Hallucinating a Cleanly Labeled Augmented Dataset from a Noisy Labeled Dataset Using GANs

F. Chiaroni, M-C. Rahal, N. Hueber, Frédéric Dufaux

▶ **To cite this version:**

F. Chiaroni, M-C. Rahal, N. Hueber, Frédéric Dufaux. Hallucinating a Cleanly Labeled Augmented Dataset from a Noisy Labeled Dataset Using GANs. 26th IEEE International Conference on Image Processing (ICIP 2019), Sep 2019, Taipei, Taiwan. pp.3616-3620, 10.1109/ICIP.2019.8803632 . hal-02054836

HAL Id: hal-02054836
https://hal.science/hal-02054836

Submitted on 2 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HALLUCINATING A CLEANLY LABELED AUGMENTED DATASET FROM A NOISY LABELED DATASET USING GAN

F. Chiaroni[⋆‡] *M-C. Rahal*[⋆] *N. Hueber*[†] *F. Dufaux*[‡]

[⋆] VEDECOM Institute, Department of delegated driving (VEH08), Perception team, {florent.chiaroni, mohamed.rahal}@vedecom.fr
[†] French-German Research Institute of Saint-Louis (ISL), ELSI team, nicolas.hueber@isl.eu
[‡] L2S-CNRS-CentraleSupelec-Univ Paris-Sud, frederic.dufaux@l2s.centralesupelec.fr

ABSTRACT

Noisy labeled learning methods deal with training datasets containing corrupted labels. However, prediction performances of existing methods on small datasets still leave room for improvements. With this objective, in this paper we present a GAN-based method to generate a clean augmented training dataset from a small and noisy labeled dataset. The proposed approach combines noisy labeled learning principles with GAN state-of-the-art techniques. We demonstrate the usefulness of the proposed approach through an empirical study on simple and complex image datasets.

***Index Terms*—** Generative adversarial networks, noisy labeled learning, image classification

1. INTRODUCTION

Nowadays, the lack of clean labeled datasets remains an issue in many image classification applications. As a consequence, we need to tackle the problem of handling noisy labeled datasets. In the context of binary classification, a noisy labeled dataset contains examples of the positive and negative classes, however, a fraction of the training examples are mislabeled.

Noisy labeled learning methods [1], [2], [3] target this issue. A state-of-the-art solution is Rank Pruning (RP) [4]. It consists in first iteratively identifying confident positive and negative examples. During the second step, it trains a classifier with identified examples by considering them as correctly labeled. However, small and complex noisy labeled datasets remain challenging. It turns out that GAN-based approaches [5], [6] have demonstrated state-of-the-art prediction performances to overcome similar issues on partially labeled datasets. In particular, GANs are compelling for sub-distributions hallucination and for data augmentation on small and complex datasets. RP and GAN-based approaches consist in preparing a clean Positive-Negative (PN) dataset from the input noisy one. They are referred to as two-stage methods.

We recall that the original GAN [7] is an unsupervised generative model. It contains a classifier model, often called discriminator D, and a generator G. D is trained to distinguish real samples x_R from generated samples $G(z)$, with z an input random vector following a uniform or normal distribution p_z. Adversarially, G is trained to generate examples which are considered as real as possible by D. In this way, the generated examples distribution converges towards the real examples distribution p_R. This two-player game can be formalized with the following minimax value function:

$$\begin{aligned}\min_G \max_D V(G,D) =& \mathbb{E}_{x_R \sim p_R}[-H(D(x_R),1)] \\ &+ \mathbb{E}_{z \sim p_z}[-H(D(G(z)),0)],\end{aligned} \tag{1}$$

with H the binary cross entropy metric. Moreover, the GAN litterature provides nowadays effective techniques to overcome the original mode collapse issue [8] and to improve the hallucinated examples quality [9]. The DCGAN [10] method stabilizes the original GAN for image datasets by using convolutional layers and batch normalization (BN) [11]. The spectral GAN [12] increases the examples quality by replacing BN with the spectral normalization (SN). Even more recently, the SAGAN [13] has incorporated attention layers to take into consideration spatial features correlations.

To sum up, on the one hand the noisy labeled learning methods can manage noisy labeled datasets. On the other hand, GAN-based approaches have demonstrated their effectiveness for the partially labeled learning task on small and complex datasets. For these reasons, we propose a novel GAN-based approach to tackle the noisy labeled learning task on small and complex datasets. The main contributions of this work consist in:

- incorporating a noisy labeled risk inside the GAN discriminator loss function;
- applying carefully regularization techniques during the GAN adversarial training. This addresses GAN mode collapse and discriminator overfitting issues;
- exploiting prior knowledge of the corrupted labels fractions in order to estimate the most appropriate adversarial training labels.

The outline of the paper is as follow. Section 2 presents the proposed approach. Section 3 presents the experimental results. Then, the article ends by a conclusion.

2. PROPOSED METHOD

The insight of the proposed approach is to train two generators to generate examples which are considered by the discriminator as the most positive, respectively most negative, with the highest confidence as possible. To correctly guide the generators, we first identify the discriminator prediction behaviour when it is trained on a noisy labeled dataset.

We start by describing the noisy labeled dataset. The positive and negative samples x_P and x_N follow distributions p_P and p_N respectively. The noisy labeled training dataset is composed of partially corrupted positive and negative samples $x_{\hat{P}}$ and $x_{\hat{N}}$ with the distributions $p_{\hat{P}}$ and $p_{\hat{N}}$ respectively. These latter are mixtures of distributions of p_P and p_N such that $p_{\hat{P}} = \pi_P p_P + (1-\pi_P)p_N$ and $p_{\hat{N}} = \pi_N p_N + (1-\pi_N)p_P$. π_P is the fraction of correctly labeled (not corrupted) positive examples, and π_N is the fraction of correctly labeled (not corrupted) negative examples. Finally, we make the assumption that $(\pi_P + \pi_N) \in (1, 2)$, such that the majority of labels are not corrupted.

2.1. Noisy labeled training

We train the discriminator D to predict the label value 0 for corrupted positive samples $x_{\hat{P}}$ and the label value 1 for corrupted negative samples $x_{\hat{N}}$ such that the corresponding training loss function L_{Noisy} is defined as

$$\begin{aligned} L_{Noisy}(D) =& \mathbb{E}_{x_{\hat{P}} \sim \boldsymbol{p_{\hat{P}}}}[H(D(x_{\hat{P}}), \mathbf{0})] \\ &+ \mathbb{E}_{x_{\hat{N}} \sim \boldsymbol{p_{\hat{N}}}}[H(D(x_{\hat{N}}), \mathbf{1})]. \end{aligned} \tag{2}$$

If we use the binary cross entropy H metric in the training loss function, we can consider this noisy labeled loss function as a biased clean labeled loss function. In other words, we can also formulate L_{Noisy} as follow[1]

$$\begin{aligned} L_{Noisy}(D) =& (\pi_P + (1-\pi_N))\mathbb{E}_{x_P \sim \boldsymbol{p_P}}[H(D(x_P), \boldsymbol{\delta_P})] \\ &+ ((1-\pi_P) + \pi_N)\mathbb{E}_{x_N \sim \boldsymbol{p_N}}[H(D(x_N), \boldsymbol{\delta_N})], \end{aligned} \tag{3}$$

with $\delta_P = \frac{(1-\pi_N)}{\pi_P + (1-\pi_N)}$ and $\delta_N = \frac{\pi_N}{(1-\pi_P)+\pi_N}$. In practice, if we do not know prior π_P and π_N, we can estimate δ_P and δ_N values with a clustering algorithm such as a Gaussian Mixture Model (GMM) [14]. It is sufficient to apply GMM on the discriminator prediction output for a training batch of noisy labeled examples (see figure 1).

[1]This equality between equations 2 and 3 can be demonstrated easily by developing and factoring those formulas by taking into consideration expectations linearity and distributions compositions.

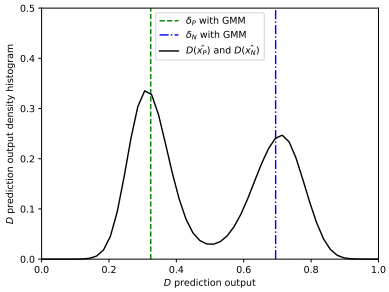

Fig. 1. Histogram of discriminator output predictions for a training batch including the same proportion of $x_{\hat{P}}$ samples and $x_{\hat{N}}$ samples. We trained D during 15 epochs on the MNIST dataset with "5" the positive class, "7" the negative class, $\pi_P = 0.7$ and $\pi_N = 0.7$. GMM clustering algorithm identifies empirically δ_P and δ_N. This histogram is empirically consistent with the proposed equality between equations 2 and 3.

2.2. Noisy labeled image classification

Concerning the noisy labeled learning task, we firstly use the proposed GAN-based approach to generate a clean augmented dataset from the noisy labeled one. Figure 2 presents the schema of this first-stage framework.

2.2.1. Generative models step: training loss functions

The proposed GAN-based model contains a discriminator D, a positive generator G_P, and a negative generator G_N. We train G_P to hallucinate fake positive samples x_{GP} and we train G_N to hallucinate fake negative samples x_{GN}. We train G_P and G_N to minimize loss functions L_{GP} and L_{GN}, using labels δ_P and δ_N, as follow

$$\begin{cases} L_{GP}(D, G_P) = \mathbb{E}_{x_{GP} \sim \boldsymbol{p_{GP}}}[H(D(x_{GP}), \boldsymbol{\delta_P})] \\ L_{GN}(D, G_N) = \mathbb{E}_{x_{GN} \sim \boldsymbol{p_{GN}}}[H(D(x_{GN}), \boldsymbol{\delta_N})]. \end{cases} \tag{4}$$

Moreover, we train adversarially D with G_P and G_N such that we define D training loss function L_D as

$$\begin{aligned} L_D(D, G_P, G_N) =& \alpha \cdot [\mathbb{E}_{x_{\hat{P}} \sim \boldsymbol{p_{\hat{P}}}}[H(D(x_{\hat{P}}), \mathbf{0})] \\ &+ \mathbb{E}_{x_{\hat{N}} \sim \boldsymbol{p_{\hat{N}}}}[H(D(x_{\hat{N}}), \mathbf{1})]] \\ &+ \beta \cdot [\mathbb{E}_{x_{GP} \sim \boldsymbol{p_{GP}}}[H(D(x_{GP}), \mathbf{1})] \\ &+ \mathbb{E}_{x_{GN} \sim \boldsymbol{p_{GN}}}[H(D(x_{GN}), \mathbf{0})]], \end{aligned} \tag{5}$$

with α and β the hyper-parameters such that $\alpha >> \beta$. This accentuates the guidelines to train G_P and G_N to converge towards the positive and negative samples distribution. As D, G_P and G_N are deep convolution models, we backpropagate the training errors in their weights with the stochastic gradient descent (SGD) method [15].

Note that in practice, we can replace H by the mean squared error (MSE) metric, while preserving the same training labels.

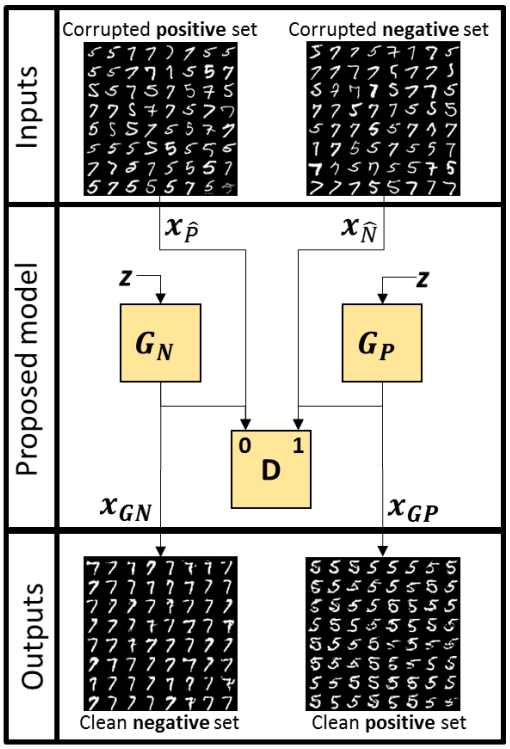

Fig. 2. Proposed GAN-based label denoising model: This ullustrates how to output a generated cleanly labeled augmented dataset from a small input noisy labeled dataset. z represents an input random vector following a uniform or normal distribution p_z, such that $x_{GP} = G_P(z)$ and $x_{GN} = G_N(z)$.

2.2.2. *Posterior step: A standard classification*

Concerning the binary noisy labeled classification task, once we have generated a cleanly labeled augmented dataset with the proposed generative model during the first stage, we can train a classifier with this relevant dataset by considering x_{GP} and x_{GN} samples as respectively real correctly labeled samples x_P and x_N.

2.2.3. *Regularizations*

In practice, regularization techniques ensure the expected behaviour. We use BN to help the generators training stability and to accelerate their convergence. However, in the discriminator we rather use SN instead of BN. As SN is a weight normalization technique, it is not influenced by the use of four different minibatch samples distributions (see figure 2). Moreover, we avoid overfitting problems on small datasets by using dropout [16] in the discriminator. More specifically, we activate it during the discriminator training while it is disabled during the generators trainings.

The next section demonstrates the effectiveness of the proposed approach through an empirical study.

3. EXPERIMENTS

The proposed approach has been tested on small and complex images datasets MNIST [17] and CIFAR-10 [18]. First, we present experimental settings. Then, we present the cleanly labeled samples generated from noisy labeled datasets. Finally, we show that the accuracy prediction performances obtained on small, complex and highly corrupted image datasets confirm the proposed approach competitiveness.

3.1. Settings

Concerning the loss functions L_D, L_{GP} and L_{GN}, we established empirically $\alpha = 5$ and $\beta = 0.5$. For the corresponding first-stage learning models D, G_P and G_N, we adapted the previous DCGAN [10] architecture to this novel framework. D contains two bottom convolutional layers, followed by two top fully-connected layers. The input convolutional layer contains 64 3*3 filters, the next one has 128 3*3 filters, and the hidden fully connected layer has 1024 filters. G_P and G_N contain symetrically two bottom fully connected layers followed by two deconvolutional layers with the same number of filters. The generators input is a vector z of 100 random values following a uniform distribution. As discussed in the regularization subsection, we use BN on the generators deconvolutional layers. In D, we apply SN on convolutional layers, and dropout of 0.5 in the fully connected hidden layer. To deal with the relatively complex CIFAR-10 image dataset containing RGB images 32*32*3, we included in D an additional hidden convolutional layer with 256 filters. G_P and G_N consequently include a hidden deconvolutional layer with the same number of filters. Concerning the second-stage classifier, we use the convolutional structure previously mentioned in [5] [2]. We use the SGD method Adam [19] for all previously enumerated learning models, and a learning rate initialized to $2 \cdot 10^{-4}$ during the first-stage and to $1 \cdot 10^{-4}$ during the classification step. We train D, G_P and G_N adversarially during 40 epochs on MNIST and during 500 epochs on CIFAR-10. Then, we train during 25 epochs the classifier, as we train RP [3] during 25 epochs.

We simulated the corrupted labels from fully cleanly labeled datasets MNIST and CIFAR-10 in order to respect prior knowledge parameters π_P and π_N. Then, we reduced the dataset size by selecting the first 1000 or 100 training examples with the associated simulated corrupted labels. A size of 100 for the MNIST task $\{5; 7 - vs - 2, 4\}$ means that we use only 25 examples for each subclass. Thus, after the dataset reduction, we systematically do an upsampling such that the training dataset used always has a size of 10000 examples. This introduces redundancy in the training dataset, but this

[2] $https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/tutorials/mnist/mnist\_softmax.py$

[3] RP code is available at: $https://github.com/cgnorthcutt/rankpruning$

Table 1. Two-stage noisy labeled learning comparative results in terms of test accuracy prediction performances on small noisy labeled image datasets. As we use a SGD optimization method, each result is respectively the average of five identical independent trainings.

Test Accuracy	ref	$\pi_P = 0.85, \pi_N = 0.85$		$\pi_P = 0.7, \pi_N = 0.85$		$\pi_P = 0.6, \pi_N = 0.65$	
$\{5;7\}$-vs-$\{2;4\}_{\text{MNIST}}$	PN	NL-GAN	RP	NL-GAN	RP	NL-GAN	RP
Size: 1000	0.97	0.961	**0.965**	**0.956**	0.948	**0.926**	0.809
Size: 100	0.9	**0.909**	0.892	**0.881**	0.853	**0.814**	0.703
Car-vs-Airplane$_{\text{CIFAR-10}}$	PN	NL-GAN	RP	NL-GAN	RP	NL-GAN	RP
Size: 1000	0.878	**0.843**	0.833	**0.824**	0.794	**0.704**	0.659
Size: 100	0.789	**0.789**	0.761	**0.782**	0.739	**0.687**	0.64

mainly enables to keep the same number of epochs iterations for any dataset reduction.

3.2. Qualitative results

Figure 3 illustrates the images that the proposed approach is able to generate on MNIST and the natural image dataset CIFAR-10. We corrupt up to 40 percents of the training labels. However, despite the fact that the generated examples are cleanly labeled, the hallucinated images quality probably still has the potential to be improved with hyper-parameters fine-tuning study in the context of this novel framework.

3.3. Comparative results

Table 1 presents comparative accuracy prediction performances on small corrupted training datasets. PN baseline reference in table 1 represents a training of the classifier, used during the second stage, on the initial dataset reduced and without corrupted labels, such that $\pi_P = 1$ and $\pi_N = 1$. The other columns show results for three experiments: $\pi_P = \pi_N = 0.85$, $\pi_P = 0.70$ and $\pi_N = 0.85$, and $\pi_P = 0.60$ and $\pi_N = 0.65$, respectively.

The proposed approach, referred to as Noisy Labeled GAN (NL-GAN), globally outperforms RP method on both MNIST and CIFAR-10 datasets. In particular, the proposed approach becomes especially interesting with high fractions of corrupted training labels. Nonetheless, because of the adversarial training, we recall that 500 first-stage epochs iterations were necessary on CIFAR-10 to get these results while only 40 epochs are necessary on MNIST. Therefore, if we can afford the computational complexity, the proposed approach remains competitive on complex image datasets like CIFAR-10.

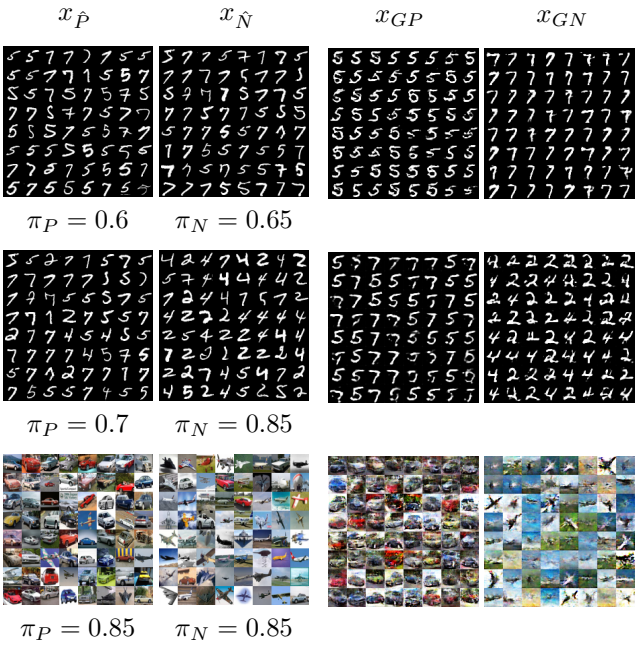

Fig. 3. Cleanly labeled dataset generation from noisy labeled datasets. The two left columns present noisy labeled minibatch input positive samples $x_{\hat{P}}$ and negative samples $x_{\hat{N}}$. The two right columns present output generated minibatch samples x_{GP} and x_{GN}. The first row presents results for MNIST classification task 5-*vs*-7 when $\pi_P = 0.6$ and $\pi_N = 0.65$. The second row presents results for MNIST classification task $\{5;7\}$-*vs*-$\{2;4\}$ when $\pi_P = 0.7$ and $\pi_N = 0.85$. The third row presents results for CIFAR-10 classification task *Car-vs-Airplane* when $\pi_P = 0.85$ and $\pi_N = 0.85$. Visually, every generated samples observed hallucinate cleanly labeled examples.

4. CONCLUSION

In this paper, we proposed a novel GAN-based framework to deal with small noisy labeled image datasets. Experimental results show that it is possible to generate a clean dataset from noisy labels with a GAN. The proposed approach compares favorably with the state-of-the-art in terms of prediction performances. Moreover, we expect that the proposed approach can further be improved by including recent GAN-based advances [20]. In particular, it may be relevant to take into consideration recent GANs using several generators [21].

5. REFERENCES

[1] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari, "Learning with Noisy Labels," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 1196–1204. Curran Associates, Inc., 2013.

[2] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang, "Learning from massive noisy labeled data for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2691–2699.

[3] Shantanu Jain, Martha White, and Predrag Radivojac, "Estimating the class prior and posterior from noisy positives and unlabeled data," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 2693–2701. Curran Associates, Inc., 2016.

[4] Curtis G. Northcutt, Tailin Wu, and Isaac L. Chuang, "Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels," *arXiv preprint arXiv:1705.01936*, 2017.

[5] Florent Chiaroni, Mohamed-Cherif Rahal, Nicolas Hueber, and Frederic Dufaux, "Learning with a generative adversarial network from a positive unlabeled dataset for image classification," in *IEEE International Conference on Image Processing*, 2018.

[6] Ming Hou, Qibin Zhao, Chao Li, and Brahim Chaib-draa, "A generative adversarial framework for positive-unlabeled classification," *arXiv preprint arXiv:1711.08054*, 2018.

[7] 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*, 2014, pp. 2672–2680.

[8] 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*, 2016, pp. 2234–2242.

[9] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018.

[10] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[11] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[12] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[13] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena, "Self-Attention Generative Adversarial Networks," *arXiv preprint arXiv:1805.08318*, 2018.

[14] Todd K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[15] Lon Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.

[19] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] Andrew Brock, Jeff Donahue, and Karen Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2019.

[21] Hongyang Zhang, Susu Xu, Jiantao Jiao, Pengtao Xie, Ruslan Salakhutdinov, and Eric P. Xing, "Stackelberg GAN: towards provable minimax equilibrium via multi-generator architectures," *CoRR*, vol. abs/1811.08010, 2018.