Label-Noise Robust Twin Auxiliary Classifier GANs

Michael Law

Abstract

Class-conditional generative models are fragile when facing noisily-labelled training data, and existing approaches combat this by assuming prior knowledge of the noise model or estimating the noise transition matrix. However, an accurate noise model is often difficult to obtain, and matrix estimation methods can become unreliable if the number of classes grows large. In this paper, we study the performance of Twin Auxiliary Classifier GAN (`TAC-GAN`) when confronted with label noise and class support overlaps in training data, providing theoretical insight to its deterioration in performance seen in empirical studies. We then propose two developments to `TAC-GAN` which integrate Co-Teaching and MentorNet, originally presented as methods for classification with noisy labels. Our integrated methods, named `CO-TAC` and `TAC-MENTOR`, assume no prior information on the label-noise model, and are transition matrix-free. Based on our empirical experiments, these methods perform at least as well as the baseline transition matrix-based methods across a range of configurations, with substantial outperformance in some cases.

1 Introduction

Since their invention, generative adversarial networks (GANs) [1] have been the centre of attention among generative models. The innovative approach of pitching networks against each other in an unsupervised competitive setting has proved effective in areas not limited to image generation, image-to-image translation [2–5], text generation [6, 7] and data augmentation [8]. Given that visual perception largely revolves around identifying and classifying entities, it is natural that conditional GANs (cGAN) [9] and its derivatives have formed one of the most active branches of GAN research. Amid the success of cGAN in generating convincing images from given class labels, a newer question has emerged: how a cGAN can be correctly and reliably trained in the presence of corrupt training labels?

Learning to *classify* data in the presence of label noise is a well-explored problem with long history [10]. Existing methods to mitigate label noise encompass surrogate losses, loss corrections and noise transition matrix estimation [11–14]. On the other hand, learning to subsequently *generate* data of correct classes is much more challenging, particularly high-dimensional data like images. Recent approaches have focused on assuming knowledge of, or explicitly estimating, the transition matrix of label corruption probabilities to improve the class accuracy of images generated by a

Date: June 30, 2020.

cGAN [15, 16]. However, having prior knowledge of the label-noise structure can be rather contrived, and matrix estimation methods may deteriorate as the number of classes increases.

Motivated by this discussion, we aim to explore new variants of cGANs that are robust to label noise yet do not make heavy assumptions on the learner's prior knowledge. In particular, the employment of auxiliary classifiers in recent cGAN iterations [17, 18] provides an opportunity to apply label-noise mitigation techniques from *classification* problems, in turn improving image *generation* outcomes. This work proposes two methods achieving this objective. Along the way, we elucidate some theoretical aspects of `TAC-GAN` [18] in the context of label-noise robustness (or lack thereof).

2 Related Work

2.1 GAN and cGAN

A GAN [1] consists of two networks: a generator G mapping random noise $\mathbf{z} \sim N(0, I)$ to synthetic images (represented by a matrix of pixel values), and a discriminator D estimating the probability that a given image is real, i.e. not generated by G. cGAN [9] is a basic extension of GAN to allow class-conditional image generation. cGAN takes G and D to be a conditional generator and conditional discriminator respectively; they are supplied with an additional class label y. G and D (parametrised by Θ_G and Θ_D respectively) train with a minimax objective

$$\min_{\Theta_G} \max_{\Theta_D} L_{cgan}(D, G) = \mathbb{E}_{(\mathbf{x},y)\sim p_{XY}}[\ln D(\mathbf{x}, y)] + \mathbb{E}_{\mathbf{z}\sim N(0,I), y\sim q_Y}[\ln(1 - D(G(\mathbf{z}, y), y))].$$

where p_{XY} is the density over (real image, ground-truth label) pairs, and q_Y is a prespecified distribution of generated labels. This forces G and D to compete: D learns to distinguish real images from those generated by G, while G learns to 'fool' D by mimicking the distribution of real images. Both networks improve through competition; in particular, G induces a conditional density $q_{X|Y}$ of images given labels, approximating the true conditional density $p_{X|Y}$.

2.2 `AC-GAN` and `TAC-GAN`

Auxiliary Classifier GAN (`AC-GAN`) [17] introduces a classifier C into the GAN ecosystem, trained from scratch alongside D and G. The loss function additionally penalises G for generating images of incorrect classes as perceived by C; that is, if $C(G(\mathbf{z}, y))$ predicts a small probability for class y, then the loss for G is large.

While `AC-GAN` demonstrably improves G's class accuracy in many GAN contexts, it suffers an image diversity issue which Twin AC-GAN (`TAC-GAN`) [18] addresses with minimal overhead. `TAC-GAN` introduces a *twin* auxiliary classifier C_{mi} which competes with G; G attempts to 'fool' C_{mi} into predicting incorrect classes, and it does so by learning to generate images of greater diversity. That said, G still adheres to the correct image classes to avoid high classification loss with respect to C, which we shall call the *primary* auxiliary classifier.

The objective functions for `AC-GAN` and `TAC-GAN` respectively are

$$\min_{\Theta_G,\Theta_C} \max_{\Theta_D} L_{ac}(D,G,C) = L_{cgan}(D,G) - \lambda_C \mathbb{E}_{(\mathbf{x},y)\sim p_{XY}}[\ln C(\mathbf{x},y)]$$
$$\min_{\Theta_G,\Theta_C} \max_{\Theta_D,\Theta_{C_{mi}}} L_{tac}(D,G,C,C_{mi}) = L_{ac}(D,G,C) + \lambda_C \mathbb{E}_{\mathbf{z}\sim N(0,I),y\sim q_Y}[\ln C_{mi}(G(\mathbf{z},y),y)].$$

2.3 Deep learning with noisy labels

Correcting noisy labels is an expensive task, so numerous methods to make classifiers robust to label noise have been proposed. One common approach is to model the noise transition matrix and appropriately modify the loss function [11, 12]. On the contrary, many recent works in this area have adopted a matrix-free approach. [19] posited that label corruption probabilities depend on the features of individual samples themselves, and proposed a framework making no assumptions on the noise distribution. [14] concurrently trains a complementary classifier to filter out corrupted samples. Co-Teaching [20] simultaneously trains two classifiers which teach each other based on their strengths and weaknesses in classification. In each mini-batch, each classifier identifies the samples which they have achieved small losses on, and instructs the other classifier to learn only from these samples. [21] and [22] build their approaches on having a small subset of training data with ground-truth labels known, which is feasible assumption to make in practice.

2.4 Label-noise robust cGANs

Generating data of correct classes is a more intricate (and newer) problem than learning to classify correctly in the presence of label noise. Robust Conditional GAN (`RCGAN`) [16] uses a known or estimated noise transition matrix to corrupt the labels of generated images, and employs an objective to force the generator into produce images of correct ground-truth labels. In a similar flavour, label-noise robust AC-GAN (`rAC-GAN`) [15] corrupts the auxiliary classifier's outputs according to the transition matrix, and uses a loss which makes the classifier gravitate towards predicting clean labels. However, one seldom knows the transition matrix in practical applications, and accurately estimating this matrix can be difficult in the presence of many image classes with few images per class. As such, [23] proposes a matrix-free method via weakly-supervised learning on complementary labels. There is still much room for exploration of label-noise robust cGANs where networks make no noise structure assumptions.

3 Theoretical Discussion

`TAC-GAN` is able to achieve both class accuracy and image diversity when trained on clean labels. In this section, we state theoretical results demonstrating that class accuracy is *not* upheld in the presence of label noise, thus justifying the need for architectural improvements to `TAC-GAN` to increase label-noise robustness, rather than mere hyperparameter tuning.

We must first define what it means for a conditional generator to perform well. Since a conditional GAN aims to learn a conditional density $q_{X|Y}$ 'close' to the true conditional density $p_{X|Y}$ of

images given labels, one may define an optimal conditional generator G to be one which induces $q_{X|Y} = p_{X|Y}$. We suggest that this is an overly restrictive definition, because a good generator need not satisfy $q_{X|Y}(x, y) = p_{X|Y}(x, y)$ for *all* image-label pairs (x, y). For example, why should we require $q_{X|Y}(\text{cat image}|\text{dog label}) \approx p_{X|Y}(\text{cat image}|\text{dog label})$, if G is *not* supposed to produce cat images from dog labels in the first place? It suffices to restrict our attention to image-label pairs where the ground-truth label is consistent with the image, hence the following definition.

Definition 1. *We say that a conditional generator G is **optimal** with respect to a true underlying conditional density $p_{X|Y}$ if G induces a conditional density $q_{X|Y}$ such that, for all labels k and images x,*

$$p_{Y|X}(k \mid x) = \max_j p_{Y|X}(j \mid x) \Rightarrow q_{X|Y}(x \mid k) = p_{X|Y}(x \mid k).$$

The equality on the left means we only care about having $q_{X|Y} = p_{X|Y}$ for image-label pairs where the label is the likeliest for the image, according to ground truth. Thus we have ignored the behaviour of $q_{X|Y}$ on inconsistent image-label pairs.

We assume that $p_{Y|X}(\cdot \mid x)$, given some image x, is not necessarily degenerate on some class k. This reflects the notion that an image does not always belong deterministically to a single class, because the sets of real images S_y belonging to different classes ('class supports') can overlap. Hence, when we refer to a 'ground-truth label distribution', we are referring to the *vector of ground-truth class probabilities* associated with an image. In general, the more ambiguous an image x is, the closer the probabilities $p_{Y|X}(k \mid x)$ are to each other. The greater the overlap between class supports, the greater the likelihood of sampling such an 'ambiguous' image. On the other hand, we only provide one-hot vectors as class labels while training.

Now we formulate the label-noise setting. Let $Y \sim Unif(\{1, \dots, n\})$ be the random variable of ground-truth labels, with mass function p_Y. Let the corruption probability from class i to class j be the entry Γ_{ij} of an $n \times n$ transition matrix Γ. Then the random variable of corrupt labels is $\tilde{Y} \in \{1, \dots, n\}$, and

$$\mathbb{P}\left(\tilde{Y} = j \mid Y = i\right) = \Gamma_{ij}\mathbb{P}\left(Y = i\right) = \Gamma_{ij}/n.$$

The following propositions demonstrate the theoretical performance of `TAC-GAN` under uniform label noise, i.e. where Γ_{ij} is constant for all $i \neq j$.

Proposition 1. *Suppose the label-noise matrix Γ represents uniform corruption with probability c, i.e. $\Gamma_{ii} = 1 - c$ and $\Gamma_{ij} = \frac{c}{n-1}$ for $i \neq j$. Then the conditional generator G in `TAC-GAN` induces the conditional density*

$$q_{X|Y}(x \mid k) \approx (1 - c)p_{X|Y}(x \mid k) + \frac{c}{n-1} \cdot \frac{p_X(x)}{p_Y(k)} \sum_{j=1, j\neq k}^{n} p_{Y|X}(j \mid x).$$

Proposition 2. *In the uniform label-noise setting with n classes and a positive probability of label corruption, the `TAC-GAN` generator is not optimal (in the sense of Definition 1). Moreover, this is caused by the primary auxiliary classifier learning a biased distribution for labels given images, $q_{Y|X} \neq p_{Y|X}$.*

Proposition 3. *The greater the overlap between class supports S_y, the lower the ground-truth class accuracy of generated images from the `TAC-GAN` generator at a given noise rate $c \in [0, 1]$.*

Proofs are given in Appendix A. Proposition 1 gives the theoretical conditional density learned by G, and Proposition 2 demonstrates that `TAC-GAN` fails to be robust to label noise. The statement of Proposition 3 is intuitive, but it deserves mention since its proof follows naturally from our mathematical setup.

4 Proposed Methods

Proposition 2 suggests that all one needs to do is correct the `TAC-GAN` primary auxiliary classifier C so that it learns an unbiased distribution, $q_{Y|X} = p_{Y|X}$. To do this in the presence of noisy labels, we extend `TAC-GAN` by incorporating recent approaches for classification with noisy labels into C's training algorithm. We also aim to do so without giving the networks prior knowledge about the label noise structure. This leads to our proposals of combining `TAC-GAN` with the matrix-free methods of Co-Teaching [20] and MentorNet [22].

4.1 `TAC-GAN` with Classifier Co-Teaching (`CO-TAC`)

An additional classifier C', trained only on real data, is added. For every mini-batch of (real images, possibly corrupt labels), C and C''s training step follows the Co-Teaching algorithm. That is, C and C' each calculate a vector of losses for every training image in the mini-batch. C identifies some $R\%$ of images in the mini-batch which it has achieved smallest losses on, and C' only backpropagates its losses for those images. Likewise, C' chooses the losses for C to backpropagate against.

In the original formulation of Co-Teaching, the proportion of images R counted in the final loss for each batch decays linearly from 1 to $k(1-\epsilon)$, where k is a constant and ϵ is the noise rate. For simplicity we fix $k = 1$. However, we do not specify ϵ to the networks, as otherwise this violates the assumption-less condition that we seek to maintain regarding the noise structure. We therefore let R decay linearly to 0.5 for the first half training epochs, and after this, we maintain an exponential moving average E of the proportion of generated images correctly classified by C and C', updated every epoch. We interpret E as an estimate of ϵ, so we set $R = 1 - E$ in the latter half of training epochs.

The full training algorithm is described in Appendix B.2.

4.2 `TAC-GAN` with MentorNet (`TAC-MENTOR`)

A multilayer perceptron M called *MentorNet* is added. Input to M consists of C's loss and loss percentile on an image in the mini-batch, as well as the training epoch percentile. It outputs a weight between 0 and 1 based on the predicted importance of this particular image to C's learning. C calculates its loss based on the supplied weights for each image in the mini-batch.

To train M requires the existence of a small subset $\mathcal{D}$ in which the correctness of every image-label pair is known. This emulates a real-world setting where the label-correctness of a small proportion of samples is known, perhaps by human verification, but it is expensive to individually verify the remaining majority of labels. Because the label-correctness of samples in $\mathcal{D}$ is known, their optimal

weights are also known (1 for a correct label, 0 for an incorrect label). M can therefore be trained on images in $\mathcal{D}$ using a cross-entropy loss whenever they are sampled. We employ the SPADE algorithm (proposed alongside MentorNet) to concurrently train M alongside C.

In our experiments, we take $\mathcal{D}$ to be a random 5% of the whole training set. The full algorithm is described in Appendix B.3.

5 Experiments

5.1 Experimental Setups

We compare the performance of our proposed methods to baseline methods (`RCGAN` [16] and `rAC-GAN` [15]) over different datasets and label-noise settings. Since both of our proposed methods are built on `TAC-GAN`, we adapt the baseline methods to the `TAC-GAN` framework for fair comparison. We also take this opportunity to give the baseline methods new aliases to avoid confusion. Since `RCGAN` modifies **G**enerated labels using an estimated transition matrix immediately after being supplied to the generator, we use `RC-TAC-G` to denote the combination of `RCGAN` with `TAC-GAN`. On the other hand, `rAC-GAN` corrupts the outputs of the auxiliary **C**lassifier. We use `RC-TAC-C` to denote the combination of `rAC-GAN` with `TAC-GAN`. Hence we have four `TAC-GAN` frameworks to compare: `CO-TAC`, `TAC-MENTOR`, `RC-TAC-G` and `RC-TAC-C`.

Dataset	**No. classes**	**Images per class**	**Size**	**Label noise types**
MNIST	10	6000	28px	Uniform, Flip
CIFAR10	10	6000	32px	Uniform, Flip
CIFAR100	100	600	32px	Uniform, Cycle

Table 1: Datasets and experimental setups.

Table 1 outlines the datasets and tests to be performed. The datasets used are the MNIST dataset [24] and the CIFAR10/100 datasets [25]. Since clean labels are available by default, this allows us to experiment with different types and levels of label corruption. Table 2 gives brief descriptions of these.

Noise type	**Description**	**Values of c to test**
Uniform	Labels y corrupt to other labels $\tilde{y}$ with probability $\frac{c}{n-1}$ for each $\tilde{y} \neq y$.	0.20, 0.35, 0.50, 0.65, 0.80
Flip	Labels are split into pairs (y_i, y_j). Ground-truth labels y_i are flipped to y_j with probability c, and vice versa. This applies to all label pairs.	0.15, 0.25, 0.35, 0.45
Cycle	Only applicable to CIFAR100. Within each of the 10 superclasses, the 5 subclasses are placed randomly into a cycle, and labels of each subclass transition with probability c to the next label in the cycle.	0.15, 0.25, 0.35, 0.45

Table 2: Specifications of label noise settings.

For flipping noise in the MNIST dataset, the pairs established are $(1, 7), (2, 5), (3, 8), (4, 5), (6, 0)$. For CIFAR10, they are (airplane, bird), (automobile, truck), (cat, dog), (deer, horse), (frog, ship). These pairings are chosen to put together visually similar classes, making the task more challenging.

For performance evaluation, we will use a combination of qualitative appraisal and quantitative metrics. This approach is widely used in current GAN research since it remains open as to what makes a reliable, measurable quantity for GAN performance. For quantitative evaluation on MNIST, we use the GAN-train and GAN-test metrics [26]. GAN-train is the class accuracy of a pre-trained classifier for real data on the generated images, and GAN-test is the accuracy of the twin auxiliary classifier C_{mi} (which is trained only on generated images) on real images. For tests on CIFAR10 and CIFAR100, we additionally use Intra-FID [27] to measure intra-class image quality.

5.2 Results and Image Previews[1]

5.2.1 MNIST

MNIST	**Uniform**									
Corruption prob c	0.20		0.35		0.50		0.65		0.80	
GAN-test or -train (%)	test	train	test	train	test	train	test	train	test	train
RC-TAC-G	82.9	98.9	78.7	98.8	63.8	98.8	51.0	98.2	36.4	96.5
RC-TAC-C	89.9	97.6	81.1	98.8	73.2	97.4	89.3	97.8	68.5	95.9
CO-TAC	97.5	97.2	96.0	97.9	94.4	98.2	98.1	92.8	72.2	78.9
TAC-MENTOR	97.4	99.1	93.8	97.9	96.4	98.6	94.7	97.9	93.5	95.2

Table 3: MNIST uniform corruption GAN-test and GAN-train results. Higher score is better.

MNIST	**Flip**							
Corruption prob c	0.15		0.25		0.35		0.45	
GAN-test or -train (%)	test	train	test	train	test	train	test	train
RC-TAC-G	82.2	96.1	78.4	98.1	63.2	97.1	54.2	72.0
RC-TAC-C	89.3	98.3	73.5	70.1	79.1	78.9	47.5	52.5
CO-TAC	98.1	98.2	95.0	96.8	94.1	95.2	93.6	97.2
TAC-MENTOR	95.9	98.9	93.4	98.4	81.2	97.9	61.4	84.9

Table 4: MNIST label-flip corruption GAN-test and GAN-train results. Higher score is better.

[1] A shortage of time and computational resources did not allow enough CIFAR100 experiments to be run to completion before the deadline to permit substantial discussion. Deep Apologies.

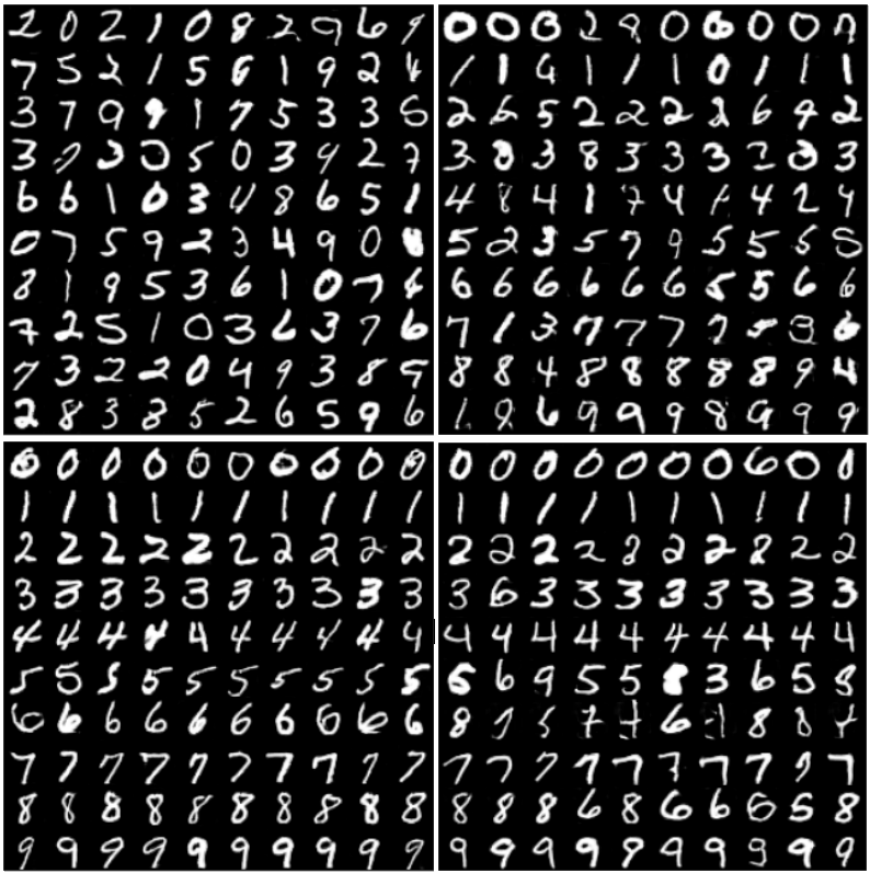

Figure 1: Generated image previews for MNIST with uniform corruption, where label corruption probability of training data = 0.8. Clockwise from top left: `RC-TAC-G`, `RC-TAC-C`, `CO-TAC`, `TAC-MENTOR`.

5.2.2 CIFAR10[2]

CIFAR10	**Uniform**									
Corruption prob c	0.20		0.35		0.50		0.65		0.80	
Intra-FID or GAN-train	I-FID	train	I-FID	train	I-FID	train	I-FID	train	I-FID	train
`RC-TAC-G`	19.8	82.5	36.3	80.1	73.0	69.7	71.7	58.2	93.7	64.3
`RC-TAC-C`	27.7	80.5	28.4	81.3	55.5	69.0	86.3	60.4	99.9	61.1
`CO-TAC`	24.4	81.9	29.4	82.2	39.1	80.6	59.3	79.0	66.8	73.1
`TAC-MENTOR`	20.6	81.3	30.1	78.5	54.5	75.6	97.9	71.6	96.2	65.3

Table 5: CIFAR10 uniform corruption Intra-FID and GAN-train results. Intra-FID: lower is better. GAN-train: measured in %, higher is better.

[2]GAN-test was not measured for CIFAR data due to an unfortunate last-minute technical bug.

CIFAR10	**Flip**							
Corruption prob c	0.15		0.25		0.35		0.45	
Intra-FID or GAN-train	I-FID	train	I-FID	train	I-FID	train	I-FID	train
`RC-TAC-G`	26.2	79.9	49.8	77.7	68.0	73.2	76.7	65.7
`RC-TAC-C`	23.1	81.2	41.6	66.1	87.2	48.4	108.2	40.9
`CO-TAC`	24.2	83.6	31.6	79.7	64.5	80.0	87.0	74.8
`TAC-MENTOR`	21.0	78.1	32.0	70.9	57.9	67.8	79.1	63.3

Table 6: CIFAR10 label-flip corruption Intra-FID and GAN-train results. Intra-FID: lower is better. GAN-train: measured in %, higher is better.

Figure 2: Generated image previews for CIFAR10 with label-flipping corruption, corruption probability 0.45. Clockwise from top left: `RC-TAC-G`, `RC-TAC-U`, `CO-TAC`, `TAC-MENTOR`.

5.3 Discussion

We will now discuss the results presented in Section 5.2.

5.3.1 MNIST

The class accuracy of generated images, measured by GAN-test, generally deteriorates with increasing label corruption probability as expected. However, it is more serious for the baseline models `RC-TAC-G` and `RC-TAC-C`, whereas the proposed solutions appear to provide more insulation against label noise, from a class accuracy perspective. On the other hand, the GAN-train statistic tends to be less susceptible to increasing label corruption probability. This suggests that while the auxiliary classifier is mostly successful at extracting ground-truth labels despite being trained with label noise, the baseline methods have difficulties translating this to good generator performance. This is contrary to the observation drawn from Proposition 2 that correcting the primary auxiliary classifier to learn an unbiased distribution of labels naturally fixes the generator as well, and deserves further investigation.

The proposed methods, while performing better than the baselines, seem not to give consistent performance - refer to the abrupt deterioration in performance for `CO-TAC` and `TAC-MENTOR` in the uniform and flipping scenarios respectively. However, when this happens, *both* GAN-test and GAN-train decline, whereas the latter was (mostly) intact when the baseline methods' performances deteriorated. One could therefore suggest that the proposed methods provide more consistent 'bridging' between classifier and generator performance. If the relationship is causal, further solutions for classification with noisy labels could be implemented in one of these frameworks to drive up generator performance too.

A qualitative evaluation of generated images largely reflects what the quantitative results suggest: the baselines suffer in terms of class accuracy, and image quality appears to suffer slightly in the `RC-TAC-C` case as well. The proposed methods improve greatly in terms of accuracy, particularly MentorNet. The samples shown on Figure 1 are the image previews corresponding to the rightmost column of Table 3. Here, `CO-TAC` fell vulnerable to class inaccuracy and/or poor image quality for certain classes. This reflects the issue of performance consistency raised above.

5.3.2 CIFAR10

Intra-FID, which measures intra-class image quality and diversity, deteriorates (increases) as the corruption probability increases. This is expected since training label corruption interferes with learning progress, ultimately leading to poorer quality generated images for a fixed number of training epochs. In the uniform corruption experiments with high corruption probability, `CO-TAC` performed much better than the other networks with respect to both metrics (intra-FID and GAN-train). `TAC-MENTOR`, on the other hand, did not record a substantial improvement on the baselines.

In the flip corruption experiments, there appears to be no clear winner; `RC-TAC-G`, `CO-TAC` and `TAC-MENTOR` outperform one another with respect to different evaluation metrics over different label corruption probabilities. `RC-TAC-C` was the underperformer in this series of experiments.

A qualitative analysis provides some insight beyond these seemingly inconclusive results. Over a range of configurations for label corruption type and corruption probability (one of whose image previews are shown in Figure 2), `RC-TAC-C` experiences an onset of mode collapse early enough into training, resulting in generated images of subpar quality. In several situations, `CO-TAC` also

experienced mode collapse, which can be seen in the 'dog' class in Figure 2. However, in classes where mode collapse did not occur, `CO-TAC` produced images of good quality, diversity and realism compared to the other models. How one can tailor `CO-TAC` to avoid mode collapse is therefore a question worth considering. `RC-TAC-G` and `TAC-MENTOR` achieve good image diversity, but on closer inspection they result in poor class accuracy. That said, `TAC-MENTOR` produces more well-defined images with greater integrity than those generated by `RC-TAC-G`.

6 Conclusion and Future Work

In this work, we developed and tested two label-noise robust extensions to `TAC-GAN` in view of a preliminary theoretical discussion. These extensions are based on relatively new approaches to classification with noisy training labels, and make no assumptions on the label noise structure. Based on our experiments, the new models perform at least as well as the baselines, and in some cases there is significant outperformance.

Future work related to the ideas discussed in this work may address the following questions:

- *How do auxiliary classifiers and generators interact with each other in an auxiliary classifier GAN?* This could enable us to understand which solutions for classification problems can be ported into GANs to improve the quality of a generator. One may even explore whether generators can be used to improve classification outcomes, since (class-conditional) generation and classification are essentially inverse problems.
- *Are there more compact ways to enhance GAN capabilities rather than introducing new networks into the ecosystem?* A basic GAN has two networks. An auxiliary classifier is used to add class-conditional generation capabilities. A twin auxiliary classifier is used to facilitate image diversity. In this work, we added yet another network to handle label noise. Having more networks makes GANs more capable yet more complex and intractable; could simpler methods suffice?
- *What are the inherent properties of GANs?* Deep networks are commonly referred as 'black boxes', and this label is true to GANs in particular. An understanding of the theoretical and/or mathematical properties of GANs would help to illuminate this black box, and in doing so potentially expose a range of new methods and techniques to solve harder problems.

7 Acknowledgements

This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne [28]. This Facility was established with the assistance of LIEF Grant LE170100200.

I would like to extend my gratitude to Dr. Mingming Gong, who provided generous and helpful supervision over the course of this project. This work would not have been possible without him.

Appendix A Mathematical Work

A.1 Proof of Proposition 1

[18] shows that the primary auxiliary classifier C successfully learns a conditional distribution $q_{Y|X}$ such that $q_{Y|X} \approx p_{Y|X}$, the true conditional distribution of true labels given images. The generator induces a conditional density $q_{X|Y}$ and hence q_X, since q_Y is given (the label is manually generated and supplied). It can be proved that the GAN minimax game leads to $q_X \approx p_X :=$ the true density over images. Assume $Y \sim DU(1, n)$, so p_Y is known. Then

$$q_{X|Y} = \frac{q_{Y|X} q_X}{p_Y}$$

is the conditional density induced by `TAC-GAN`. We have that

$$q_{X|Y} = \frac{q_{Y|X} q_X}{p_Y} \approx \frac{p_{Y|X} p_X}{p_Y} =: p_{X|Y}, \tag{1}$$

thus fulfilling our goal to learn $p_{X|Y}$.

Now factor in label noise. Let $\tilde{Y}$ the distribution over corrupt labels defined by

$$\mathbb{P}\left(\tilde{Y} = j \mid Y = i\right) = \Gamma_{ij} \mathbb{P}\left(Y = i\right)$$

where Γ is the corruption matrix.

With label noise, the primary auxiliary classifier in `TAC-GAN` learns $q_{\tilde{Y}|X} \approx p_{\tilde{Y}|X}$, but *thinks* it is learning $q_{Y|X} \approx p_{Y|X}$. This is because labels are now drawn from $\tilde{Y}$, not Y. This 'mistake' causes `TAC-GAN` to use the conditional density

$$q_{X|Y} := \frac{q_{\tilde{Y}|X} q_X}{p_Y} \tag{2}$$

as an estimator for $p_{X|Y}$. Proposition 1 illuminates the consequences of this. We proceed with its proof.

Proposition 1. *Suppose the label-noise matrix $\Gamma_{n \times n}$ represents uniform corruption with probability c, i.e. $\Gamma_{ii} = 1 - c$ and $\Gamma_{ij} = \frac{c}{n-1}$ for $i \neq j$. Then the induced conditional density $q_{X|Y}$ is approximately given by*

$$q_{X|Y}(x \mid k) \approx (1 - c) p_{X|Y}(x \mid k) + \frac{c}{n-1} \cdot \frac{p_X(x)}{p_Y(k)} \sum_{j=1, j \neq k}^{n} p_{Y|X}(j \mid x).$$

Proof. For any class label k,

$$
\begin{aligned}
q_{X|Y}(x \mid k) &= \frac{q_{\tilde{Y}|X}(k \mid x) q_X(x)}{p_Y(k)} && \text{using (2)} \\
&\approx \frac{p_{\tilde{Y}|X}(k \mid x) p_X(x)}{p_Y(k)} && \text{as } q_{\tilde{Y}|X} \approx p_{\tilde{Y}|X}, q_X \approx p_X \\
&= \frac{p_X(x)}{p_Y(k)} \cdot \sum_{j=1}^{n} \left(\underbrace{p_{\tilde{Y}|(X,Y)}(k \mid (x,j))}_{=p_{\tilde{Y}|Y}(k|j)=\Gamma_{jk}} p_{Y|X}(j \mid x) \right) \\
&= \frac{p_X(x)}{p_Y(k)} \cdot \sum_{j=1}^{n} \left(\Gamma_{jk} p_{Y|X}(j \mid x) \right) \\
&= (1-c) \frac{p_X(x)}{p_Y(k)} \cdot p_{Y|X}(k \mid x) + \frac{c}{n-1} \cdot \frac{p_X(x)}{p_Y(k)} \cdot \sum_{j=1, j \neq k}^{n} p_{Y|X}(j \mid x) \\
&= (1-c) p_{X|Y}(x \mid k) + \frac{c}{n-1} \cdot \frac{p_X(x)}{p_Y(k)} \sum_{j=1, j \neq k}^{n} p_{Y|X}(j \mid x).
\end{aligned}
$$

□

A.2 Proof of Proposition 2

Proposition 2. *In the uniform label-noise setting with n classes and a positive probability of label corruption, the `TAC-GAN` generator is not optimal (in the sense of Definition 1). Moreover, this is caused by the primary auxiliary classifier learning a biased distribution for labels given images, $q_{Y|X} \neq p_{Y|X}$.*

Proof. Suppose for a contradiction that the generator is optimal. Then for an image x and label k such that $p_{Y|X}(k \mid x) = \max_j p_{Y|X}(j \mid x)$, the induced conditional density $q_{X|Y}$ satisfies $q_{X|Y}(x \mid k) = p_{X|Y}(x \mid k)$. By Proposition 1 this holds iff

$$
\begin{aligned}
\frac{c}{n-1} \cdot \frac{p_X(x)}{p_Y(k)} \sum_{j=1, j \neq k}^{n} p_{Y|X}(j \mid x) &= c p_{X|Y}(x \mid k) \\
p_X(x) \sum_{j=1, j \neq k}^{n} p_{Y|X}(j \mid x) &= (n-1) p_{X|Y}(x \mid k) p_Y(k) \\
&= (n-1) p_{Y|X}(k \mid x) p_X(x) \\
\sum_{j=1, j \neq k}^{n} p_{Y|X}(j \mid x) &= (n-1) p_{Y|X}(k \mid x) \\
1 - p_{Y|X}(k \mid x) &= (n-1) p_{Y|X}(k \mid x) \\
p_{Y|X}(k \mid x) &= 1/n.
\end{aligned}
$$

But this implies that

$$\sum_{j=1, j\neq k}^{n} p_{Y|X}(j \mid x) = \frac{n-1}{n}$$
$$\Rightarrow \exists j' \neq k,\ p_{Y|X}(j' \mid x) \geq 1/n = p_{Y|X}(k \mid x), \qquad n-1 \text{ terms in sum, pigeonhole principle}$$

contradicting the initial condition that $p_{Y|X}(k \mid x) = \max_j p_{Y|X}(j \mid x)$.

If the `TAC-GAN` primary auxiliary classifier had instead learned an unbiased conditional distribution $q_{Y|X} = p_{Y|X}$, then (1) holds and gives $q_{X|Y} = p_{X|Y}$. This concludes the proof. □

A.3 Proof of Proposition 3

Proposition 3. *The greater the overlap between class supports S_y, the lower the ground-truth class accuracy of generated images from the `TAC-GAN` generator at a given noise rate $c \in [0, 1]$.*

Proof. We prove the result for uniform noise among n classes with corruption probability c. Let S be the union of all class supports. If the generator is supplied with class label k, we measure the probability of generating an image of the correct ground-truth class by the quantity $\int_S p_{Y|X}(k \mid x) q_{X|Y}(x \mid k) dx$. By Proposition 1,

$$\begin{aligned}
\int_S p_{Y|X}(k \mid x) q_{X|Y}(x \mid k) dx &\approx (1-c) \underbrace{\int_S p_{Y|X}(k \mid x) p_{X|Y}(x \mid k) dx}_{=:\alpha_k} \\
&\quad + \frac{c}{n-1} \cdot \frac{p_X(x) p_{Y|X}(k \mid x)}{p_Y(k)} \int_S \sum_{j=1, j\neq k}^{n} p_{Y|X}(j \mid x) dx \\
&= (1-c)\alpha_k + \frac{c}{n-1} \cdot p_{X|Y}(x \mid k) \int_S 1 - p_{Y|X}(k \mid x) dx \\
&= (1-c)\alpha_k + \frac{c}{n-1} \left[\int_S p_{X|Y}(x \mid k) dx - \alpha_k \right] \\
&= (1-c)\alpha_k + \frac{(1-\alpha_k)c}{n-1}.
\end{aligned}$$

This is decreasing in $\alpha_k := \int_S p_{Y|X}(k \mid x) p_{X|Y}(x \mid k) dx$, which is a measure for the degree of overlap between the class support S_k and the other class supports. □

To make the meaning of α_k clear, suppose that there is no overlap, so S_k is disjoint from all other class supports. If $p_{X|Y}(x \mid k) > 0$ for some $x \in S$, it must hold with probability one that $x \in S_k$, so $p_{Y|X}(k \mid x) = 1$. Thus $\alpha_k = \int_S p_{X|Y}(x \mid k) dx = 1$. On the other hand, suppose that all class labels in the training set were random and so the S_j's and all class-specific distributions are identical. Then $p_{Y|X}(k \mid x) = 1/n$, and $\alpha_k = 1/n$.

α_k is a rather crude measure of class support overlaps, because it does not contain information about *how* the class supports overlap, and its value is almost always unknown. Nevertheless, Proposition 3 theoretically backs the observation that class support overlaps adversely affect the class accuracy of conditional GANs.

Appendix B Algorithms

B.1 Common elements

The following are common to all configurations:

Symbol	Description
G, Θ_G	Conditional generator with parameters
D	Discriminator
C	Primary auxiliary classifier
Θ_{DC}	Parameters of joint network consisting of D, C and C_{mi} (as per original `TAC-GAN`)
$\mathcal{S}$	Training dataset $\{(\text{image}_i, \text{label}_i)\}$
n	Number of classes
Γ	$n \times n$ transition matrix, Γ_{ij} = prob. of corrupting image with label i to label j. $\Gamma_{i\cdot} = i^{th}$ row of Γ
T_{train}	Number of training epochs
m	Mini-batch size
d	Noise dimension
η	Learning rate
$\ell(\cdot, \cdot)$	Cross-entropy loss

Table B1: Common elements of all configurations.

Algorithm for training G

The training procedure for G does not change across the configurations, so we state it here first.

Algorithm 1 `TAC-GAN`: G training step

Input noise $\mathbf{z} \in \mathbb{R}^{m \times d}$, labels $\mathbf{y} \in \{1, \ldots, n\}^m$

1: $\mathcal{L}_{gan} \leftarrow \ell(D(G(\mathbf{z}, \mathbf{w})), \mathbf{1})$ ▷ D estimates prob. of image being real
2: $\mathcal{L}_c \leftarrow \ell(C(G(\mathbf{z}, \mathbf{w})), \mathbf{w})$
3: $\mathcal{L}_{cmi} \leftarrow \ell(C_{mi}(G(\mathbf{z}, \mathbf{w})), \mathbf{w})$
4: $\mathcal{L}_G \leftarrow \mathcal{L}_{gan} + \mathcal{L}_c - \mathcal{L}_{cmi}$ ▷ $-\mathcal{L}_{cmi}$ because G is competing with C_{mi}
5: **update** $\Theta_G = \Theta_G - \eta \nabla \mathcal{L}_G$

B.2 TAC-GAN with Classifier Co-Teaching (CO-TAC)

Here we add another auxiliary classifier C', trained on real data. Its parameters $\Theta_{C'}$ are not shared with Θ_{DC}.

Algorithm 2 CO-TAC

for $T = 1, \ldots, T_{train}$ **do**
  if $T < T_{train}/2$ **then**
    $R(T) \leftarrow 1 - T/T_{train}$
  else
    $R(T) \leftarrow 1 - E$
  end if
  $r \leftarrow \lfloor mR(T) \rfloor$ ▷ no. small losses to choose per mini-batch

  for mini-batch $(\mathbf{x}, \mathbf{y}) := \{(x_i, y_i)\}_{i \leq m}$ in $\mathcal{S}$ **do**
    sample corrupted labels $\tilde{\mathbf{y}} \in \{1, \ldots, n\}^m$ by $\tilde{\mathbf{y}}_i \sim \Gamma_{y_i \cdot}$
    sample noise $\mathbf{z} \in \mathbb{R}^{m \times d}$ by $\mathbf{z}_{i \cdot} \sim N(\mathbf{0}, I_d)$
    sample generated labels $\mathbf{w} \in \{1, \ldots, n\}^m$ by $\mathbf{w}_i \sim Unif(\{1, \ldots, n\})$

    train D, C, C', C_{mi}**:**
      $\mathcal{L}_{Dgan} \leftarrow \ell(D(\mathbf{x}), \mathbf{1}) + \ell(D(G(\mathbf{z}, \mathbf{w})), \mathbf{0})$
      $\mathcal{L}_{Cr} \leftarrow \ell(C(\mathbf{x}), \tilde{\mathbf{y}})$, no reduction
      $\mathcal{L}_{C'r} \leftarrow \ell(C'(\mathbf{x}), \tilde{\mathbf{y}})$, no reduction
      $J \leftarrow \arg\min_{\{H \subseteq [m] : |H| = r\}} \sum_{h \in H} \mathcal{L}_{Cr}[h]$ ▷ select r smallest losses
      $K \leftarrow \arg\min_{\{H \subseteq [m] : |H| = r\}} \sum_{h \in H} \mathcal{L}_{C'r}[h]$
      $\mathcal{L}_{Cr} \leftarrow \frac{1}{r} \sum_{k \in K} \mathcal{L}_{Cr}[k]$ ▷ backprop losses chosen by other classifier
      $\mathcal{L}_{C'r} \leftarrow \frac{1}{r} \sum_{j \in J} \mathcal{L}_{C'r}[j]$
      $\mathcal{L}_{mi} \leftarrow \ell(C_{mi}(G(\mathbf{z}, \mathbf{w})), \mathbf{w})$
      $\mathcal{L}_{DC} \leftarrow \mathcal{L}_{Dgan} + \mathcal{L}_{Cr} + \mathcal{L}_{mi}$ ▷ as per TAC-GAN classifier/discriminator loss
      update $\Theta_{DC} = \Theta_{DC} - \eta \nabla \mathcal{L}_{DC}$
      update $\Theta_{C'} = \Theta_{C'} - \eta \nabla \mathcal{L}_{C'r}$
    end train D, C, C', C_{mi}

    train G**:** input $\mathbf{z}, \mathbf{w}$ (see **Algorithm 1**)
    end train G
  end for
  $E \leftarrow$ update exp. moving average with prop. of generated images classified correctly by C this epoch
end for
Output Θ_G

B.3 TAC-GAN with MentorNet (TAC-MENTOR)

Table B2 summarises the additional elements of TAC-MENTOR.

Symbol	Description
M	MentorNet (multilayer perceptron)
T_{burn}	Number of burn-in epochs for MentorNet, during which weights are sampled from Bernoulli(p)
$\mathcal{S}^*$	Small subset of $\mathcal{S}$; images whose true labels are known to MentorNet. We take $\lvert\mathcal{S}^*\rvert = 0.05\lvert\mathcal{S}\rvert$

Table B2: Additional elements for TAC-MENTOR.

Algorithm 3 TAC-MENTOR

for $T = 1, \ldots, T_{train}$ **do**
  for mini-batch $(\mathbf{x}, \mathbf{y}) := \{(x_i, y_i)\}_{i \leq m}$ in $\mathcal{S}$ **do**
    sample corrupted labels $\tilde{\mathbf{y}} \in \{1, \ldots, n\}^m$ by $\tilde{y}_i \sim \Gamma_{y_i \cdot}$
    sample noise $\mathbf{z} \in \mathbb{R}^{m \times d}$ by $\mathbf{z}_{i \cdot} \sim N(\mathbf{0}, I_d)$
    sample generated labels $\mathbf{w} \in \{1, \ldots, n\}^m$ by $w_i \sim Unif(\{1, \ldots, n\})$

    train D, C, C_{mi}**, possibly** M**:**
      $\mathcal{L}_{Dgan} \leftarrow \ell(D(\mathbf{x}), \mathbf{1}) + \ell(D(G(\mathbf{z}, \mathbf{w})), \mathbf{0})$
      $\mathcal{L}_{Cr} \leftarrow \ell(C(\mathbf{x}), \tilde{\mathbf{y}})$, no reduction
      $\ell_{EMA} \leftarrow$ update exp. moving average with 75-pctile loss in $\mathcal{L}_{Cr}$
      if $T \leq T_{burn}$ **then**
        sample backprop weights $\boldsymbol{\lambda} \in \{0, 1\}^m$ by $\lambda_i \sim$ Bernoulli(p)
      else
        $\boldsymbol{\lambda} \leftarrow M(\phi(\mathcal{L}_{Cr}, \ell_{EMA})) := [M(\phi([\mathcal{L}_{Cr}]_i, \ell_{EMA}))]_{i \leq m}$ ▷ M decides weights using loss, loss EMA

        train M**:**
          for all i such that $(x_i, y_i) \in (\mathbf{x}, \mathbf{y}) \cap \mathcal{S}^*$ **do**
            $\mathcal{L}_M \leftarrow \ell(\lambda_i, \mathbf{1}(\tilde{y}_i = y_i))$ ▷ optimal weight 1 if label is correct, 0 else
            update $\Theta_M = \Theta_M - \eta \nabla \mathcal{L}_M$
          end for
        end train M

      end if
      $\mathcal{L}_{Cr} \leftarrow \frac{1}{m} \boldsymbol{\lambda}^T \mathcal{L}_{Cr}$
      $\mathcal{L}_{mi} \leftarrow \ell(C_{mi}(G(\mathbf{z}, \mathbf{w})), \mathbf{w})$
      $\mathcal{L}_{DC} \leftarrow \mathcal{L}_{Dgan} + \mathcal{L}_{Cr} + \mathcal{L}_{mi}$
      update $\Theta_{DC} = \Theta_{DC} - \eta \nabla \mathcal{L}_{DC}$
    end train D, C, C_{mi}**, possibly** M

    train G**:** input $\mathbf{z}, \mathbf{w}$ (see **Algorithm 1**)
    end train G
  end for

end for
Output Θ_G

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, 2014.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks, 2016.

[3] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[4] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. Geometry-Consistent Generative Adversarial Networks for One-Sided Unsupervised Domain Mapping, 2018.

[5] Matthew Amodio and Smita Krishnaswamy. TraVeLGAN: Image-to-image Translation by Transformation Vector Learning. *CoRR*, abs/1902.09631, 2019.

[6] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long Text Generation via Adversarial Training with Leaked Information, 2017.

[7] William Fedus, Ian Goodfellow, and Andrew Dai. MaskGAN: Better Text Generation via Filling in the ____. 2018.

[8] Veit Sandfort, Ke Yan, Perry J. Pickhardt, and Ronald M. Summers. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Scientific Reports*, 9, 2019.

[9] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets, 2014.

[10] Dana Angluin and Philip Laird. Learning From Noisy Examples. *Mach. Learn.*, 2(4):343–370, April 1988.

[11] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

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

[13] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are Anchor Points Really Indispensable in Label-Noise Learning?, 2019.

[14] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. NLNL: Negative Learning for Noisy Labels. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019.

[15] Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. Label-Noise Robust Generative Adversarial Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[16] Kiran Koshy Thekumparampil, Ashish Khetan, Zinan Lin, and Sewoong Oh. Robustness of Conditional GANs to Noisy Labels, 2018.

[17] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis With Auxiliary Classifier GANs, 2016.

[18] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin Auxiliary Classifiers GAN, 2019.

[19] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep Self-Learning From Noisy Labels. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019.

[20] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels, 2018.

[21] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2691–2699, 2015.

[22] 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, 2017.

[23] Yanwu Xu, Mingming Gong, Junxiang Chen, Tongliang Liu, Kun Zhang, and Kayhan Batmanghelich. Generative-Discriminative Complementary Learning, 2019.

[24] Yann LeCun and Corinna Cortes. MNIST handwritten digit database, 2010.

[25] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Canadian Institute For Advanced Research, 2009.

[26] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my GAN?, 2018.

[27] Takeru Miyato and Masanori Koyama. cGANs with Projection Discriminator. In *International Conference on Learning Representations*, 2018.

[28] Linh Vu Lev Lafayette, Greg Sauter and Bernard Meade. Spartan Performance and Flexibility: An HPC-Cloud Chimera, 2017.