ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200228

Negative-Aware Training: Be Aware of Negative Samples

Xin Li[1] and **Xiaodong Jia** [†1] and **Xiao-Yuan Jing** [†123]

Abstract. Negative samples, whose class labels are not included in training sets, are commonly classified into random classes with high confidence and this severely limits the applications of traditional models. To solve this problem, we propose an approach called Negative-Aware Training (NAT), which introduces negative samples and trains them along with the original training set. The object function of NAT forces the classifier to output equal probability for each class on negative samples, other settings stay unchanged. Moreover, we introduce NAT into GAN and propose NAT-GAN, in which discriminator distinguishes between both generated samples and negative samples. With the assist of NAT, NAT-GAN can find more accurate decision boundaries, thus converges steadier and faster. Experimental results on synthesis and real-word datasets demonstrate that: 1) NAT gets better performance on negative samples in accordance with our proposed negative confidence rate metric. 2) NAT-GAN gets better quality scores than several traditional GANs and achieves state-of-the-art Inception Score (9.2) on CIFAR 10. Our demo and code are available at https://natpaper.github.io.

1 Introduction

Deep neural network has shown dramatic performance in various tasks [12, 18, 20, 5]. Despite its high performance, deep neural network is still delicate in dealing with negative samples. Negative samples – classes of which are not included in training set (we refer samples in training set as positive samples) – are commonly predicted to random classes with high confidence. This phenomenon frequently occurs in real-world applications, which becauses training set cannot always contain all the classes in the real environment. As shown in Figure 1 (right), well-trained networks are expected to output uniform probability distribution on samples with unknown classes. In practice, however, their outputs are often like what shown in Figure 1 (left). Unfortunately, this can be a serious problem in some real-world applications. Take self-driving cars as an example, it may cause a terrible accident if the classifier predicts an object to random label with high confidence, however unknown is common as the classifier doesn't know all exist classes. This problem can never be solved by adding more classes into the training set because the black swan event always happens. We surprisingly found that a network trained on CIFAR 10 with 95.54% accuracy predicts 97.1% samples from CIFAR 100 over 0.4 confidence, and 55.8% over 0.9 confidence. What's more astonishing is that this even happened on random noises.

[1] School of Computer Science, Wuhan University, China. Email: lixincs@whu.edu.cn, jxdshimon@gmail.com, Jingxy_2000@126.com.
[†] Corresponding author.
[2] School of Automation, Nanjing University of Posts and Telecommunications, China.
[3] School of Computer, Guangdong University of Petrochemical Technology, China.

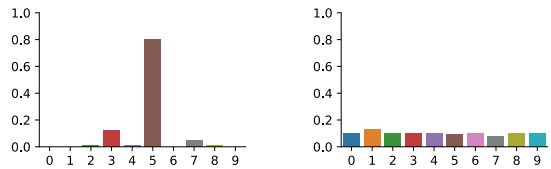

Figure 1: Network predicts random labels (left) as it is unaware of negative samples, but with negative-aware training, the network predicts uniform distribution (right) which indicates the class is not known.

Formally, let X be the domain of positive samples in training set and Ω be the domain of all the positive samples and negative samples. The domain of negative samples can be written as $\Psi = \Omega - X$. The classifier $\mathcal{F}$ is trained to map X to χ, which is the target labels of X. If $\mathcal{F}$ is unaware of the negative domain, it will map $\Omega = X + \Psi$ to χ, but $\mathcal{F}$ should map Ψ to ψ as we expected, where ψ is the negative prediction, the information of Ψ should be taken into consideration.

Taking both the positive and negative samples into consideration is similar to discriminator of Generative adversarial nets (GANs) [4], which seek to distinguish real (positive) samples from fake (negative) samples. GANs have shown promising results in various challenging tasks, such as realistic image generation [15, 23], conditional image generation [7, 8], and image manipulation [25]. Some GANs exploit label information and get more inspiring results [10, 19, 16, 24]. There are various works aiming at improving the performance of GAN, such as Wasserstein distance [1], spectral normalization [13], large batchsize [2], and evolution [21].

The key to address the random prediction problem stated above lies in how to use negative samples. Treating negative samples as a new class is one simple approach. GANs such as AM-GAN [24] and SGAN [19] train discriminators in this way, but it results in a bad discriminator and brings new problem for supervised learning – unbalanced classification, which caused by the large mount of negative samples, and thus hurts the performance. CatGAN [19] optimizes the entropy of all classes, which is similar to our proposed NAT-GAN, but NAT-GAN not only steps further but also is simpler. We argue that entropy constraint actually alleviates the unbalancing problem. Our approach sets clear targets for both negative and positive samples, and adds external negative samples to assist the discriminator, which will guide the network to learn X and Ψ well and help the generator converge better.

The contributions are summarized as follows:

1. We propose a training approach called Negative-Aware Training (NAT), which introduces negative samples into supervised learning and forces classifiers to output equal probability on negative samples. With NAT, the problem of classifier predicts random la-

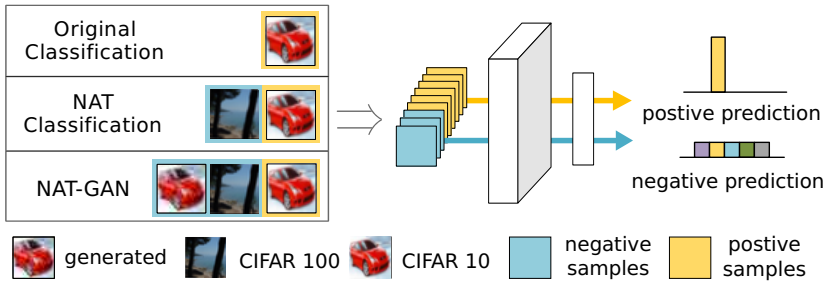

Figure 2: Negative-Aware Training model. It makes a certain prediction on positive samples (yellow) and leads to peaked conditional class distribution, uncertain prediction on negative samples (blue) and leads to a uniform distribution.

bels with high confidence on negative samples can be addressed. Moreover, we propose a metric called Negative Confidence Rate to evaluate the classifiers' performance on negative samples (Section 2.1).

2. By introducing external negative samples into GAN, we propose NAT-GAN. The negative information helps the discriminator find the decision boundary easily and allows the generator to converge faster and steadier. Moreover, with the help of negative samples, the distribution generated by NAT-GAN is better than that generated by the original GAN, thus improving the qualities of generated samples (Section 2.2).
3. Extensive experimental results on synthesis and real-word datasets not only demonstrate the generalization and superiority of NAT, but also indicate that NAT-GAN converges steadier and faster, outperforms several competing GANs, achieves state-of-the-art Inception Score (9.2) on CIFAR 10 (Section 3).

2 Methodology

2.1 NAT on supervised classification

The output probability distributions—as we focus on classification task and softmax is commonly used to produce probability for each class—are the network confidences on specific data. The network is forced to produce correct labels with probability 1 on positive samples, and produce equal probability on negative samples, whose confidences of all classes are $1/c$, where c is the number of training classes. $1/c$ is the lowest bound of probability prediction can be reached and also is the ground truth of negative samples on supervised classification. It is highly intuitive and the distributions of positive and negative are more separate. From the perspective of information theory, we minimize the entropy of positive prediction and maximize the entropy of negative prediction. The approach we proposed is called Negative-Aware Training (NAT) and is illustrated in Figure 2.

Formally, the original network classifies the negative samples Ψ as ordinary data as it is unaware of negative samples Ψ, which means in training we learn $\mathcal{F}(X) \rightarrow x$, but in practice, we do $\mathcal{F}(\Omega)$ and that results in $\mathcal{F}(\Psi) \rightarrow x$. Our approach involves negative samples during training and forces network to produce $1/c$ for each class and that gives rise to negative-aware network. The information of negative samples Ψ and positive samples X are known, the domain of negative samples Ψ is learned and such that $\mathcal{F}$ projects Ψ to ψ, not to χ.

Original training settings and cost function stay unchanged, which means we don't need to change the network architecture and hyperparameters. Both positive samples and negative samples are jointly trained but with different strategies. Classification cost function, like cross entropy, stays unchanged while cost function on negative samples, e.g., KL(Kullback-Leibler) divergence, forces the network to produce $1/c$ for each label. Thus, the overall cost function can be written as:

$$J_{NAT} = \frac{1}{n}\sum_{i=1}^{n}\left(L_{pos}\mathbb{I}_{pos} + D(\hat{y}_{neg} \parallel y_{neg})\mathbb{I}_{neg}\right), \tag{1}$$

where neg and pos indicate negative and positive samples. L_{pos} is the original loss function for positive samples, $\hat{y}_{neg}$ is the output probability on negative samples, y_{neg} is ground truth:

$$\hat{y}_{neg(i)} = \frac{\exp(o_i)}{\sum_{i=1}^{n}\exp(o_i)}, \tag{2}$$

$$y_{neg} = \left(\frac{1}{c}, \ldots, \frac{1}{c}\right)^T, \tag{3}$$

where c is the number of classes, o is the network output. D is the distribution distance metric which measures the similarity of output distribution $\hat{y}_{neg}$ and y_{neg}, and can be customized for specific tasks. KL divergence is used in our work:

$$\mathrm{KL}(y_{neg}\|\hat{y}_{neg}) = \sum_{i=1}^{c} y_{neg(i)} \log \frac{y_{neg(i)}}{\hat{y}_{neg(i)}}. \tag{4}$$

One may ask: can similar classes be used as negative? We argue that, if one evaluates the classes are similar and expects the classifier to output information about it, then that class should NOT be used. For example, if one views cat and leopard are different classes, then the negative relation is established, otherwise, cat and leopard are not negatives relation if one expects the classifier outputs similar distributions. The negative relation is judged by human.

Evaluation Metric. In order to measure the classifier's performance on negative samples, we define Negative Confidence Rate (NCR) as follows:

$$NCR_t = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(max(\hat{y}_{neg}) > t), \tag{5}$$

where n is the number of negative samples and t is the threshold. NCR_t indicates the ratio of maximum confidence over threshold t, bigger NCR_t means poorer performance on negative samples.

2.2 NAT on GAN

Training categorical GAN is much similar to classification with NAT, generated samples are negative Ψ and real data is X, but discriminator still meets two problems. First, generated samples distribution is not ideally covered Ψ as we expected, because the generator

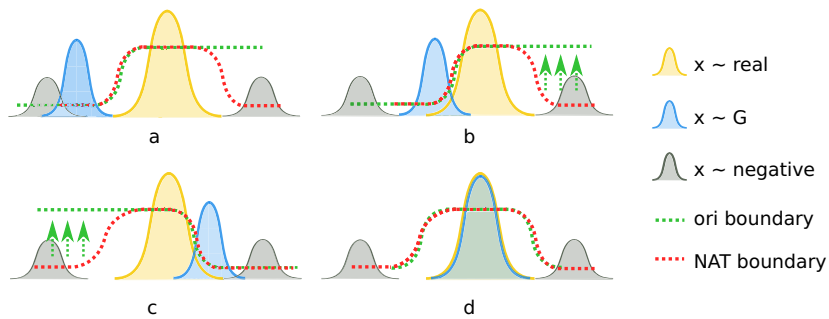

Figure 3: NAT-GAN generates robust decision boundary compared with original GAN. Original GAN is learning the boundary between generated and real distributions at the beginning of adversarial procedure (a, green), what NAT does is provide prior knowledge of real boundary (a, red). The generator is better (b) but the original boundary is poor as the distribution on the right, where located another negative distribution, is still encouraged. That may provide temporary wrong gradient information and results in (c), the generated distribution shifting to the right side and the boundary changed dramatically. NAT, however, is more robust with the help of negative information. In the end, generated distribution matches the real distribution well (d), but NAT assists with a steady adversarial procedure, and GAN converges faster and more accurate. This experiment is displayed in Figure 5.

is learned from discriminator and aims to fool discriminator, which means negative samples can still fake discriminator in practice. Second, generated distribution is shifting during learning and thus the classification boundary is changing with regard to generator, which results in poor discriminator and poor gradient information (shown in Figure 3).

Generator's goal is to minimize the distance of real and fake, here we use $\mathrm{KL}(p_{data}\|p_G)$. Since we introduce negative samples and that leads to conditional probability, distance between p_{data} and $p_{G|neg}$ is smaller than p_{data} and p_G, then:

$$\begin{aligned}&\mathrm{KL}(p_{data}\|p_{G|neg})\\&= p_{data}\log p_{data} - p_{data}\log p_{G|neg}\\&\leq p_{data}\log p_{data} - p_{data}\log p_G\\&= \mathrm{KL}(p_{data}\|p_G).\end{aligned} \tag{6}$$

AM-GAN introduces an additional fake class, which leads to the discriminator suffers from the unbalanced classification problem. We propose NAT-GAN, based on AM-GAN and CatGAN, and removed the additional fake class, target probability distribution of negative samples is set explicitly, which is $1/c$ for each class. Note that generated samples are negative as well. In NAT-GAN, external negative samples are added to train discriminator and further improving the performance of generator. The main difference between CatGAN and NAT-GAN is the discriminator, we explicitly add external negative samples to train discriminator, meanwhile, the generator stays unchanged (see Figure 2).

Generator loss function of AM-GAN and NAT-GAN is:

$$L_G = \mathbb{E}_{(x,y)\sim G}[H(\Upsilon(y), D(x))], \tag{7}$$

where $\Upsilon(y)$ is target distribution and given by dynamic labeling [24]: $y = \mathrm{argmax}_{i\in\{1,\dots,c\}} D_i(x)$, $\Upsilon_i(y) = 1$ if $i = y$, else $\Upsilon_i(y) = 0$. H is the cross entropy, G and D indicate generator and discriminator, $D(x)$ is the output of discriminator on x.

Discriminator of original AM-GAN is:

$$\begin{aligned}L_D^{AM-GAN} &= \mathbb{E}_{(x,y)\sim p_{\mathrm{data}}}[H(\Upsilon(y), D(x))]\\&+\mathbb{E}_{x\sim G}[H(\Upsilon(c+1), D(x))],\end{aligned} \tag{8}$$

where c is number of positive classes, we remove the additional class and minimize the KL divergence between output and target distribution:

$$\begin{aligned}L_D^{NAT-GAN} &= \mathbb{E}_{(x,y)\sim p_{\mathrm{data}}}[H(\Upsilon(y), D(x))]\\&+\mathbb{E}_{x\sim G,neg}[KL(\Upsilon_{neg} \parallel D(x))],\end{aligned} \tag{9}$$

where $\Upsilon_{neg} = (1/c, \dots, 1/c)^T$.

Generator and discriminator losses of CatGAN are:

$$\begin{aligned}L_G^{CatGAN} &= -\mathbb{E}_{(x,y)\sim G}[H(\Upsilon(y), D(x))]\\&+H_{x\sim G}\left[\frac{1}{M}\sum_{i=1}^{M} D(x)\right],\end{aligned} \tag{10}$$

$$\begin{aligned}L_D^{CatGAN} &= \mathbb{E}_{(x,y)\sim G}[H(\Upsilon(y), D(x))]\\&-\mathbb{E}_{(x,y)\sim p_{\mathrm{data}}}[H(\Upsilon(y), D(x))]\\&+H_{x\sim p_{\mathrm{data}}}\left[\frac{1}{N}\sum_{i=1}^{N} D(x^i)\right],\end{aligned} \tag{11}$$

where M and N are numbers of samples, the last items of L_G^{CatGAN} and L_D^{CatGAN} are marginal class distributions [19], they explicitly optimize the diversity of generated samples.

NAT-GAN alleviates the overfitting on fake class. Let l be the output logits vector and $\sigma(l) = D(x)$ be the softmax probability distribution, Υ be the target probability distribution, then:

$$-\frac{\partial H(\Upsilon, \sigma(l))}{\partial l} = \Upsilon - \sigma(l), \tag{12}$$

For AM-GAN, the object function punishes $\Upsilon_{c+1} - \sigma(l)_{c+1}$ for generated samples and that results in unbalance on $c+1$-th class. However, NAT-GAN and CatGAN punish the weights of all classes, that alleviates the overfit on additional fake class and assists with steady gradient information.

Evaluation Metrics. Inception Score (IS) [17] is well correlated with human evaluation, AM Score[24] is proposed to measure the quality of generated samples as a compensation of IS. They are calculated via:

$$\text{Inception Score} = \exp\left(\mathbb{E}_{x\sim G}[\mathrm{KL}(C(x)\|\overline{C}^G)]\right), \tag{13}$$

$$\text{AM Score} \triangleq \mathrm{KL}\left(\overline{C}^{\text{train}}\|\overline{C}^G\right) + \mathbb{E}_x[H(C(x))], \tag{14}$$

where $\overline{C}^G = \mathbb{E}_x[C(x)]$ is the overall probability distribution of the generated samples over classes judged by C. AM Score requires $\overline{C}^G$

Table 1: CIFAR 10 and SVHN NCR results

Positive		CIFAR 10				SVHN			
Negative		CIFAR 100 Train Set	CIFAR 100 Test Set	ILSVRC	Random Noise	CIFAR 100 Train Set	CIFAR 100 Test Set	ILSVRC	Random Noise
$NCR_{0.4}$	baseline	0.960	0.971	0.949	0.736	0.928	0.934	0.921	0.944
	NAT	**0.028**	**0.072**	**0.063**	**0**	**0.0003**	**0.003**	**0.002**	**0**
$NCR_{0.6}$	baseline	0.817	0.839	0.764	0.145	0.727	0.738	0.706	0.760
	NAT	**0.012**	**0.045**	**0.036**	**0**	**0.0001**	**0.001**	**0.0009**	**0**
$NCR_{0.8}$	baseline	0.649	0.671	0.563	0.006	0.520	0.530	0.489	0.561
	NAT	**0.005**	**0.026**	**0.019**	**0**	**0**	**0.0008**	**0.0005**	**0**
$NCR_{0.9}$	baseline	0.534	0.558	0.434	0.0001	0.388	0.402	0.357	0.429
	NAT	**0.002**	**0.018**	**0.012**	**0**	**0**	**0.0006**	**0.0004**	**0**

close to $\overline{C}^{\text{train}}$ and each sample x has a low entropy $C(x)$. The minimal value of AM Score is zero and the smaller the better. Inception Score requires each samples distribution $C(x)$ different from the overall distribution of the generator $\overline{C}^{G}$, which indicates good diversity and quality over the generated samples. Steady and smooth gradient information is given as the discriminator loss of ours explicitly punishes KL divergence between distributions of real and generated, but generator optimizes KL divergence, and this adversarial procedure matches the goals of Inception Score and AM Score.

FID [6] compares the statistics of generated samples to real samples:

$$\begin{aligned}\text{FID}(x, g) = &\|\mu_x - \mu_g\|_2^2 \\ &+ \text{Tr}\left(\Sigma_x + \Sigma_g - 2\left(\Sigma_x\Sigma_g\right)^{\frac{1}{2}}\right),\end{aligned} \tag{15}$$

where x is short of $x \sim p_{data}$ and g is short of $x \sim G$. μ, Σ, Tr are mean, covariance and diagonal elements sum respectively. FID measures the quality and diversity of generated samples and is sensitive to diversity especially.

3 Experiments

3.1 NAT classification

CIFAR 10 [11] and SVHN [14] are image datasets with 10 classes and used for our classification task. CIFAR 100 training set is used as negative samples during training. Three datasets will be tested after training: CIFAR 100 test set, selected data samples from ILSVRC 2012 [3] (resized to 32, refered as ILSVRC), and random noises generated from standard normal distribution, they are all normalized. Baseline networks are trained only on CIFAR 10 training set, and the networks with NAT setting trained both on CIFAR 10 training set and CIFAR 100 training set (as negative samples). We use ResNet 18 [5], and set thresholds of NCR from $\{0.4, 0.6, 0.8, 0.9\}$, batch size is 128, and there are 5 to 25 random selected negative samples per batch.

Table 1 displays the NCR results. and the Figure 4 shows the maximum prediction of each network on CIFAR 100 test set. The superiority of NAT is clear, baseline predicts negative samples with high confidence while the network with NAT setting performs well not only on CIFAR 100 training set, but also on other unseen datasets, that demonstrates the generalization of NAT. Surprisingly, the network with NAT even got 0% NCR on random noise in our experiments, but the results vary on different conditions. Meanwhile, the performance of the original classification task still holds, baseline got

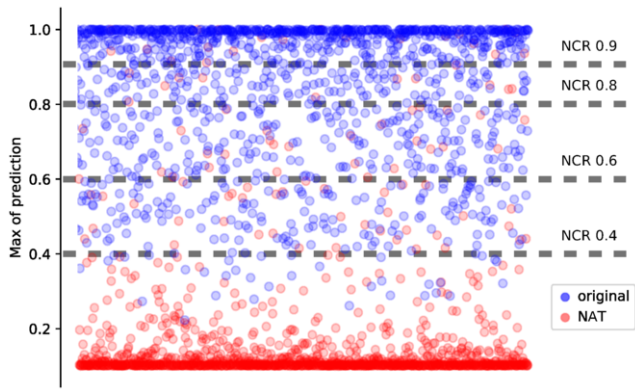

Figure 4: The maximum predictions of baseline (blue) and NAT (red) on CIFAR 100 test set. Most predictions of baseline are of incorrect high confidences, while NAT is more robust on negative samples.

Table 2: CIFAR 10 NCR results of 8, 32, 64 negative samples per batch (batch size 128)

# per batch		CIFAR 100 Train Set	CIFAR 100 Test Set	ILSVRC	Random Noise
8	$NCR_{0.4}$	0.123	0.155	0.218	0.142
	$NCR_{0.6}$	0.077	0.111	0.154	0.001
	$NCR_{0.8}$	0.045	0.073	0.104	0
	$NCR_{0.9}$	0.029	0.052	0.076	0
32	$NCR_{0.4}$	0.025	0.087	0.139	0.082
	$NCR_{0.6}$	0.010	0.059	0.099	0.002
	$NCR_{0.8}$	0.004	0.039	0.069	0
	$NCR_{0.9}$	0.002	0.027	0.051	0
64	$NCR_{0.4}$	0.010	0.082	0.132	0.608
	$NCR_{0.6}$	0.003	0.054	0.093	0.144
	$NCR_{0.8}$	0.001	0.035	0.064	0.011
	$NCR_{0.9}$	0.0003	0.027	0.047	0.008

Table 3: NCR overfit results trained on CIFAR 10 and ILSVRC samples

Test Sets	ILSVRC Train	ILSVRC Test	CIFAR 100	Random Noise
$NCR_{0.4}$	0	0.079	0.597	1
$NCR_{0.6}$	0	0.047	0.464	1
$NCR_{0.8}$	0	0.025	0.349	0.051
$NCR_{0.9}$	0	0.014	0.262	0

Table 4: Comparison of IS, AM Score and FID

Dataset	Setting	IS	AM Score	FID
CIFAR 10	SN-GAN[13]	8.22	-	21.7
	MMD-GAN[22]	8.29	-	16.21
CIFAR	AM-GAN	8.88	0.073	13.49
	CatGAN_0	9.02	0.085	14.20
	CatGAN_1	9.06	0.078	15.27
	CatGAN_3	9.05	0.078	14.89
	CatGAN_10	9.08	0.078	15.43
	NAT-GAN_0	9.11	0.056	**13.03**
	NAT-GAN_1	9.21	**0.034**	15.12
	NAT-GAN_3	**9.26**	0.043	14.58
	NAT-GAN_10	9.25	0.036	15.09
Tiny-ImageNet subset	AM-GAN	5.91	0.641	43.85
	CatGAN_0	10.46	0.453	28.93
	CatGAN_1	10.66	0.444	29.07
	CatGAN_3	10.65	0.446	28.63
	CatGAN_10	10.60	0.452	28.38
	NAT-GAN_0	11.48	0.440	25.54
	NAT-GAN_1	11.60	0.436	24.52
	NAT-GAN_3	**11.78**	**0.428**	**24.38**
	NAT-GAN_10	11.73	0.436	25.02

95.54% test accuracy and NAT got 95.03%. Treating negative samples as additional class suffers greatly from unbalancing problem, it got 91% accuracy and overfit severely.

We also evaluate the performance of network trained on CIFAR 10 and ILSVRC samples, as shown in Table 3. And the ratios of negative samples, which are 8, 32, 64 negative samples per batch and shown in Table 2. An observation is that high performance requires high quality negative samples. Overfit on negative will occur if the negative is much different with positive, CIFAR 100 is better than ILSVRC samples for training CIFAR 10 in our experiments. For the ratio of negative, the overfit is neglectable as long as the positive samples hold the major percentage.

3.2 NAT-GAN

3.2.1 1D NAT-GAN

We easily verify the performance of NAT-GAN on 1 dimension GAN as shown in Figure 5 and NAT website. Prior information of negative distribution assists with robust gradient information and steadier adversarial procedure. The decision boundary is learned faster and the generator converged better compared with original GAN, decision boundary of which is noisy and shifting rapidly.

The quality of negative distribution is important, if there are more available negative distribution, the adversarial procedure is not only faster but also alleviates the fluctuation.

3.2.2 NAT-GAN on images

Further, we evaluate NAT-GAN on CIFAR 10 dataset, in which CIFAR 100 is the negative dataset, and a subset of Tiny-ImageNet, in which the first 100 classes are positive, and the last 20 classes are negative. NAT-GAN shares the same setting with AM-GAN, but the discriminator of ours removed the additional fake class, and use KL divergence to minimize the distribution distance of fake data outputs and target distribution, which is 0.1 on CIFAR 10, and 0.01 on Tiny-ImageNet. The optimizer is Adam(beta1=0.5, beta2=0.999) [9], batch size is 100, and initial learning rate is 0.0004.

We experimented several settings with different numbers of negative samples, AM-GAN, CatGAN, CatGAN_X, NAT-GAN_X, X indicates number of negative samples per batch, in NAT-GAN_3 we add 3 external negative samples one batch. In NAT-GAN_0 we didn't add external negative samples and is similar to CatGAN but simpler. We found that AM-GAN won't converge with external negative samples per epoch because it suffers from unbalancing problem. The evaluation metrics are AM Score, FID and Inception Score, better generated samples achieve lower AM Score and FID but higher Inception Score.

As shown in Table 4, NAT-GAN surpasses AM-GAN on AM Score and achieves state-of-the-art Inception Score (over 9.2) on CIFAR 10, and can be further improved with longer training. However, it is slightly weaker on FID due to the constraint of negative, which means the diversity is decreasing, but still outperforms other methods without negative, like SN-GAN(21.7) [13] and MMD-GAN(16.21) [22], and our NAT-GAN_0 got best FID (13). The quality of generated samples is improved as negative samples provide prior knowledge, and that made a stronger generator compared with counterparts. We observed that the IS becomes worse (9.03) if add too much – like 20 – external negative samples as the negative regularization severely hurts the diversity. Losses of generators on multiple datasets are shown in Figure 6, which demonstrates that negative information assists with better convergence.

4 Conclusion

We study the function of negative samples in supervised classification and GAN, and propose a training strategy called Negative-Aware Training (NAT) and NAT-GAN, both positive samples and external negative samples are jointly trained. Cost function on negative samples, which is distribution metric like KL divergence, forces the network to output equal probability, and original architecture and settings stay unchanged. **1)** The network with NAT is more robust and neutral, addresses the problem of random prediction on negative samples with high confidence and holds the performance on original classification task, and helps the application in practice. Experiments on CIFAR 10 and Tiny-ImageNet shows that with NAT, the network performs well both on negative samples used for training and on other negative samples based on proposed Negative Confidence Rate (NCR), and demonstrates the generalization of NAT. **2)** Our NAT-GAN, based on CatGAN and AM-GAN, uses both external negative samples and generated samples to train discriminator. The external negative samples assist with steady adversarial procedure and better boundary information, help GAN convergence faster and achieve inspiring state-of-the-art AM Score and Inception Score (9.2) on CIFAR 10.

Collecting negative samples requires less and weak human knowledge compared with positive samples. And when collecting positive dataset is needed in real application, it is trivial to collected negative samples in the meantime. Negative samples provide prior knowledge and boundary information, but the measurement of how good are negative samples is still an open question, further application and expansion on other tasks will be studied in future work.

ACKNOWLEDGEMENTS

This work was supported by the NSFC-Key Project under Grant No. 61933013, the NSFC-Key Project of General Technology Fundamental Research United Fund under Grant No. U1736211, the Natural Science Foundation of Guangdong Province under Grant No. 2019A1515011076, the Key Project of Natural Science Foundation of Hubei Province under Grant No. 2018CFA024, the Inno-

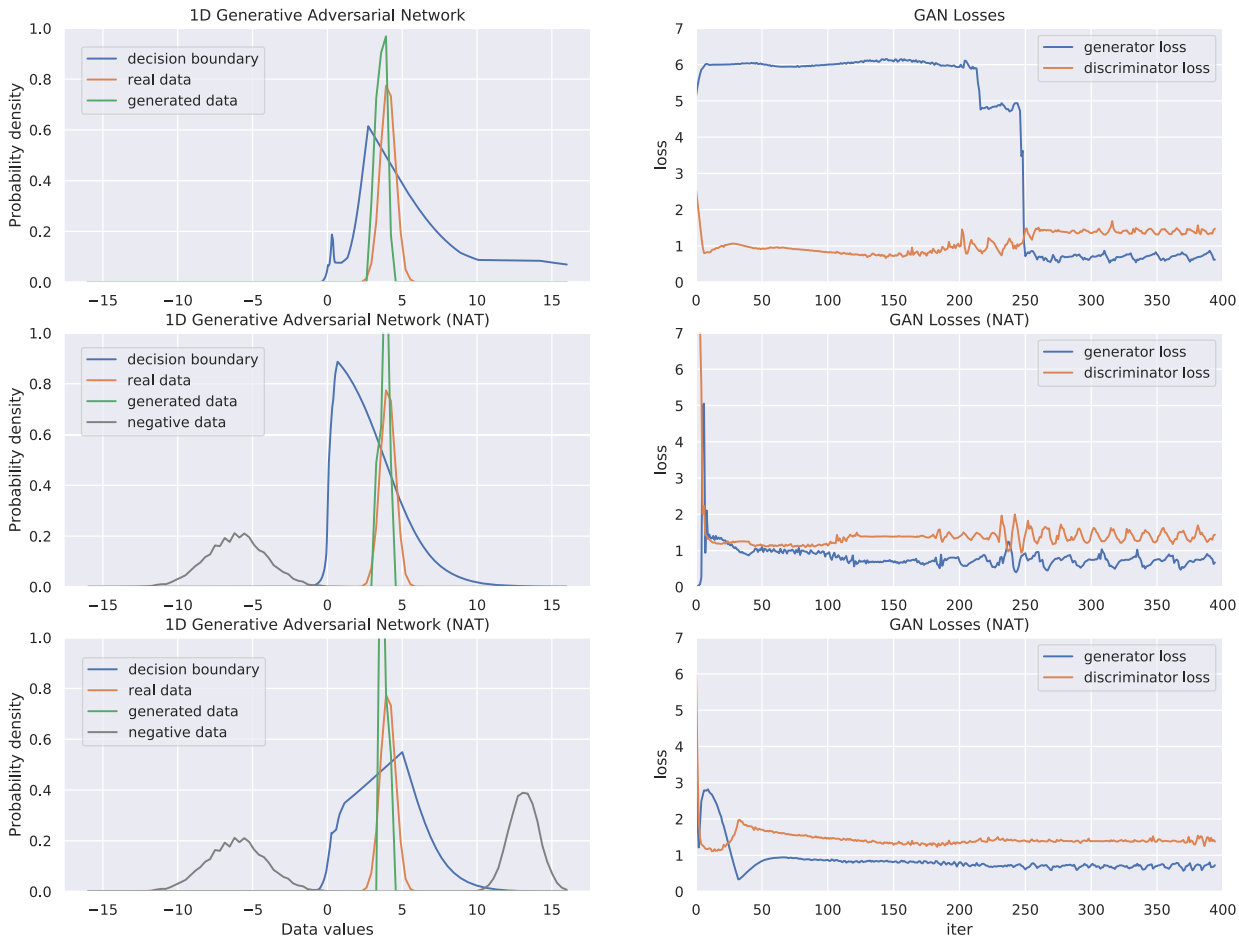

Figure 5: Comparison of NAT and original on 1D GAN. Negative distributions assist generator to learn faster and steadier, the demo video is available on NAT website (and also 2D GAN).

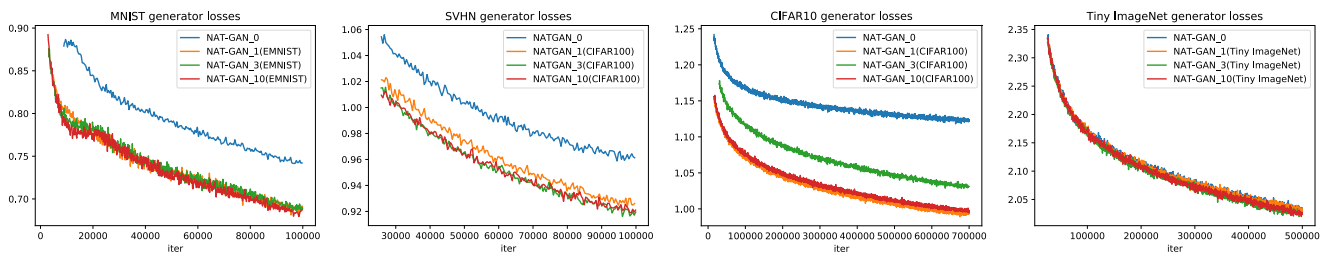

Figure 6: Generator losses of NAT-GAN on multiple datasets.

vation Group of Guangdong Education Department under Grant NO. 2018KCXTD019.

REFERENCES

[1] Martín Arjovsky, Soumith Chintala, and Léon Bottou, 'Wasserstein gan', *CoRR*, **abs/1701.07875**, (2017).

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan, 'Large scale gan training for high fidelity natural image synthesis', *CoRR*, **abs/1809.11096**, (2018).

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, 'Imagenet: A large-scale hierarchical image database', *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255, (2009).

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, 'Generative adversarial nets', in *Advances in neural information processing systems*, pp. 2672–2680, (2014).

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, (2016).

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, 'Gans trained by a two time-scale update rule converge to a local nash equilibrium', in *Advances in Neural Information Processing Systems*, pp. 6626–6637, (2017).

[7] Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie, 'Stacked generative adversarial networks', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5077–5086, (2017).

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, 'Image-to-image translation with conditional adversarial networks', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, (2017).

[9] Diederik P. Kingma and Jimmy Ba, 'Adam: A method for stochastic optimization', *CoRR*, **abs/1412.6980**, (2015).
[10] Andreas Krause, Pietro Perona, and Ryan G Gomes, 'Discriminative clustering by regularized information maximization', in *Advances in neural information processing systems*, pp. 775–783, (2010).
[11] Alex Krizhevsky and Geoffrey Hinton, 'Learning multiple layers of features from tiny images', Technical report, Citeseer, (2009).
[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in neural information processing systems*, pp. 1097–1105, (2012).
[13] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida, 'Spectral normalization for generative adversarial networks', in *International Conference on Learning Representations*, (2018).
[14] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng, 'Reading digits in natural images with unsupervised feature learning', *NIPS*, (01 2011).
[15] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski, 'Plug & play generative networks: Conditional iterative generation of images in latent space', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4467–4477, (2017).
[16] Augustus Odena, Christopher Olah, and Jonathon Shlens, 'Conditional image synthesis with auxiliary classifier gans', in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2642–2651. JMLR. org, (2017).
[17] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, 'Improved techniques for training gans', in *Advances in neural information processing systems*, pp. 2234–2242, (2016).
[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.
[19] Jost Tobias Springenberg, 'Unsupervised and semi-supervised learning with categorical generative adversarial networks', *CoRR*, **abs/1511.06390**, (2016).
[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, 'Going deeper with convolutions', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, (2015).
[21] Chaoyue Wang, Chang Xu, Xin Yao, and Dacheng Tao, 'Evolutionary generative adversarial networks', *IEEE Transactions on Evolutionary Computation*, (2019).
[22] Wei Wang, Yuan Sun, and Saman Halgamuge, 'Improving MMD-GAN training with repulsive loss function', in *International Conference on Learning Representations*, (2019).
[23] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas, 'Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5907–5915, (2017).
[24] Zhiming Zhou, Han Cai, Shu Rong, Yuxuan Song, Kan Ren, Weinan Zhang, Jun Wang, and Yong Yu, 'Activation maximization generative adversarial nets', in *International Conference on Learning Representations*, (2018).
[25] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros, 'Generative visual manipulation on the natural image manifold', in *European Conference on Computer Vision*, pp. 597–613. Springer, (2016).