN models and denoiser+GANs. (2) Weakly constrained NR-GANs stability work well across various settings, while (3) strongly constrained NR-GANs show a better performance when noise is within assumption.⁷

---

⁷We provide further analyses and examples of generated images in our arXiv [33].
1. FFHQ contains 3 million bedroom images, randomly split into training and test sets in the ratio of 99 to 1. Referring to this study, we used the same training and test sets used in Section 8.2. Evaluation on complex datasets. We provide other case results in our arXiv [33].

8.2. Evaluation on complex datasets

Inspired by the resent large-scale study on GANs [44], we also examined the performance on more complex datasets. Referring to this study, we used the 128 × 128 versions of LSUN BEDROOM [74] and FFHQ [37]. LSUN BEDROOM contains about 3 million bedroom images, randomly split into training and test sets in the ratio of 99 to 1. FFHQ contains 70k face images, partitioned into 60k training and 10k test images. As these datasets are calculation-demanding, we selected six noises for LSUN BEDROOM and two noises for FFHQ. We provide the implementation details in our arXiv [33].

Table 2 list the results. Just like the CIFAR-10 results, we found that the best NR-GAN outperforms standard GAN and its performance is closer to that of AmbientGAN. In contrast, differently from the CIFAR-10 results, we found that in complex datasets, some weakly constrained SD-NR-GANs suffer from learning difficulty (e.g., SD-NR-GAN-III in LSUN BEDROOM (M)). This is undesirable but understandable because in complex datasets it is highly challenging to isolate noise from the dependent signal without an explicit knowledge about their dependency. This is related to GAN training dynamics and addressing this limitation is our future work. As reference, we provide qualitative results in our arXiv [33].

8.3. Application to image denoising

NR-GANs can generate an image and noise, respectively. By utilizing this, we create clean and noisy image pairs synthetically and use them for learning a denoiser. We call this method {GeneratedNoise2GeneratedClean

\[ \text{GN2GC} \]. In particular, we employed the generators that achieve the best FID in Table 2 (denoted by bold font). Note that NR-GANs are trained only using noisy images; therefore, GN2GC can be used in the same data setting as self-supervised learning methods (N2V [43] and N2S [3]). We used the same training and test sets used in Section 8.2. We present the implementation details in our arXiv [33].

We summarize the results in Table 3. We found that GN2GC not only outperforms the state-of-the-art self-supervised learning methods (N2V and N2S) but also is comparable with N2N, which learns in advantageous conditions. The requirement for pre-training GANs could narrow the applications of GN2GC; however, we believe that its potential for image denoising would increase along with rapid progress of GANs. We show examples of denoised images in our arXiv [33].

9. Conclusion

To achieve noise robust image generation without full knowledge of the noise, we developed a new family of GANs called NR-GANs which learn a noise generator with a clean image generator, while imposing a distribution or transformation constraint on the noise generator. In particular, we introduced five variants: two SI-NR-GANs and three SD-NR-GANs, which have different assumptions. We examined the effectiveness and limitations of NR-GANs on three benchmark datasets and demonstrated the applicability in image denoising. In the future, we hope that our findings facilitate the construction of a generative model in a real-world scenario where only noisy images are available.

Acknowledgement. We thank Naoya Fushishita, Takayuki Hara, and Atsuhiro Noguchi for helpful discussions. This work was partially supported by JST CREST Grant Number JPMJCR1403, and partially supported by JSPS KAKENHI Grant Number JP19H01115.

---

8Strictly speaking, the previous study [44] used CelebA-HQ [36] instead of FFHQ. The reason why we used FFHQ is that FFHQ is the latest and more challenging dataset that includes vastly more variation.

9We provide other case results in our arXiv [33].
References


[44] Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in GANs. In ICML, 2019.
[53] Eran Malach and Shai Shalev-Shwartz. Decoupling “when to update” from “how to update”. In NIPS, 2017.


