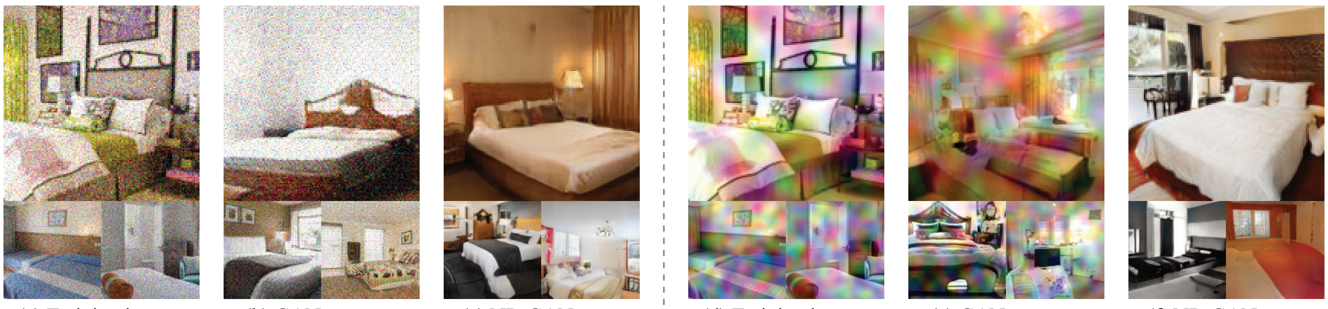
Noise Robust Generative Adversarial Networks

Takuhiro Kaneko[1] Tatsuya Harada[1,2]

[1]The University of Tokyo [2]RIKEN



(a) Training images (Gaussian noise) (b) GAN (baseline) (Trained with (a)) (c) NR-GAN (ours) (Trained with (a)) (d) Training images (Brown Gaussian noise) (e) GAN (baseline) (Trained with (d)) (f) NR-GAN (ours) (Trained with (d))

Figure 1. **Examples of noise robust image generation.** 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.

Abstract

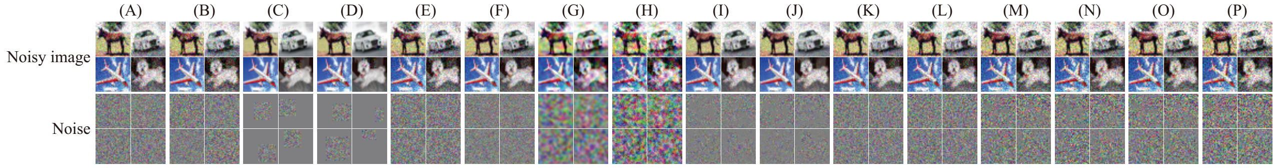
Generative adversarial networks (GANs) are neural networks that learn data distributions through adversarial training. In intensive studies, recent GANs have shown promising results for reproducing training images. However, in spite of noise, they reproduce images with fidelity. As an alternative, we propose a novel family of GANs called noise robust GANs (NR-GANs), which can learn a clean image generator even when training images are noisy. In particular, NR-GANs can solve this problem without having complete noise information (e.g., the noise distribution type, noise amount, or signal-noise relationship). To achieve this, we introduce a noise generator and train it along with a clean image generator. However, without any constraints, there is no incentive to generate an image and noise separately. Therefore, we propose distribution and transformation constraints that encourage the noise generator to capture only the noise-specific components. In particular, considering such constraints under different assumptions, we devise two variants of NR-GANs for signal-independent noise and three variants of NR-GANs for signal-dependent noise. On three benchmark datasets, we demonstrate the effectiveness of NR-GANs in noise robust image generation. Furthermore, we show the applicability of NR-GANs in image denoising. Our code is available at `https://github.com/takuhirok/NR-GAN/`.

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, 57, 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-



	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)	(J)	(K)	(L)	(M)	(N)	(O)	(P)
Abbreviation	**AGF**	**AGV**	**LGF**	**LGV**	**U**	**MIX**	**BG**	**A+G**	**MGF**	**MGV**	**A+I**	**A+I**	**PF**	**PV**	**A+M**	**A+M**
Noise distribution type	Additive Gaussian	Additive Gaussian	Local Gaussian	Local Gaussian	Uniform	Mixture	Brown Gaussian	(A) + (G)	Multiplicative Gaussian	Multiplicative Gaussian	(A) + (I)	(A) + (I)	Poisson	Poisson	(A) + (M)	(A) + (M)
Noise amount	Fixed	Variable	Fixed	Variable	Fixed	Fixed	Fixed	Fixed	Fixed	Variable	Few (A)	Much (A)	Fixed	Variable	Few (A)	Much (A)
Signal-dependent									✓	✓	✓	✓	✓	✓	✓	✓
Pixel-wise correlated							✓	✓								
Local			✓	✓												

Figure 2. **Noise categorization and examples.** In this paper, we handle various types of noise. (A)(B) Additive Gaussian noise with fixed $\sigma = 25$ (A) and variable $\sigma \in [5, 50]$ (B), where σ is the standard deviation of a Gaussian distribution. (C)(D) Local Gaussian noise with fixed patch $p_h = p_w = 16$ (C) and variable patch $p_h, p_w \in [8, 24]$ (D), where p_h and p_w are the height and width of noise region, respectively. σ is fixed to 25 in both cases. (E) Uniform noise $[-50, 50]$. (F) Mixture noise [81]. Mixture of 10% uniform noise $[-50, 50]$, 20% Gaussian noise ($\sigma = 25$), and 70% Gaussian noise ($\sigma = 15$). (G) Brown Gaussian noise [45]. A Gaussian filter (kernel size is 5×5) is applied to (A). (H) Additive Brown Gaussian noise. (G) is added to (A) while remaining the original (A). (I)(J) Multiplicative Gaussian noise with fixed $\sigma = 25$ (I) and variable $\sigma \in [5, 50]$ (J). (K)(L) Additive and multiplicative Gaussian noise. (K) Sum of few (A) ($\sigma = 5$) and (I). (L) Sum of much (A) ($\sigma = 25$) and (I). (M)(N) Poisson noise with fixed $\lambda = 30$ (M) and variable $\lambda \in [10, 50]$ (N), where λ is the total number of events. (O)(P) Poisson-Gaussian noise. (O) Sum of few (A) ($\sigma = 5$) and (M). (P) Sum of much (A) ($\sigma = 25$) and (M).

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 simulatively 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, 57, 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 $\boldsymbol{y}$ be the observable noisy image and $\boldsymbol{x}$ and $\boldsymbol{n}$ be the underlying signal (i.e., clean image) and noise, respectively, where $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{n} \in \mathbb{R}^{H \times W \times C}$ (H, W, and C are the height, width, and channels of an image, respectively). In particular, we assume that $\boldsymbol{y}$ can be decomposed additively: $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$.[1] Our task is to learn a clean image generator that can reproduce clean images, such that $p^g(\boldsymbol{x}) = p^r(\boldsymbol{x})$, when trained with noisy images $\boldsymbol{y}^r \sim p^r(\boldsymbol{y})$. This is a challenge for standard GAN as it attempts to mimic the observable images including the noise; namely, it learns $p^g(\boldsymbol{y}) = p^r(\boldsymbol{y})$.

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 is 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).

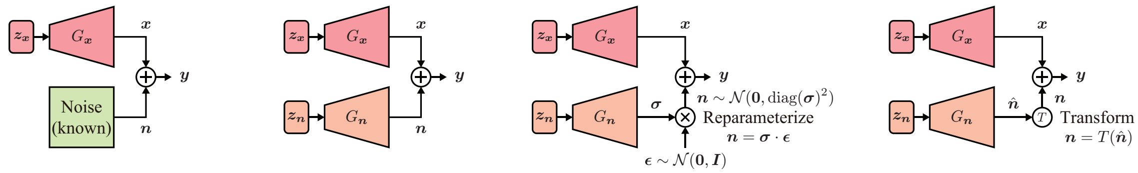
4. Baseline: AmbientGAN

AmbientGAN [5] (Figures 3(a) and 4(a)) is a variant of GANs, which learns an underlying distribution $p^r(\boldsymbol{x})$ only from noisy images $\boldsymbol{y}^r \sim p^r(\boldsymbol{y})$.[2] This is a challenging because the desired images $\boldsymbol{x}^r \sim p^r(\boldsymbol{x})$ are not observable during training. To overcome this challenge, AmbientGAN introduces a noise simulation model $\boldsymbol{y} = F_{\boldsymbol{\theta}}(\boldsymbol{x})$ under the assumption that it is priorly known. The main idea of AmbientGAN is to incorporate this noise simulation model into the adversarial training framework:

$$\begin{aligned}\min_{G_{\boldsymbol{x}}} \max_{D_{\boldsymbol{y}}} \; & \mathbb{E}_{\boldsymbol{y}^r \sim p^r(\boldsymbol{y})}[\log D_{\boldsymbol{y}}(\boldsymbol{y}^r)] \\ & + \mathbb{E}_{\boldsymbol{z}_{\boldsymbol{x}} \sim p(\boldsymbol{z}_{\boldsymbol{x}}), \boldsymbol{\theta} \sim p(\boldsymbol{\theta})}[\log(1 - D_{\boldsymbol{y}}(F_{\boldsymbol{\theta}}(G_{\boldsymbol{x}}(\boldsymbol{z}_{\boldsymbol{x}}))))]. \quad (1)\end{aligned}$$

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

[2] Strictly, AmbientGAN can handle more general *lossy* data, such as missing data. Here, we narrow the target in accordance with our task.



(a) AmbientGAN (baseline) (b) Two-generator model (c) SI-NR-GAN-I (proposed) (d) SI-NR-GAN-II (proposed)

	(a) AmbientGAN (baseline)	(b) Two-generator model	(c) SI-NR-GAN-I (proposed)	(d) SI-NR-GAN-II (proposed)
Noise distribution type	Known	–	Known	Unknown
Noise amount	Known	–	Unknown	Unknown
Applicable noise	Specific modeling	–	(A)–(D)	(A)–(H)

Figure 3. **Comparison of AmbientGAN (baseline) and SI-NR-GANs (proposed).** Because the discriminators are the same, we only depict the generators. (a) AmbientGAN assumes that the noise model is pre-defined. (b) To mitigate this requirement, we introduce a two-generator model and learn a noise generator $G_{\boldsymbol{n}}$ along with a clean image generator $G_{\boldsymbol{x}}$. (c) To make $G_{\boldsymbol{n}}$ capture only the noise specific components, SI-NR-GAN-I regularizes the output distribution of $G_{\boldsymbol{n}}$ using a reparameterization trick under the assumption that the noise distribution type is known. (d) Furthermore, considering the situation when the noise distribution type is unknown, we develop SI-NR-GAN-II, which applies transformations $\boldsymbol{n} = T(\hat{\boldsymbol{n}})$ to extract the transformation-invariant element, i.e., noise.

Just like standard GAN, a generator $G_{\boldsymbol{x}}$ transforms the latent vector $\boldsymbol{z_x}$ into an image $\boldsymbol{x}^g = G_{\boldsymbol{x}}(\boldsymbol{z_x})$. However, differently from the standard GAN discriminator, which directly distinguishes a real image $\boldsymbol{y}^r$ from a generated image $\boldsymbol{x}^g$, the AmbientGAN discriminator $D_{\boldsymbol{y}}$ distinguishes $\boldsymbol{y}^r$ from a *noised* generated image $\boldsymbol{y}^g = F_{\boldsymbol{\theta}}(\boldsymbol{x}^g)$. Intuitively, this modification allows noisy $p^g(\boldsymbol{y})$ $(= p^g(F_{\boldsymbol{\theta}}(\boldsymbol{x})))$ to get close to noisy $p^r(\boldsymbol{y})$ $(= p^r(F_{\boldsymbol{\theta}}(\boldsymbol{x})))$. When $F_{\boldsymbol{\theta}}$ is invertible or uniquely determined, underlying clean $p^g(\boldsymbol{x})$ also approaches underlying clean $p^r(\boldsymbol{x})$.

5. Signal-independent noise robust GANs

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

$$\min_{G_{\boldsymbol{x}}, G_{\boldsymbol{n}}} \max_{D_{\boldsymbol{y}}} \mathbb{E}_{\boldsymbol{y}^r \sim p^r(\boldsymbol{y})}[\log D_{\boldsymbol{y}}(\boldsymbol{y}^r)] + \mathbb{E}_{\boldsymbol{z_x} \sim p(\boldsymbol{z_x}), \boldsymbol{z_n} \sim p(\boldsymbol{z_n})}[\log(1 - D_{\boldsymbol{y}}(G_{\boldsymbol{x}}(\boldsymbol{z_x}) + G_{\boldsymbol{n}}(\boldsymbol{z_n})))]. \tag{2}$$

Nevertheless, without any constraints, there is no incentive to make $G_{\boldsymbol{x}}$ and $G_{\boldsymbol{n}}$ generate an image and a noise, respectively. Therefore, we provide a constraint on $G_{\boldsymbol{n}}$ 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 $\boldsymbol{n}$ is conditionally pixel-wise independent given the signal $\boldsymbol{x}$. (ii) The noise distribution type (e.g., Gaussian) is priorly known. Note that the noise amount needs not to be known. (iii) The signal $\boldsymbol{x}$ 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 $G_{\boldsymbol{n}}$ 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:[3]

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}, \text{where } \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \mathrm{diag}(\boldsymbol{\sigma})^2), \tag{3}$$

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

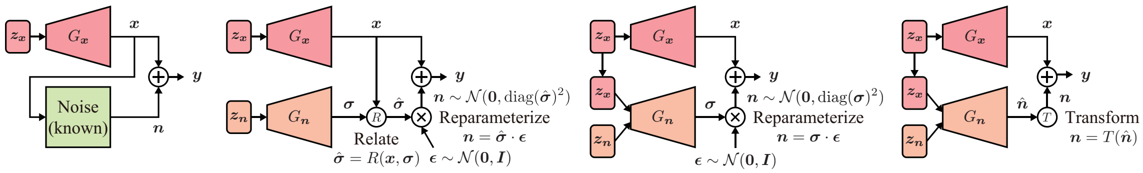
In SI-NR-GAN-I, $\boldsymbol{\sigma}$ is learned through training in a pixel-wise manner. Therefore, the same model can be applied to various noises (e.g., Figure 2(A)–(D), in which each pixel's noise follows a Gaussian distribution, while the noise amount is different in a sample-wise (e.g., (B)) or 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 $\boldsymbol{n}$ is rotation-, channel-shuffle-, or color-inverse-invariant. (ii) The signal $\boldsymbol{x}$ is rotation-, channel-shuffle-, or color-inverse-variant.*

[3]Strictly, our approach is applicable as long as a noise follows a differentiable distribution [39].



(a) AmbientGAN (baseline) (b) SD-NR-GAN-I (proposed) (c) SD-NR-GAN-II (proposed) (d) SD-NR-GAN-III (proposed)

Noise distribution type	Known	Known	Known	Unknown
Signal-noise relationship	Known	Known	Unknown	Unknown
Noise amount	Known	Unknown	Unknown	Unknown
Applicable noise	Specific modeling	(I), (J)/(M), (N)	(A)–(D), (I)–(P)	(A)–(P)

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.

Among the noises in Figure 2, this assumption holds in all signal-independent noises (A)–(H). This assumption is reasonable when $\boldsymbol{n}$ is a zero-mean signal-independent noise and $\boldsymbol{x}$ is a natural image.[4] Under this assumption, we establish SI-NR-GAN-II (Figure 3(d)). In this model, we redefine the noise generator as $\hat{\boldsymbol{n}} = G_{\boldsymbol{n}}(\boldsymbol{z_n})$ ($\hat{\boldsymbol{n}} \in \mathbb{R}^{H \times W \times C}$) and apply transformations to $\hat{\boldsymbol{n}}$ by $\boldsymbol{n} = T(\hat{\boldsymbol{n}})$, where T is a transformation function. As T, we can use arbitrary transformation as long as it is applicable to $\boldsymbol{n}$ but not allowable to $\boldsymbol{x}$. In practice, we use three transformations: (i) *rotation* – rotating $\hat{\boldsymbol{n}}$ by $d \in \{0°, 90°, 180°, 270°\}$ 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 $\hat{\boldsymbol{n}}$ 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.*

[4]In fact, these kinds of transformations (especially, rotation) are commonly used in self-supervised learning [19, 41, 10, 48], which utilize the transformations to learn natural image representations. Inspired by this, we employ the transformations to isolate noises from natural images.

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

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}, \text{where } \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \mathrm{diag}(\boldsymbol{\sigma} \cdot \boldsymbol{x})^2). \quad (4)$$

To represent this noise with trainable $\boldsymbol{\sigma}$, we redesign the noise generator as $\boldsymbol{\sigma} = G_{\boldsymbol{n}}(\boldsymbol{z_n})$. Then, we convert $\boldsymbol{\sigma}$ using a signal-noise relational function $R(\boldsymbol{x}, \boldsymbol{\sigma}) = \boldsymbol{\sigma} \cdot \boldsymbol{x} = \hat{\boldsymbol{\sigma}}$. Finally, we obtain $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \mathrm{diag}(\hat{\boldsymbol{\sigma}})^2)$ by using the reparameterization trick described in Section 5.1.

Poisson noise (or shot noise) is sampled by $\boldsymbol{y} \sim \mathrm{Poisson}(\boldsymbol{\lambda} \cdot \boldsymbol{x})/\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ 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:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}, \text{where } \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \mathrm{diag}(\boldsymbol{\sigma} \cdot \sqrt{\boldsymbol{x}})^2), \quad (5)$$

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

In both noise cases, the noise amount $\boldsymbol{\sigma}$ 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 $R(\boldsymbol{x}, \boldsymbol{\sigma})$ *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 $\boldsymbol{z_x}$ into the input of $G_{\boldsymbol{n}}$, i.e.,

$\boldsymbol{\sigma} = G_{\boldsymbol{n}}(\boldsymbol{z_n}, \boldsymbol{z_x})$. Similarly to SI-NR-GAN-I, we sample $\boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \text{diag}(\boldsymbol{\sigma})^2)$ 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 $R(\boldsymbol{x}, \boldsymbol{\sigma})$; therefore, the same model can also be applied to signal-independent noise (Figure 2(A)–(D)), i.e., $R(\boldsymbol{x}, \boldsymbol{\sigma}) = \boldsymbol{x}$, and the combination of multiple noises (Figure 2(K)(L)(O)(P)), e.g., $R(\boldsymbol{x}, \boldsymbol{\sigma}_d, \boldsymbol{\sigma}_i) = \boldsymbol{\sigma}_d \cdot \boldsymbol{x} + \boldsymbol{\sigma}_i$.

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*.[5] 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 $\boldsymbol{z_x}$ into the input of $G_{\boldsymbol{n}}$, i.e., $\hat{\boldsymbol{n}} = G_{\boldsymbol{n}}(\boldsymbol{z_n}, \boldsymbol{z_x})$. Similarly to SI-NR-GAN-II, we impose a transformation constraint on $G_{\boldsymbol{n}}$ by applying $\boldsymbol{n} = T(\hat{\boldsymbol{n}})$, where T is defined as *color inversion*. The noise origin $\hat{\boldsymbol{n}}$ 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, $G_{\boldsymbol{x}}$ and $G_{\boldsymbol{n}}$ are learned simultaneously. Ideally, we expect that $G_{\boldsymbol{x}}$ and $G_{\boldsymbol{n}}$ would be optimized at the same speed; however, through experiments, we found that $G_{\boldsymbol{n}}$ tends to learn faster than $G_{\boldsymbol{x}}$ 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 $G_{\boldsymbol{n}}$. Intuitively, this regularization makes $G_{\boldsymbol{n}}$ sensitive to $\boldsymbol{z_n}$ 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 $G_{\boldsymbol{x}}$ 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 $\boldsymbol{x}$ before providing to $D_{\boldsymbol{y}}$. In particular, we found that applying vertical and horizontal blur filters respectively and providing both to $D_{\boldsymbol{y}}$ works well. In the experiments, we apply this technique to all GANs in the Poisson or Poisson-Gaussian noise setting.[6] 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 $60k$ 32×32 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 σ. 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 σ 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 $G_{\boldsymbol{n}}$. (4) **SD-NR-GAN-0**, which has the same generators as SD-NR-GAN-II and -III but has no constraint on $G_{\boldsymbol{n}}$.

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

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

[6]Strictly 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 $G_{\boldsymbol{x}}$ and $G_{\boldsymbol{n}}$. 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].

Signal-independent	(A) **AGF**	(B) **AGV**	(C) **LGF**	(D) **LGV**	(E) **U**	(F) **MIX**	(G) **BG**	(H) **A+G**
AmbientGAN†	26.7	28.0	21.8	21.7	28.3	25.1	30.3	40.8
P-AmbientGAN	33.9	122.2	38.8	38.0	43.0	32.3	164.2	269.7
GAN	145.8	136.0	38.8	38.8	146.4	125.6	165.3	265.9
SI-NR-GAN-0	40.7	39.5	23.1	24.3	38.6	32.7	71.6	139.7
SI-NR-GAN-I	**26.7**	**27.5**	**22.1**	22.4	40.1	**24.8**	163.4	253.2
SI-NR-GAN-II	29.8	29.7	**22.1**	**21.7**	**31.6**	26.5	**32.2**	**44.0**
CBM3D+GAN	35.1	38.4	37.0	33.9	38.9	30.2	136.6	169.1
N2V+GAN	34.6	36.7	22.7	22.6	36.4	32.0	163.8	247.8
N2N+GAN‡	33.5	36.5	22.4	22.0	32.4	30.7	29.5	48.3
Signal-dependent	(I) **MGF**	(J) **MGV**	(K) **A+I**	(L) **A+I**	(M) **PF**	(N) **PV**	(O) **A+M**	(P) **A+M**
AmbientGAN†	21.4	21.8	21.9	27.4	31.3	32.3	30.9	35.3
P-AmbientGAN	27.1	68.7	39.7	137.7	43.8	100.7	43.0	94.2
GAN	82.7	77.4	93.2	155.8	152.4	160.1	149.1	175.8
SD-NR-GAN-0	82.7	59.5	69.9	75.1	71.7	70.2	72.0	69.0
SD-NR-GAN-I	**22.5**	**23.0**	25.3	112.4	**30.8**	**32.0**	**31.4**	70.6
SD-NR-GAN-II	24.4	24.2	**23.3**	**28.5**	34.0	33.9	34.0	**35.4**
SD-NR-GAN-III	37.5	33.4	33.5	33.9	53.1	55.1	52.4	47.2
CBM3D+GAN	26.9	27.8	27.6	40.0	–	–	–	–
GAT-BM3D+GAN	–	–	–	–	38.4	40.2	38.7	50.1
N2V+GAN	25.8	26.6	26.7	36.4	37.1	38.3	37.5	41.2
N2N+GAN‡	24.9	26.6	26.2	34.1	36.7	39.7	36.4	39.5

Table 1. **Comparison of FID on CIFAR-10.** A smaller value is better. We compared 152 conditions. The second and thirteenth rows denote the abbreviations defined in Figure 2. We report the median score across three random seeds. The symbol † indicates that the ground-truth noise models are given. The symbol ‡ denotes that noisy image pairs are given during the training. The other models are trained using only noisy images (not including pairs) without complete noise information. Bold font indicates the best score except for the models denoted by †‡.

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-NR-GAN 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]

[7] We provide further analyses and examples of generated images in our arXiv [33].

Signal-independent	LSUN BEDROOM			FFHQ
	(A) **AGF**	(B) **AGV**	(G) **BG**	(A) **AGF**
AmbientGAN†	19.4	25.0	9.7	28.3
GAN	98.9	100.9	125.3	81.6
SI-NR-GAN-I	**13.8**	**14.2**	128.2	**35.7**
SI-NR-GAN-II	15.7	16.8	**10.8**	37.1

Signal-dependent	LSUN BEDROOM			FFHQ
	(I) **MGF**	(L) **A+I**	(M) **PF**	(I) **MGF**
AmbientGAN†	11.7	19.2	32.6	18.7
GAN	54.0	109.8	121.7	48.0
SD-NR-GAN-I	**11.6**	55.4	**23.3**	**26.5**
SD-NR-GAN-II	21.7	**15.0**	42.8	49.0
SD-NR-GAN-III	50.7	53.1	138.6	37.2

Table 2. **Comparison of FID on LSUN BEDROOM and FFHQ.** A smaller value is better. Because the training is time-consuming, experiments were run once. The notation is the same as that in Table 1.

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].[8] 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 calculationally 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* (*GN2GC*). In particular, we employed the generators that achieve the best FID in Table 2 (denoted by bold font).[9] 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].

[8] Strictly 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.

	LSUN BEDROOM			FFHQ	LSUN BEDROOM			FFHQ
	(A) **AGF**	(B) **AGV**	(G) **BG**	(A) **AGF**	(I) **MGF**	(L) **A+I**	(M) **PF**	(I) **MGF**
N2C♯	32.90	33.06	29.67	31.93	36.70	32.26	31.77	36.37
N2N‡	32.30	32.23	28.76	31.33	35.99	31.36	30.55	35.88
N2V	31.98	31.85	20.73	30.95	35.37	31.09	30.30	34.95
N2S	31.79	31.70	20.74	30.74	35.12	30.94	30.19	34.67
GN2GC	**32.36**	**32.47**	**26.61**	**31.34**	**36.01**	**31.62**	**31.08**	**35.69**
CBM3D	31.41	31.54	20.75	30.29	33.60	30.43	–	32.73
GAT-BM3D	–	–	–	–	–	–	29.80	–

Table 3. **Comparison of PSNR on LSUN BEDROOM and FFHQ.** A larger value is better. We report the median score across three random seeds. The symbols ♯‡ indicate that the models are trained in advantageous conditions (♯clean target images and ‡noisy image pairs are given, respectively). The other models are trained using only noisy images (not including pairs). Bold font indicates the best score except for the models denoted by ♯‡.

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.

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

References

[1] Josue Anaya and Adrian Barbu. RENOIR–A dataset for real low-light image noise reduction. *J. Vis. Commun. Image Represent.*, 51:144–154, 2018.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

[3] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In *ICML*, 2019.

[4] Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The Cramer distance as a solution to biased Wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

[5] Ashish Bora, Eric Price, and Alexandros G. Dimakis. AmbientGAN: Generative models from lossy measurements. In *ICLR*, 2018.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[7] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.

[8] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *CVPR*, 2018.

[9] Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On self modulation for generative adversarial networks. In *ICLR*, 2019.

[10] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised GANs via auxiliary rotation loss. In *CVPR*, 2019.

[11] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *ICIP*, 2007.

[12] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007.

[13] Steven Diamond, Vincent Sitzmann, Stephen Boyd, Gordon Wetzstein, and Felix Heide. Dirty pixels: Optimizing image classification architectures for raw sensor data. *arXiv preprint arXiv:1701.06487*, 2017.

[14] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. In *ICLR Workshop*, 2015.

[15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR*, 2017.

[16] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Trans. Image Process.*, 22(4):1620–1630, 2012.

[17] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745, 2006.

[18] Aritra Ghosh, Himanshu Kumar, and P.S. Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017.

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[20] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[22] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014.

[23] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. In *NIPS*, 2017.

[24] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019.

[25] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

[26] Samuel W. Hasinoff. Photon, Poisson noise. *Computer Vision: A Reference Guide*, pages 608–610, 2014.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a Nash equilibrium. In *NIPS*, 2017.

[29] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. In *NIPS*, 2009.

[30] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018.

[31] Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *ICDM*, 2016.

[32] Takuhiro Kaneko and Tatsuya Harada. Label-noise robust multi-domain image-to-image translation. *arXiv preprint arXiv:1905.02185*, 2019.

[33] Takuhiro Kaneko and Tatsuya Harada. Noise robust generative adversarial networks. *arXiv preprint arXiv:1911.11776*, 2019.

[34] Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. Class-distinct and class-mutual image generation with GANs. In *BMVC*, 2019.

[35] Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. Label-noise robust generative adversarial networks. In *CVPR*, 2019.

[36] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

[37] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[38] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

[39] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[40] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of GANs. *arXiv preprint arXiv:1705.07215*, 2017.

[41] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *CVPR*, 2019.

[42] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, 2009.

[43] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void – Learning denoising from single noisy images. In *CVPR*, 2019.

[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.

[45] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *ICML*, 2018.

[46] Jae Hyun Lim and Jong Chul Ye. Geometric GAN. *arXiv preprint arXiv:1705.02894*, 2017.

[47] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? A large-scale study. In *NeurIPS*, 2018.

[48] Mario Lucic, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *ICML*, 2019.

[49] Florian Luisier, Thierry Blu, and Michael Unser. Image denoising in mixed Poisson-Gaussian noise. *IEEE Trans. Image Process.*, 20(3):696–708, 2010.

[50] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009.

[51] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Trans. Image Process.*, 17(1):53–69, 2007.

[52] Markku Makitalo and Alessandro Foi. Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise. *IEEE Trans. Image Process.*, 22(1):91–103, 2012.

[53] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update". In *NIPS*, 2017.

[54] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

[55] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NIPS*, 2016.

[56] Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *ICLR*, 2019.

[57] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *ICML*, 2018.

[58] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[59] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017.

[60] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017.

[61] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, 2019.

[62] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015.

[63] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

[64] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[65] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving GANs using optimal transport. In *ICLR*, 2018.

[66] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. In *ICLR Workshop*, 2015.

[67] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. MemNet: A persistent memory network for image restoration. In *ICCV*, 2017.

[68] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018.

[69] Kiran Koshy Thekumparampil, Ashish Khetan, Zinan Lin, and Sewoong Oh. Robustness of conditional GANs to noisy labels. In *NeurIPS*, 2018.

[70] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016.

[71] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, 2017.

[72] Bing Xu, Junfei Zhang, Rui Wang, Kun Xu, Yong-Liang Yang, Chuan Li, and Rui Tang. Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation. *ACM Trans. Graph.*, 38(6):1–12, 2019.

[73] Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. In *ICLR*, 2019.

[74] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[75] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

[76] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019.

[77] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.*, 26(7):3142–3155, 2017.

[78] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Trans. Image Process.*, 27(9):4608–4622, 2018.

[79] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.

[80] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.

[81] Qian Zhao, Deyu Meng, Zongben Xu, Wangmeng Zuo, and Lei Zhang. Robust principal component analysis with complex noise. In *ICML*, 2014.

[82] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *CVPR*, 2016.

[83] Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and Sung-Eui Yoon. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Comput. Graph. Forum*, 34(2):667–681, 2015.