

Label-Noise Robust Generative Adversarial Networks

Takuhiro Kaneko¹ Yoshitaka Ushiku¹ Tatsuya Harada^{1,2}

¹The University of Tokyo ²RIKEN

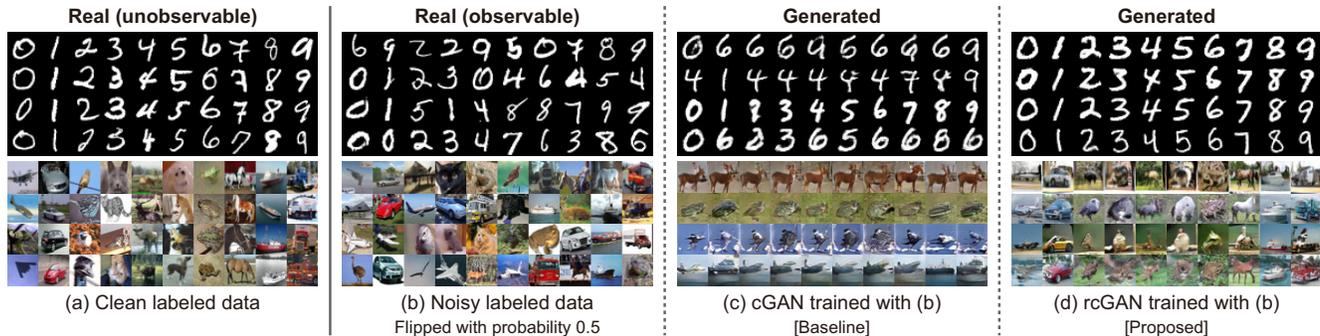


Figure 1. Examples of label-noise robust conditional image generation. Each column shows samples belonging to the same class. In (c) and (d), each row contains samples generated with a fixed z and a varied y^g . Our goal is, given noisy labeled data (b), to learn a conditional generative distribution that corresponds with clean labeled data (a). When naive cGAN (c) is trained with (b), it fails to learn the disentangled representations, disturbed by noisy labeled data. In contrast, proposed rcGAN (d) succeeds in learning the representations disentangled on the basis of clean labels, which are close to (a), even when we can only access the noisy labeled data (b) during training.

Abstract

Generative adversarial networks (GANs) are a framework that learns a generative distribution through adversarial training. Recently, their class conditional extensions (e.g., conditional GAN (cGAN) and auxiliary classifier GAN (AC-GAN)) have attracted much attention owing to their ability to learn the disentangled representations and to improve the training stability. However, their training requires the availability of large-scale accurate class-labeled data, which are often laborious or impractical to collect in a real-world scenario. To remedy this, we propose a novel family of GANs called label-noise robust GANs (rGANs), which, by incorporating a noise transition model, can learn a clean label conditional generative distribution even when training labels are noisy. In particular, we propose two variants: rAC-GAN, which is a bridging model between AC-GAN and the label-noise robust classification model, and rcGAN, which is an extension of cGAN and solves this problem with no reliance on any classifier. In addition to providing the theoretical background, we demonstrate the effectiveness of our models through extensive experiments using diverse GAN configurations, various noise settings, and multiple evaluation metrics (in which we tested 402 conditions in total).

1. Introduction

In computer vision and machine learning, generative modeling has been actively studied to generate or reproduce samples indistinguishable from real data. Recently, deep generative models have emerged as a powerful framework for addressing this problem. Among them, generative adversarial networks (GANs) [14], which learn a generative distribution through adversarial training, have become a prominent one owing to their ability to learn any data distribution without explicit density estimation. This mitigates oversmoothing resulting from data distribution approximation, and GANs have succeeded in producing high-fidelity data for various tasks [24, 38, 69, 7, 19, 29, 53, 25, 66, 77, 18, 31, 10, 61, 60, 76, 8, 22].

Along with this success, various extensions of GANs have been proposed. Among them, class conditional extensions (e.g., conditional GAN (cGAN) [37, 39] and auxiliary classifier GAN (AC-GAN) [41]) have attracted much attention mainly for two reasons. (1) By incorporating class labels as supervision, they can learn the representations that are disentangled between the class labels and the others. This allows them to selectively generate images conditioned on the class labels [37, 41, 22, 73, 23, 10]. Recently, this usefulness has also been demonstrated in class-specific data augmentation [12, 75]. (2) The added supervision simpli-

fies the learned target from an overall distribution to the conditional distribution. This helps stabilize the GAN training, which is typically unstable, and improves image quality [41, 39, 69, 7].

In contrast to these powerful properties, a possible limitation is that typical models rely on the availability of large-scale accurate class-labeled data and their performance depends on their accuracy. Indeed, as shown in Figure 1(c), when conventional cGAN is applied to noisy labeled data (where half labels are randomly flipped, as shown in Figure 1(b)), its performance is significantly degraded, influenced by the noisy labels. When datasets are constructed in real-world scenarios (e.g., crawled from websites or annotated via crowdsourcing), they tend to contain many mislabeled data (e.g., in Clothing1M [63], the overall annotation accuracy is only 61.54%). Therefore, this limitation would restrict application.

Motivated by these backgrounds, we address the following problem: “How can we learn a clean label conditional distribution even when training labels are noisy?” To solve this problem, we propose a novel family of GANs called *label-noise robust GANs (rGANs)* that incorporate a noise transition model representing a transition probability between the clean and noisy labels. In particular, we propose two variants: *rAC-GAN*, which is a bridging model between AC-GAN [41] and the label-noise robust classification model, and *rcGAN*, which is an extension of cGAN [37, 39] and solves this problem with no reliance on any classifier. As examples, we show generated image samples using rcGAN in Figure 1(d). As shown in this figure, our rcGAN is able to generate images conditioned on clean labels even where conventional cGAN suffers from severe degradation.

Another important issue regarding learning deep neural networks (DNNs) using noisy labeled data is the memorization effect. In image classification, a recent study [67] empirically demonstrated that DNNs can fit even noisy (or random) labels. Another study [5] experimentally showed that there are qualitative differences between DNNs trained on clean and noisy labeled data. To the best of our knowledge, no previous studies have sufficiently examined such an effect for conditional deep generative models. Motivated by these facts, in addition to providing a theoretical background on rAC-GAN and rcGAN, we conducted extensive experiments to examine the gap between theory and practice. In particular, we evaluated our models using diverse GAN configurations from standard to state-of-the-art in various label-noise settings including synthetic and real-world noise. We also tested our methods in the case when a noise transition model is known and in the case when it is not. Furthermore, we introduce an improved technique to stabilize training in a severely noisy setting (e.g., that in which 90% of the labels are corrupted) and show the effectiveness.

Overall, our contributions are summarized as follows:

- We tackle a novel problem called *label-noise robust conditional image generation*, in which the goal is to learn a clean label conditional generative distribution even when training labels are noisy.
- To solve this problem, we propose a new family of GANs called *rGANs* that incorporate a noise transition model into conditional extensions of GANs. In particular, we propose two variants, i.e., *rcGAN* and *rAC-GAN*, for the two representative class conditional GANs, i.e., cGAN and AC-GAN.
- In addition to providing a theoretical background, we examine the gap between theory and practice through extensive experiments (in which we tested 402 conditions in total). Details and more results are available at <https://takuhirok.github.io/rGAN/>.

2. Related work

Deep generative models. Generative modeling has been a fundamental problem and has been actively studied in computer vision and machine learning. Recently, deep generative models have emerged as a powerful framework. Among them, three popular approaches are GANs [14], variational autoencoders (VAEs) [27, 50], and autoregressive models (ARs) [57]. All these models have pros and cons. One well-known problem with GANs is training instability; however, the recent studies have been making a great stride in solving this problem [11, 43, 51, 74, 3, 4, 35, 15, 24, 62, 38, 36, 69, 7]. In this paper, we focus on GANs because they have flexibility to the data representation, allowing for incorporating a noise transition model. However, with regard to VAEs and ARs, conditional extensions [26, 34, 65, 58, 47] have been proposed, and incorporating our ideas into them is a possible direction of future work.

Conditional extensions of GANs. As discussed in Section 1, conditional extensions of GANs have been actively studied to learn the representations that are disentangled between the conditional information and the others or to stabilize training and boost image quality. Other than class or attribute labels [37, 41, 22, 73, 23, 10], texts [45, 71, 70, 64], object locations [44], images [11, 19, 29, 61], or videos [60] are used as conditional information, and the effectiveness of conditional extensions of GANs has also been verified for them. In this paper, we focus on the situation in which noise exists in the label domain because obtaining robustness in such a domain has been a fundamental and important problem in image classification and has been actively studied, as discussed in the next paragraph. However, also in other domains (e.g., texts or images), it is highly likely that noise may exist when data are collected in real-world scenarios (e.g., crawled from websites or annotated via crowdsourcing). We believe that our findings would help the research also in these domains.

Label-noise robust models. Learning with noisy labels has been keenly studied since addressed in the learning theory community [1, 40]. Lately, this problem has also been studied in image classification with DNNs. For instance, to obtain label-noise robustness, one approach replaces a typical cross-entropy loss with a noise-tolerant loss [2, 72]. Another approach cleans up labels or selects clean labels out of noisy labels using neural network predictions or gradient directions [46, 55, 33, 20, 49, 16]. The other approach incorporates a noise transition model [54, 21, 42, 13], similarly to ours. These studies show promising results in both theory and practice and our study is based on their findings.

The main difference from them is that their goal is to obtain label-noise robustness in image classification, but our goal is to obtain such robustness in conditional image generation. We remark that our developed rAC-GAN internally uses a classifier; thus, it can be viewed as a bridging model between noise robust image classification and conditional image generation. Note that we also developed rcGAN, which is a classifier-free model, motivated by the recent studies [41, 39] that indicate that AC-GAN tends to lose diversity through a side effect of generating recognizable (i.e., classifiable) images. Another related topic is *pixel*-noise robust image generation [6, 30]. The difference from them is that they focused on the noise inserted in a *pixel* domain, but we focus on the noise in a *label* domain.

3. Notation and problem statement

We begin by defining notation and the problem statement. Throughout, we use superscript r to denote the real distribution and g the generative distribution. Let $\mathbf{x} \in \mathcal{X}$ be the target data (e.g., images) and $y \in \mathcal{Y}$ the corresponding class label. Here, \mathcal{X} is the data space $\mathcal{X} \subseteq \mathbb{R}^d$, where d is the dimension of the data, and \mathcal{Y} is the label space $\mathcal{Y} = \{1, \dots, c\}$, where c is the number of classes. We assume that y is noisy (and we denote such noisy label by \tilde{y}) and there exists a corresponding clean label \hat{y} that we cannot observe during training. In particular, we assume *class-dependent* noise in which each clean label $\hat{y} = i$ is corrupted to a noisy label $\tilde{y} = j$ with a probability $p(\tilde{y} = j | \hat{y} = i) = T_{i,j}$, independently of \mathbf{x} , where we define a noise transition matrix as $T = (T_{i,j}) \in [0, 1]^{c \times c}$ ($\sum_i T_{i,j} = 1$). Note that this assumption is commonly used in label-noise robust image classification (e.g., [2, 72, 54, 21, 42, 13]).

Our task is, when given noisy labeled samples $(\mathbf{x}^r, \tilde{y}^r) \sim \tilde{p}^r(\mathbf{x}, \tilde{y})$, to construct a label-noise robust conditional generator such that $\hat{p}^g(\mathbf{x}, \hat{y}) = \tilde{p}^r(\mathbf{x}, \hat{y})$, which can generate \mathbf{x} conditioned on *clean* \hat{y} rather than conditioned on *noisy* \tilde{y} . This task is challenging for typical conditional generative models, such as AC-GAN [41] (Figure 2(b)) and cGAN [37, 39] (Figure 2(d)), because they attempt to construct a generator conditioned on the observable labels; i.e., in this case, they attempt to construct a noisy-label-

dependent generator that generates \mathbf{x} conditioned on *noisy* \tilde{y} rather than conditioned on *clean* \hat{y} . Our main idea for solving this problem is to incorporate a noise transition model, i.e., $p(\tilde{y} | \hat{y})$, into these models (viewed as orange rectangles in Figures 2(c) and (e)). In particular, we develop two variants: rAC-GAN and rcGAN. We describe their details in Sections 4 and 5, respectively.

4. Label-noise robust AC-GAN: rAC-GAN

4.1. Background: AC-GAN

AC-GAN [41] is one of representative conditional extensions of GANs [14]. AC-GAN learns a conditional generator G that transforms noise \mathbf{z} and label y^g into data $\mathbf{x}^g = G(\mathbf{z}, y^g)$ with two networks. One is a discriminator D that assigns probability $p = D(\mathbf{x})$ for samples $\mathbf{x} \sim p^r(\mathbf{x})$ and assigns $1 - p$ for samples $\mathbf{x} \sim p^g(\mathbf{x})$. The other is an auxiliary classifier $C(y|\mathbf{x})$ that represents a probability distribution over class labels given \mathbf{x} . These networks are optimized by using two losses, namely, an adversarial loss and an auxiliary classifier loss.

Adversarial loss. An adversarial loss is defined as

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{\mathbf{x}^r \sim p^r(\mathbf{x})} [\log D(\mathbf{x}^r)] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), y^g \sim p(y)} [\log(1 - D(G(\mathbf{z}, y^g)))] \quad (1)$$

where D attempts to find the best decision boundary between real and generated data by maximizing this loss, and G attempts to generate data indistinguishable by D by minimizing this loss.

Auxiliary classifier loss. An auxiliary classifier loss is used to make the generated data belong to the target class. To achieve this, first C is optimized using a classification loss of real data:

$$\mathcal{L}_{\text{AC}}^r = \mathbb{E}_{(\mathbf{x}^r, y^r) \sim p^r(\mathbf{x}, y)} [-\log C(y = y^r | \mathbf{x}^r)], \quad (2)$$

where C learns to classify real data to the corresponding class by minimizing this loss. Then, G is optimized by using a classification loss of generated data:

$$\mathcal{L}_{\text{AC}}^g = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), y^g \sim p(y)} [-\log C(y = y^g | G(\mathbf{z}, y^g))], \quad (3)$$

where G attempts to generate data belonging to the corresponding class by minimizing this loss.

Full objective. In practice, shared networks between D and C are commonly used [41, 15]. In this setting, the full objective is written as

$$\mathcal{L}_{D/C} = -\mathcal{L}_{\text{GAN}} + \lambda_{\text{AC}}^r \mathcal{L}_{\text{AC}}^r, \quad (4)$$

$$\mathcal{L}_G = \mathcal{L}_{\text{GAN}} + \lambda_{\text{AC}}^g \mathcal{L}_{\text{AC}}^g, \quad (5)$$

where λ_{AC}^r and λ_{AC}^g are the trade-off parameters between the adversarial loss and the auxiliary classifier loss for the real and generated data, respectively. D/C and G are optimized by minimizing $\mathcal{L}_{D/C}$ and \mathcal{L}_G , respectively.

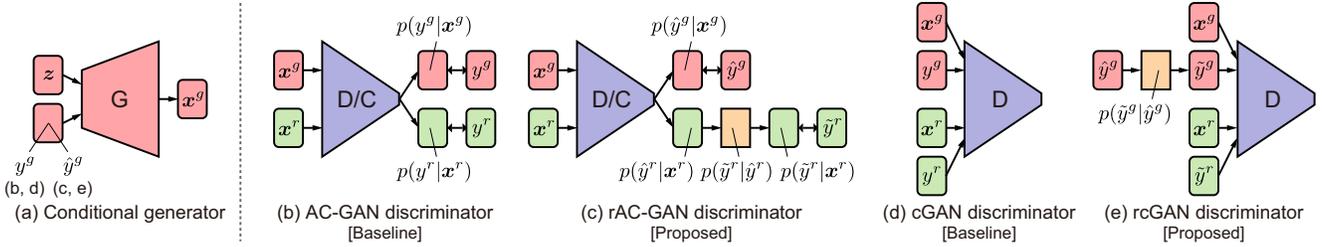


Figure 2. Comparison of naive and label-noise robust GANs. We denote the generator, discriminator, and auxiliary classifier by G , D , and C , respectively. Among all models, conditional generators (a) are similar. In our rAC-GAN (c) and rcGAN (e), we incorporate a noise transition model (viewed as an orange rectangle) into AC-GAN (b) and cGAN (d), respectively.

4.2. rAC-GAN

By the above definition, when y^r is noisy (i.e., \tilde{y}^r is given) and C fits such noisy labels,¹ AC-GAN learns the *noisy* label conditional generator $G(z, \tilde{y}^g)$. In contrast, our goal is to construct the *clean* label conditional generator $G(z, \hat{y}^g)$. To achieve this goal, we incorporate a noise transition model (i.e., $p(\tilde{y}|\hat{y})$; viewed as an orange rectangle in Figure 2(c)) into the auxiliary classifier. In particular, we reformulate the auxiliary classifier loss as

$$\begin{aligned} \mathcal{L}_{\text{rAC}}^r &= \mathbb{E}_{(\mathbf{x}^r, \tilde{y}^r) \sim \tilde{p}^r(\mathbf{x}, \tilde{y})} [-\log \tilde{C}(\tilde{y} = \tilde{y}^r | \mathbf{x}^r)] \\ &= \mathbb{E}_{(\mathbf{x}^r, \tilde{y}^r) \sim \tilde{p}^r(\mathbf{x}, \tilde{y})} [-\log \sum_{\hat{y}^r} p(\tilde{y} = \tilde{y}^r | \hat{y} = \hat{y}^r) \hat{C}(\hat{y} = \hat{y}^r | \mathbf{x}^r)] \\ &= \mathbb{E}_{(\mathbf{x}^r, \tilde{y}^r) \sim \tilde{p}^r(\mathbf{x}, \tilde{y})} [-\log \sum_{\hat{y}^r} T_{\tilde{y}^r, \hat{y}^r} \hat{C}(\hat{y} = \hat{y}^r | \mathbf{x}^r)], \quad (6) \end{aligned}$$

where we denote the *noisy* label classifier by \tilde{C} and the *clean* label classifier by \hat{C} (and we explain the reason why we call it *clean* in Theorem 1). Between the first and second lines, we assume that the noise transition is independent of \mathbf{x} , as mentioned in Section 3. Note that this formulation (called the *forward correction*) is often used in label-noise robust classification models [54, 21, 42, 13] and rAC-GAN can be viewed as a bridging model between GANs and them. In naive AC-GAN, \tilde{C} is optimized for $\mathcal{L}_{\text{AC}}^r$, whereas in our rAC-GAN, \hat{C} is optimized for $\mathcal{L}_{\text{rAC}}^r$. Similarly, G is optimized using \hat{C} rather than using \tilde{C} :

$$\mathcal{L}_{\text{rAC}}^g = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), \hat{y}^g \sim p(\hat{y})} [-\log \hat{C}(\hat{y} = \hat{y}^g | G(\mathbf{z}, \hat{y}^g))]. \quad (7)$$

Theoretical background. In the above, we use a cross-entropy loss, which is a kind of proper composite loss [48]. In this case, Theorem 2 in [42] shows that minimizing the

¹Zhang et al. [67] discuss generalization and memorization of DNNs and empirically demonstrated that DNNs are capable of fitting even noisy (or random) labels. Although other studies empirically demonstrated that some techniques (e.g., dropout [5], mixup [68], and high learning rate [55]) are useful for preventing DNNs from memorizing noisy labels, their theoretical support still remains as an open issue. In this paper, we conducted experiments on various GAN configurations to investigate such effect in our task. See Section 7.1 for details.

forward corrected loss (i.e., Equation 6) is equal to minimizing the original loss under the clean distribution. More precisely, the following theorem holds.

Theorem 1. *When T is nonsingular,*

$$\begin{aligned} &\operatorname{argmin}_{\hat{C}} \mathbb{E}_{(\mathbf{x}^r, \tilde{y}^r) \sim \tilde{p}^r(\mathbf{x}, \tilde{y})} [-\log \sum_{\hat{y}^r} T_{\tilde{y}^r, \hat{y}^r} \hat{C}(\hat{y} = \hat{y}^r | \mathbf{x}^r)] \\ &= \operatorname{argmin}_{\hat{C}} \mathbb{E}_{(\mathbf{x}^r, \hat{y}^r) \sim \hat{p}^r(\mathbf{x}, \hat{y})} [-\log \hat{C}(\hat{y} = \hat{y}^r | \mathbf{x}^r)]. \quad (8) \end{aligned}$$

For a detailed proof, refer to Theorem 2 in [42]. This supports the idea that, by minimizing $\mathcal{L}_{\text{rAC}}^r$ for noisy labeled samples, we can obtain \hat{C} that classifies \mathbf{x} as its corresponding clean label \hat{y} . In rAC-GAN, G is optimized for this *clean* classifier \hat{C} ; hence, in G 's input space, \hat{y}^g is encouraged to represent clean labels.

5. Label-noise robust cGAN: rcGAN

5.1. Background: cGAN

cGAN [37, 39] is another representative conditional extension of GANs [14]. In cGAN, a conditional generator $G(z, y^g)$ and a conditional discriminator $D(x, y)$ are jointly trained using a conditional adversarial loss.

Conditional adversarial loss. A conditional adversarial loss is defined as

$$\begin{aligned} \mathcal{L}_{\text{cGAN}} &= \mathbb{E}_{(\mathbf{x}^r, y^r) \sim p^r(\mathbf{x}, y)} [\log D(\mathbf{x}^r, y^r)] \\ &\quad + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), y^g \sim p(y)} [\log(1 - D(G(\mathbf{z}, y^g), y^g))], \quad (9) \end{aligned}$$

where D attempts to find the best decision boundary between real and generated data conditioned on y by maximizing this loss. In contrast, G attempts to generate data indistinguishable by D with a constraint on y^g by minimizing this loss. In an optimal condition [14], cGAN learns $G(z, y)$ such that $p^g(\mathbf{x}, y) = p^r(\mathbf{x}, y)$.

5.2. rcGAN

By the above definition, when y^r is noisy (i.e., \tilde{y}^r is given), cGAN learns the *noisy* label conditional generator $G(z, \tilde{y}^g)$. In contrast, our goal is to construct the *clean* label conditional generator $G(z, \hat{y}^g)$. To achieve this goal, we

insert a noise transition model (viewed as an orange rectangle in Figure 2(e)) before \hat{y}^g is given to D . In particular, we sample \tilde{y}^g from $\tilde{y}^g \sim p(\tilde{y}|\hat{y}^g)$ and redefine Equation 9 as

$$\begin{aligned} \mathcal{L}_{\text{rcGAN}} = & \mathbb{E}_{(\mathbf{x}^r, \tilde{y}^r) \sim \tilde{p}^r(\mathbf{x}, \tilde{y})} [\log D(\mathbf{x}^r, \tilde{y}^r)] \\ & + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), \hat{y}^g \sim p(\hat{y}), \tilde{y}^g \sim p(\tilde{y}|\hat{y}^g)} [\log(1 - D(G(\mathbf{z}, \hat{y}^g), \tilde{y}^g))], \end{aligned} \quad (10)$$

where D attempts to find the best decision boundary between real and generated data conditioned on *noisy* labels \tilde{y} , by maximizing this loss. In contrast, G attempts to generate data indistinguishable by D with a constraint on *clean* labels \hat{y}^g (and we explain the rationale behind calling it *clean* in Theorem 2), by minimizing this loss.

Theoretical background. In an optimal condition, the following theorem holds.

Theorem 2. *When T is nonsingular (i.e., T has a unique inverse), G is optimal if and only if $\hat{p}^g(\mathbf{x}, \hat{y}) = \hat{p}^r(\mathbf{x}, \hat{y})$.*

Proof. For G fixed, rcGAN is the same as cGAN where y is replaced by \tilde{y} . Therefore, by extending Proposition 1 and Theorem 1 in [14] (GAN optimal solution) to a conditional setting, the optimal discriminator D for fixed G is

$$D(\mathbf{x}, \tilde{y}) = \frac{\tilde{p}^r(\mathbf{x}, \tilde{y})}{\tilde{p}^r(\mathbf{x}, \tilde{y}) + \tilde{p}^g(\mathbf{x}, \tilde{y})}. \quad (11)$$

Then G is optimal if and only if

$$\tilde{p}^g(\mathbf{x}, \tilde{y}) = \tilde{p}^r(\mathbf{x}, \tilde{y}). \quad (12)$$

As mentioned in Section 3, we assume that label corruption occurs with $p(\tilde{y}|\hat{y})$, i.e., independently of \mathbf{x} . In this case,

$$\begin{aligned} \tilde{p}(\mathbf{x}, \tilde{y}) &= \tilde{p}(\tilde{y}|\mathbf{x})p(\mathbf{x}) = \sum_{\hat{y}} p(\tilde{y}|\hat{y})\hat{p}(\hat{y}|\mathbf{x})p(\mathbf{x}) \\ &= \sum_{\hat{y}} p(\tilde{y}|\hat{y})\hat{p}(\mathbf{x}, \hat{y}) = \sum_{\hat{y}} T_{\tilde{y}, \hat{y}}\hat{p}(\mathbf{x}, \hat{y}). \end{aligned} \quad (13)$$

Substituting Equation 13 into Equation 12 gives

$$\sum_{\hat{y}} T_{\tilde{y}, \hat{y}}\hat{p}^g(\mathbf{x}, \hat{y}) = \sum_{\hat{y}} T_{\tilde{y}, \hat{y}}\hat{p}^r(\mathbf{x}, \hat{y}). \quad (14)$$

By considering the matrix form,

$$T^\top \hat{P}^g = T^\top \hat{P}^r, \quad (15)$$

where $\hat{P}^g = [\hat{p}^g(\mathbf{x}, \hat{y} = 1), \dots, \hat{p}^g(\mathbf{x}, \hat{y} = c)]^\top$ and $\hat{P}^r = [\hat{p}^r(\mathbf{x}, \hat{y} = 1), \dots, \hat{p}^r(\mathbf{x}, \hat{y} = c)]^\top$. When T has an inverse,

$$T^\top \hat{P}^g = T^\top \hat{P}^r \Leftrightarrow \hat{P}^g = (T^\top)^{-1} T^\top \hat{P}^r = \hat{P}^r. \quad (16)$$

As the corresponding elements in \hat{P}^g and \hat{P}^r are equal, $\hat{p}^g(\mathbf{x}, \hat{y}) = \hat{p}^r(\mathbf{x}, \hat{y})$. \square

This supports the idea that, in an optimal condition, rcGAN learns $G(\mathbf{z}, \hat{y})$ such that $\hat{p}^g(\mathbf{x}, \hat{y}) = \hat{p}^r(\mathbf{x}, \hat{y})$.

6. Advanced techniques for practice

6.1. Noise transition probability estimation

In the above, we assume that T is known, but this assumption may be too strict for real-world applications. However, fortunately, previous studies [54, 21, 42, 13] have been eagerly tackling this problem and several methods for estimating T' (where we denote the estimated T by T') have been proposed. Among them, we tested a *robust two-stage training algorithm* [42] in the experiments and analyzed the effects of estimated T' . We show the results in Section 7.2.

6.2. Improved technique for severely noisy data

Thorough extensive experiments, we find that some GAN configurations suffer from performance degradation in a severely noisy setting (e.g., in which 90% of the labels are corrupted). In this type of environment, each label is flipped with a high probability. This disturbs G from associating an image with a label. To strengthen their connection, we incorporate mutual information regularization [9]:

$$\mathcal{L}_{\text{MI}} = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), \hat{y}^g \sim p(\hat{y})} [-\log Q(\hat{y} = \hat{y}^g | G(\mathbf{z}, \hat{y}^g))], \quad (17)$$

where $Q(\hat{y}|\mathbf{x})$ is an auxiliary distribution approximating a true posterior $p(\hat{y}|\mathbf{x})$. We optimize G and Q by minimizing this loss with trade-off parameters λ_{MI}^g and λ_{MI}^q , respectively. This formulation is similar to Equation 7, but the difference is whether G is optimized for \hat{C} (optimized using real images and noisy labels) or for Q (optimized using generated images and clean labels). We demonstrate the effectiveness of this technique in Section 7.3.

7. Experiments

7.1. Comprehensive study

In Sections 4 and 5, we showed that our approach is theoretically grounded. However, generally, in DNNs, there is still a gap between theory and practice. In particular, the label-noise effect in DNNs just recently began to be discussed in image classification [67, 5], and it is demonstrated that such a gap exists. However, in conditional image generation, such an effect has not been sufficiently examined. To advance this research, we first conducted a comprehensive study, i.e., compared the performance of conventional AC-GAN and cGAN and proposed rAC-GAN and rcGAN using diverse GAN configurations in various label-noise settings with multiple evaluation metrics.² Due to the space limitation, we briefly review the experimental setup and only provide the important results in this paper. See the supplementary materials at <https://takuhirok.github.io/rGAN/> for details and more results.

Dataset. We verified the effectiveness of our method on two benchmark datasets: CIFAR-10 and CIFAR-100 [28],

²Through Sections 7.1–7.3, we tested 392 conditions in total. For each condition, we trained two models with different initializations.



Figure 3. Generated image samples on CIFAR-10. Each column shows samples belonging to the same class. Each row contains samples generated with a fixed z and a varied y^g . In symmetric noise (a), cSN-GAN is primarily influenced by noisy labels and fails to learn the disentangled representations. In asymmetric noise (b), it is expected that fourth and sixth columns will include cat and dog, respectively. However, in AC-CT-GAN and cSN-GAN, these columns contain the inverse. As evidence, we list the accuracy in the fourth column for cat/dog classes in Table 1. These scores indicate that the proposed models are robust but the baselines are weak for the flipped classes.

which are commonly used in both image generation and label-noise robust image classification. Both datasets contain $60k$ 32×32 natural images, which are divided into $50k$ training and $10k$ test images. CIFAR-10 and CIFAR-100 have 10 and 100 classes, respectively. We assumed two label-noise settings that are popularly used in label-noise robust image classification: (1) **Symmetric** (class-independent) noise [59]: For all classes, ground truth labels are replaced with uniform random classes with probability μ . (2) **Asymmetric** (class-dependent) noise [42]: Ground truth labels are flipped with probability μ by mimicking real mistakes between similar classes. Following [42], for CIFAR-10, ground truth labels are replaced with *truck* \rightarrow *automobile*, *bird* \rightarrow *airplane*, *deer* \rightarrow *horse*, and *cat* \leftrightarrow *dog*, and for CIFAR-100, ground truth labels are flipped into the next class circularly within the same superclasses. In both settings, we selected μ from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$.

GAN configurations. A recent study [32] shows the sensitivity of GANs to hyperparameters. However, when clean labeled data are not available, it is impractical to tune the hyperparameters for each label-noise setting. Hence, instead of searching for the best model with hyperparameter tuning, we tested various GAN configurations using default parameters that are typically used in clean label settings and examined the label-noise effect. We chose four models to cover standard, widely accepted, and state-of-the-art models: **DCGAN** [43], **WGAN-GP** [15], **CT-GAN** [62], and **SNGAN** [38]. We implemented AC-GAN, rAC-GAN, cGAN, and rcGAN based on them. For cGAN and rcGAN, we used the *concat* discriminator [37] for DCGAN and the *projection* discriminator [39] for the others.

Evaluation metrics. As discussed in previous studies [56, 32, 52], evaluation and comparison of GANs can be challenging partially because of the lack of an explicit likelihood measure. Considering this fact, we used four metrics for a comprehensive analysis: (1) the Fréchet Inception distance (**FID**), (2) **Intra FID**, (3) the **GAN-test**, and (4) the **GAN-train**. The FID [17] measures the distance between p^r and p^g in Inception embeddings. We used it to assess the

	AC-CT-GAN	rAC-CT-GAN	cSN-GAN	rcSN-GAN
cat/dog	13.4/83.9	84.8/10.3	35.6/55.9	75.9/13.0

Table 1. Accuracy in the fourth column in Figure 3(b) (ground truth: cat) for the flipped classes (cat \leftrightarrow dog)

quality of an overall generative distribution. Intra FID [39] calculates the FID for each class. We used it to assess the quality of a conditional generative distribution.³ The GAN-test [52] is the accuracy of a classifier trained on real images and evaluated on generated images. This metric approximates the precision (image quality) of GANs. The GAN-train [52] is the accuracy of a classifier trained on generated images and evaluated on real images in a test. This metric approximates the recall (diversity) of GANs.

Results. We present the quantitative results for each condition in Figure 4 and provide a comparative summary between the proposed models (i.e., rAC-GAN and rcGAN) and the baselines (i.e., AC-GAN and cGAN) across all conditions in Figure 5. We show the generated image samples on CIFAR-10 with $\mu = 0.7$ in Figure 3. Regarding the FID (i.e., evaluating the quality of the overall generative distribution), the baselines and the proposed models are comparable in most cases, but when we use CT-GAN and SN-GAN (i.e., state-of-the-art models) in symmetric noise, the proposed models tend to outperform the baselines (32/40 conditions). This indicates that the label ambiguity caused by symmetric noise could disturb the learning of GANs if they have the high data-fitting ability. However, this degradation can be mitigated by using the proposed methods.

Regarding the other metrics (i.e., evaluating the quality of the conditional generative distribution), rAC-GAN and rcGAN tend to outperform AC-GAN and cGAN, respectively, across all the conditions. The one exception is rAC-WGAN-GP on CIFAR-10 with symmetric noise, but we find that it can be improved using the technique introduced in Section 6.2. We demonstrate this in Section 7.3. Among the four models, CT-GAN and SN-GAN work relatively well for rAC-GAN and rcGAN, respectively. This tendency

³We used Intra FID only for CIFAR-10 because, in CIFAR-100, the number of clean labeled data for each class (500) is too few.

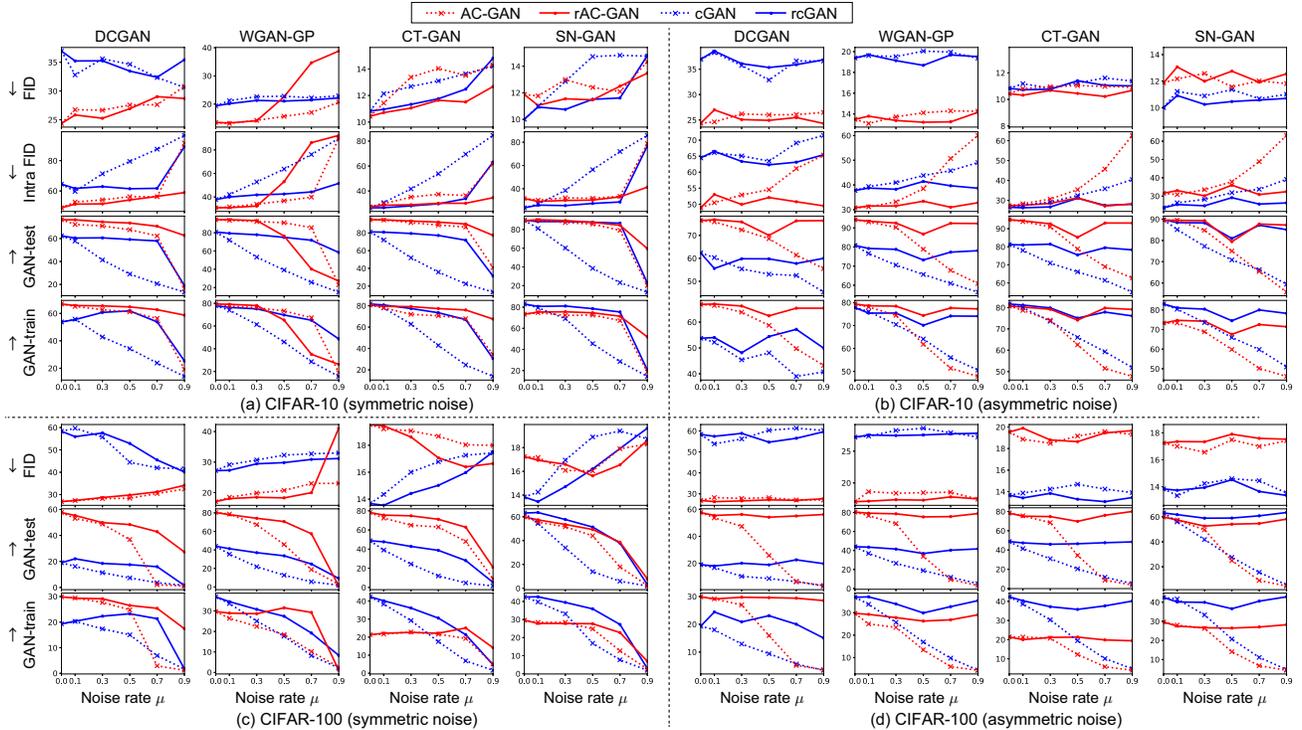


Figure 4. Quantitative results on CIFAR-10 and CIFAR-100. \downarrow indicates the smaller the value, the better the performance. \uparrow indicates the larger the value, the better the performance. Note that the scale is adjusted on each graph for ease of viewing.

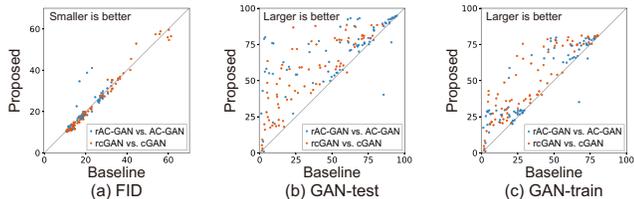


Figure 5. Comparison between the proposed models and the baselines across all the conditions in Figure 4.

	AC-GAN	rAC-GAN	cGAN	rcGAN
Symmetric	-0.846 \pm 0.084	-0.786 \pm 0.163	-0.989 \pm 0.013	-0.818 \pm 0.142
Asymmetric	-0.976 \pm 0.008	-0.476 \pm 0.119	-0.985 \pm 0.029	-0.427 \pm 0.274

Table 2. Pearson correlation coefficient between the noise rate and GAN-train. The scores are averaged over all GAN configurations. is also observed in clean label settings (i.e., $\mu = 0$). This indicates that the performance of rAC-GAN and rcGAN is closely related to the advance in the baseline GANs.

To analyze the dependency on the noise rate, we calculated the Pearson correlation coefficient between GAN-train and the noise rate. We list these in Table 2. These results indicate that cGAN has the highest dependency on the noise rate, while AC-GAN shows robustness for symmetric noise but weakness for asymmetric noise. This would be related to the memorization effect in a DNN classifier. cGAN is a classifier-free model; therefore, it learns the distribution conditioned on the labels regardless of whether they are noisy or clean. In contrast, AC-GAN internally uses a classifier that prioritizes learning simple (i.e., clean) labels [5]. In symmetric noise, the corruption variety is large, making it difficult to memorize labels. As a result, AC-GAN pri-

oritizes learning simple (i.e., clean) labels. In contrast, in asymmetric noise, the label corruption pattern is restrictive; as a result, AC-GAN easily fits noisy labels.

7.2. Effects of estimated T'

In Section 7.1, we report the results using known T . As a more practical setting, we also evaluate our method with T' estimated by a robust two-stage training algorithm [42]. We used CT-GAN for rAC-GAN and SN-GAN for rcGAN, which worked relatively well in both noisy and clean settings in Section 7.1. We list the scores in Table 3. In CIFAR-10, even using T' , rAC-CT-GAN and rcSN-GAN tend to outperform conventional AC-CT-GAN and cSN-GAN, respectively, and show robustness to label noise. In CIFAR-100, when the noise rate is low, rAC-CT-GAN and rcSN-GAN work moderately well; however, in highly noisy settings, their performance is degraded. Note that such a tendency has also been observed in noisy label image classification with T' [42], in which the authors argue that the high-rate mixture and limited number of images per class (500) make it difficult to estimate the correct T . Further improvement remains as an open issue.

7.3. Evaluation of improved technique

As shown in Figure 4, rAC-GAN and rcGAN show robustness for label noise in almost all cases, but we find that they are still weak to severely noisy settings (i.e., symmetric noise with $\mu = 0.9$) even though using known T . To

Model	Metric	CIFAR-10 (symmetric noise)					CIFAR-10 (asymmetric noise)					CIFAR-100 (symmetric noise)					CIFAR-100 (asymmetric noise)				
		0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
rAC-CT-GAN with T'	FID ↓	10.9	11.4	11.3	11.5	13.0	10.8	10.2	10.2	10.4	11.0	19.7	19.3	17.7	17.3	18.5	19.4	19.3	19.7	18.8	19.0
	Intra FID ↓	28.7	31.0	30.1	31.7	38.9	28.5	27.4	31.2	35.0	36.8	–	–	–	–	–	–	–	–	–	–
	GAN-test ↑	95.3	93.2	92.0	87.7	70.4	94.9	92.9	85.2	78.5	76.6	76.6	67.1	68.1	<i>1.0</i>	2.5	74.1	68.9	28.7	7.2	2.2
	GAN-train ↑	78.7	75.9	76.9	73.7	63.4	79.8	79.5	74.0	69.1	67.3	21.2	21.4	23.3	<i>1.0</i>	2.3	19.1	19.9	10.7	5.5	3.9
rcSN-GAN with T'	FID ↓	10.7	11.9	12.4	12.1	15.0	10.8	10.8	11.0	10.9	11.3	14.3	16.6	17.5	20.0	19.8	13.8	14.1	14.7	14.7	13.9
	Intra FID ↓	25.5	29.4	29.4	29.7	87.4	25.7	26.0	28.7	32.6	33.9	–	–	–	–	–	–	–	–	–	–
	GAN-test ↑	85.3	79.0	84.8	82.8	15.9	86.6	87.2	84.0	74.9	71.2	53.4	36.6	37.7	<i>1.0</i>	1.7	65.0	63.0	32.4	7.8	3.8
	GAN-train ↑	80.7	78.1	77.4	75.6	15.0	80.5	79.0	75.7	69.3	65.7	40.1	32.8	31.3	<i>1.0</i>	1.8	41.7	39.3	20.1	<i>6.1</i>	3.9

Table 3. Quantitative results using the estimated T' . The second row indicates a noise rate. Bold and italic fonts indicate that the score is better or worse by more than 3 points over or under the baseline models (i.e., AC-CT-GAN or cSN-GAN), respectively.

Model	Metric	CIFAR-10 (symmetric noise)				CIFAR-100 (symmetric noise)			
		A	B	C	D	A	B	C	D
Improved rAC-GAN	FID ↓	27.9	14.7	12.4	13.5	33.1	20.4	17.2	18.4
	Intra FID ↓	55.7	34.6	33.4	36.9	–	–	–	–
	GAN-test ↑	65.1	77.7	78.2	63.5	26.2	22.5	21.5	15.4
	GAN-train ↑	59.9	70.8	69.1	59.7	17.1	16.3	14.8	11.7
Improved rcGAN	FID ↓	30.4	16.9	14.2	14.9	<i>50.2</i>	25.8	18.0	18.7
	Intra FID ↓	76.9	39.6	52.9	48.2	–	–	–	–
	GAN-test ↑	27.3	65.7	38.9	48.8	4.5	12.0	9.5	6.1
	GAN-train ↑	31.9	60.7	36.7	47.3	6.0	10.3	7.5	4.4

Table 4. Quantitative results using the improved technique. In the second row, A, B, C, and D indicate DCGAN, WGAN-GP, CT-GAN, and SN-GAN, respectively. We evaluated in severely noisy settings (i.e., symmetric noise with $\mu = 0.9$). Bold and italic fonts indicate that the score is better or worse by more than 3 points over or under naive models (i.e., rAC-GAN or rcGAN), respectively.

Metric	Clean		Noisy				Mixed			
	AC	c	AC	rAC	c	rc	AC	rAC	c	rc
FID ↓	6.8	12.0	4.4	4.6	9.4	9.4	4.8	4.7	10.5	9.7
GAN-train ↑	56.6	53.9	49.5	51.7	48.6	49.8	52.8	57.0	51.7	55.0

Table 5. Quantitative results on Clothing1M. AC, rAC, c, and rc denote AC-CT-GAN, rAC-CT-GAN, cSN-GAN, and rcSN-GAN, respectively. Bold font indicates better scores in each block.

improve the performance, we developed an improved technique (Section 6.2). In this section, we validate its effect. We list the scores in Table 4. We find that the improved degree depends on the GAN configurations, but, on the whole, the performance is improved by the proposed technique. In particular, we find that the improved technique is most effective for rAC-WGAN-GP, in which all the scores doubled compared to those of naive rAC-WGAN-GP.

7.4. Evaluation on real-world noise

Finally, we tested on Clothing1M [63] to analyze the effectiveness on real-world noise.⁴ Clothing1M contains 1M clothing images in 14 classes. The data are collected from several online shopping websites and include many mislabeled samples. This dataset also contains 50k, 14k, and 10k of clean data for training, validation, and testing, respectively. Following the previous studies [63, 42], we approximated T using the partial (25k) training data that have both clean and noisy labels. We tested on three settings: (1) 50k **clean** data, (2) 1M **noisy** data, and (3) **mixed** data that consists of clean data (bootstrapped to 500k) and 1M noisy data, which are used in [63] to boost the performance of image classification. We used AC-CT-GAN/rAC-CT-GAN and cSN-GAN/rcSN-GAN. We resized images from 256×256 to 64×64 to shorten the training time.

⁴We tested 10 conditions in total. For each condition, we trained three models with different initializations.

Results. We list the scores in Table 5.⁵ The comparison of FID values indicates that the scores depend on the number of data (**noisy, mixed > clean**) rather than the difference between the baseline and proposed models. This suggests that, in this type noise setting, the scale of the dataset should be made large, even though labels are noisy, to capture an overall distribution. In contrast, the comparison of the GAN-train between the clean and noisy data settings indicates the importance of label accuracy. In the noisy data setting, the scores improve using rAC-GAN or rcGAN but they are still worse than those using AC-GAN and cGAN in the clean data setting. The balanced models are rAC-GAN and rcGAN in the mixed data setting. They are comparable to the models in the noisy data setting in terms of the FID and outperform the models in the clean data setting in terms of the GAN-train. Recently, data augmentation [12, 75] has been studied intensively as an application of conditional generative models. We expect the above findings to provide an important direction in this space.

8. Conclusion

Recently, conditional extensions of GANs have shown promise in image generation; however, the limitation here is that they need large-scale accurate class-labeled data to be available. To remedy this, we developed a new family of GANs called rGANs that incorporate a noise transition model into conditional extensions of GANs. In particular, we introduced two variants: rAC-GAN, which is a bridging model between GANs and the noise-robust classification models, and rcGAN, which is an extension of cGAN and solves this problem with no reliance on any classifier. In addition to providing a theoretical background, we demonstrate the effectiveness and limitations of the proposed models through extensive experiments in various settings. In the future, we hope that our findings facilitate the construction of a conditional generative model in real-world scenarios in which only noisy labeled data are available.

Acknowledgement. We thank Hiroharu Kato, Yusuke Mukuta, and Mikihiro Tanaka for helpful discussions. This work was supported by JSPS KAKENHI Grant Number JP17H06100, partially supported by JST CREST Grant Number JPMJCR1403, Japan, and partially supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) as ‘‘Seminal Issue on Post-K Computer.’’

⁵We did not use Intra FID because the number of clean labeled data for each class is few. We did not use the GAN-test because this dataset is challenging, and a learned classifier tends to be deceived by noisy data.

References

- [1] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988. 3
- [2] P. S. Aritra Ghosh, Himanshu Kumar. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017. 3
- [3] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017. 2
- [4] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 2
- [5] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien. A closer look at memorization in deep networks. In *ICML*, 2017. 2, 4, 5, 7
- [6] A. Bora, E. Price, and A. G. Dimakis. AmbientGAN: Generative models from lossy measurements. In *ICLR*, 2018. 3
- [7] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1, 2
- [8] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Neural photo editing with introspective adversarial networks. In *ICLR*, 2017. 1
- [9] X. Chen, Y. Duan, R. Houthoof, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016. 5
- [10] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 1, 2
- [11] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a Laplacian pyramid of adversarial networks. In *NIPS*, 2015. 2
- [12] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan. Synthetic data augmentation using GAN for improved liver lesion classification. In *ISBI*, 2018. 1, 8
- [13] J. Goldberger and E. Ben-Reuven. Training deep neural networks using a noise adaptation layer. In *ICLR*, 2017. 3, 4, 5
- [14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 2, 3, 4, 5
- [15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of Wasserstein GANs. In *NIPS*, 2017. 2, 3, 6
- [16] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*, 2018. 3
- [17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a Nash equilibrium. In *NIPS*, 2017. 6
- [18] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Trans. on Graph.*, 36(4):107:1–107:14, 2017. 1
- [19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1, 2
- [20] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018. 3
- [21] I. Jindal, M. Nokleby, and X. Chen. Learning deep networks from noisy labels with dropout regularization. In *ICDM*, 2016. 3, 4, 5
- [22] T. Kaneko, K. Hiramatsu, and K. Kashino. Generative attribute controller with conditional filtered generative adversarial networks. In *CVPR*, 2017. 1, 2
- [23] T. Kaneko, K. Hiramatsu, and K. Kashino. Generative adversarial image synthesis with decision tree latent controller. In *CVPR*, 2018. 1, 2
- [24] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 1, 2
- [25] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017. 1
- [26] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *NIPS*, 2014. 2
- [27] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2
- [28] A. Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, 2009. 5
- [29] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 1, 2
- [30] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2Noise: Learning image restoration without clean data. In *ICML*, 2018. 3
- [31] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017. 1
- [32] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are GANs created equal? A large-scale study. *arXiv preprint arXiv:1711.10337*, 2017. 6
- [33] E. Malach and S. Shalev-Shwartz. Decoupling "when to update" from "how to update". In *NIPS*, 2017. 3
- [34] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov. Generating images from captions with attention. In *ICLR*, 2016. 2
- [35] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 2
- [36] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for GANs do actually converge? In *ICML*, 2018. 2
- [37] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1, 2, 3, 4, 6
- [38] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 1, 2, 6

- [39] T. Miyato and M. Koyama. cGANs with projection discriminator. In *ICLR*, 2018. 1, 2, 3, 4, 6
- [40] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *NIPS*, 2013. 3
- [41] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, 2017. 1, 2, 3
- [42] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017. 3, 4, 5, 6, 7, 8
- [43] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2, 6
- [44] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *NIPS*, 2016. 2
- [45] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. 2
- [46] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015. 3
- [47] S. Reed, A. van den Oord, N. Kalchbrenner, V. Bapst, M. Botvinick, and N. de Freitas. Generating interpretable images with controllable structure. In *ICLR Workshop*, 2017. 2
- [48] M. D. Reid and R. C. Williamson. Composite binary losses. *JMLR*, 11:2387–2422, 2010. 4
- [49] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018. 3
- [50] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014. 2
- [51] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *NIPS*, 2016. 2
- [52] K. Shmelkov, C. Schmid, and K. Alahari. How good is my GAN? In *ECCV*, 2018. 6
- [53] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017. 1
- [54] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. In *ICLR Workshop*, 2015. 3, 4, 5
- [55] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018. 3, 4
- [56] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *ICLR*, 2016. 6
- [57] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 2
- [58] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelCNN decoders. In *NIPS*, 2016. 2
- [59] B. van Rooyen, A. Menon, and R. C. Williamson. Learning with symmetric label noise: The importance of being uninged. In *NIPS*, 2015. 6
- [60] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 1, 2
- [61] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *CVPR*, 2018. 1, 2
- [62] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang. Improving the improved training of Wasserstein GANs: A consistency term and its dual effect. In *ICLR*, 2018. 2, 6
- [63] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015. 2, 8
- [64] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. G. X. Huang, and X. He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018. 2
- [65] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *ECCV*, 2016. 2
- [66] Z. Yi, H. Zhang, P. Tan, and M. Gong. DualGAN: Unsupervised dual learning for image-to-image translation. In *ICCV*, 2017. 1
- [67] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017. 2, 4, 5
- [68] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 4
- [69] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 1, 2
- [70] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1710.10916*, 2017. 2
- [71] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 2
- [72] Z. Zhang and M. R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *arXiv preprint arXiv:1805.07836*, 2018. 3
- [73] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *CVPR*, 2017. 1, 2
- [74] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. In *ICLR*, 2017. 2
- [75] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang. Camera style adaptation for person re-identification. In *CVPR*, 2018. 1, 8
- [76] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 1
- [77] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 1