Few-Shot Conditional GANs with Weakly Supervised Pseudo-Label Refinement

Nicolas Aagnes
Stanford University
ICME
naagnes@stanford.edu

Edward Vendrow
Stanford University
Department of Computer Science
evendrow@stanford.edu

Abstract

Conditional generative adversarial networks (cGANs) are a variation of the GAN architecture which accepts a class label to condition the generator to generate a sample of a particular class. While cGANs require labeled data for training, many public image datasets are unlabelled. To this end, the pseudo-label refinement process concurrently trains a cGAN and classifier, with each creating data for the other, allowing for high quality cGAN training with noisy initial labels. Our work provides a deep analysis of the pseudo-label refinement strategy, meaningfully quantifying the robustness of the method to different levels and types of label noise. Furthermore, we improve on previous works by relaxing the requirement of a small labelled dataset, instead introducing the novel addition of using a one-shot learning method to create the initial noisy dataset. Combinding one-shot learning with pseudo-label refinement, we successfully demonstrate conditional GAN training with just a single labeled example per class. This novel result is an important step toward training conditional GANs without needing large labelled datasets. All source code is available at https://github.com/nicolas-aagnes/few-shot-cgan.

1 Introduction

Conditional generative adversarial networks (cGANs) [8] are a variation of the GAN architecture which accepts a class label so as to condition the generator to generate a sample from the desired class. cGANs require labeled data to train, but labels are often unavailable: state-of-the-art GANs such as StyleGAN [5] are often trained on large, unlabeled image datasets such as FFHQ [6], MetFaces [4], and AFHQ [3]. However, it is often reasonable to find a small, labelled dataset from a similar domain, which can be used to train a classifier that can then be used to generate pseudo-labels for the target dataset. This setting is commonly referred to as Unsupervised Domain Adaptation (UDA), whereby a model is trained to perform well on an unlabelled target domain using labelled data from a source domain.

A major challenge in UDA is learning from noisy labels that arise from the domain shift between a target and source dataset. The noisy labels will inevitably have a classification error that is tied to the classifier accuracy used to generate the pseudo labels, and the distribution, or amount of noisy labels, may vary between classes in the target dataset. The limited knowledge of the pseudo label noise distribution is a serious drawback and limitation of UDA, as it is difficult to know *a priori* if a training algorithm will perform well due to the uncertainty in the amount of noisy labels and their distribution.

To this end, we plan to conduct a systematic study of how the amount and distribution of noise levels affects the training of conditional GANs. Furthermore, we will be exploring a recent method which iteratively trains a cGAN together with a classifier to generate more accurate pseudo labels over the course of training. Previous work has found deep neural net classifiers to be robust to noisy labels [11], and can therefore further improve the accuracy of cGANs if utilized during training.

Our primary goal is to analyze the limits under which a well performing cGAN can be successfully trained when subject to noisy labels. We begin by conducting experiments with artificially injected noise in the training dataset labels to precisely quantify the robustness of the method to label noise. Afterward, we introduce a novel addition to the method by inferring the initial labels using one-shot learning models, demonstrating full conditional GAN training using just a single label per class.

Overall, we find unexpected and surprising results that show well performing cGANs can be trained under extreme conditions. For example, we find that it is possible to train a cGAN on the MNIST dataset to almost 100% accuracy even though more than 95% of the labels were incorrectly labelled. The results of our findings may help future researchers and practitioners gauge the amount of accurate labels that are needed to train well-performing cGANs, which is of particular importance in fields such as Unsupervised Domain Adaptation and in Meta-Learning, where few-shot models are often used to generate labels for large, unlabelled datasets.

2 Related Work

In addition to Unsupervised Domain Adaptation, noisy labels also arise from few-shot models that are used to label an entire dataset given only a few labelled images. The task of few-shot conditional generation has been studied across types of generative models and different problem setups. Sinha et al. [12] introduce a diffusion-decoding VAE which can be used with a classifier trained over the latent space to generate a specific class via rejection sampling. Bartunov et al. [1] introduce the generative matching network, a new type of generative model, which can be conditioned on additional input data to generate specified types of data.

Another source of noisy labels which has gained a lot of traction in recent years stems from meta-learning based approaches. Meta-learning models are trained to quickly adapt to new unseen tasks by utilizing training algorithms that are optimized for "learning to learn". Recent advancements such as the MetaOptNet model [7] have shown superb results on many of the most common meta-learning benchmarks, and can be used to generate pseudo labels for an entire dataset given just a few labelled images per class. For example, the meta-learning model can be given 20, 5 or even just 1 labeled image per class to adapt to, and can be utilized to label the rest of the dataset once its adaptation procedure is complete. These are referred to as 20-shot, 5-shot and 1-shot learning models respectively.

Our experiments largely build on the work of Morerio et al. [9], who use an iterative process of pseudo-label generation, cGAN training, and classifier refinement. Their method heavily relies on two results from previous work: deep learning-based classifiers are robust to uniform noise if trained on sufficiently large datasets [11], and that cGANs are, to a certain extent, robust to structured noise [11]. By iteratively training the cGAN and classifier together, the classifier is able to filter out the uniform noise whilst the cGAN dilutes the structured noise. As the noise levels decrease during training both the classifier and generator benefit from each other and decrease their respective accuracies over the course of training.

Morerio et al. [9] initially introduced their iterative training procedure as a way of showing that cGANs can be accurately trained despite structured noisy labels. They test their framework using UDA between the SVHN, MNIST, and MNIST-M datasets and show promising results. However, they mention that further analysis of their framework under more extreme circumstances and with more challenging datasets as an important area of future work. We utilize this framework to push noisy cGAN training to the limit and explore the boundaries under which it is possible to still train high-performing cGANs. Lastly, we expand on their framework to improve the accuracy of traning a cGAN with pseudo-labels generated from few-shot learning models.

3 Problem Statement

As a baseline, we first use the MNIST dataset as a toy dataset for our experiments, such as analysing the effect of the cGAN's performance for varying degrees of noise and noise distributions. This dataset is ideal for initial experimentation since the small size of the data will make experimenting with training classifiers and cGANs fast and computationally inexpensive. The results from this dataset will validate our methodology by showing that the pseudo-label refinement method works to train a good cGAN.

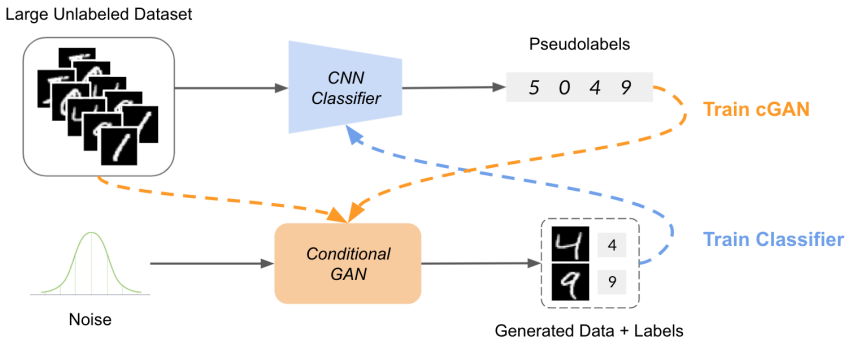

Figure 1: We progressively improve our cGAN label accuracy by iteratively training the cGAN and a label classifier. First, a low-accuracy classifier infers labels for the large unlabeled dataset. Second, the dataset and these pseudolabels are used to train the cGAN, which can now generate more training data for the classifier. This process is repeated to iteratively improve both models.

We plan to study how robustly this method operates with respect to the quantity and degree of noisy labels. Fewer labelled data points will mean a worse initial classifier and thus worse pseudo-labels, making it more difficult for a cGAN to train properly. Although cGANs have been shown to be somewhat robust to noisy labels, high levels of noise, leading to inaccurate conditional generation, may prevent the classifier from improving, rendering the method unsuccessful. Furthermore, we will examine how the method performs with varying degrees of unbalanced noise distribution, where the noisy labels are not uniformly distributed over classes. Lastly, just how certain CNN architectures have shown to be more robust than others to noise, we will experiment with different cGAN architectures to see if some models within this framework are more robust than others for conditional image generation with noisy labels.

We evaluate our method based on the performance of the cGAN generator and classifier. The classifier is evaluated on its prediction accuracy and the generator will evaluted on the accuracy of conditional generation by using an "oracle" classifier (trained on the full, original, labelled dataset) to determine if the desired and generated label is the same. It is important to note that at no stage during the training procedure do we utilize the full set of correct labels, as we only use our artificially created noisy labels for training the classifier and generator. However, the oracle classifiers which are only used for evaluating the models are trained on the correct set of labels.

4 Technical Approach

4.1 Overview

The pseudo-label refinement algorithm iteratively updates the generator and classifier using each other's output. Specifically, the output from the conditional GAN is used as training data for the classifier, while unlabeled images with pseudolabels generated by the classifier are used to train the conditional GAN. Figure 1 shows an illustration of this process. This method requires some initial noisy pseudolabels for the unlabeled data in order to start the procedure, which come from training an initial classifier using just a small labeled dataset or a few-shot model.

The generator is updated using the standard, respective cGAN objective, except that the conditioned classes in the cGAN objective are pseudolabels coming from the classifier C:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\big[\log D(\mathbf{x}|C(\mathbf{x}))\big] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\big[\log(1-D(G(z|C(\mathbf{x}))|C(\mathbf{x})))\big]$$

The classifier is updated via the standard cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{c} y_i \cdot \log \hat{y}_i$$

Algorithm 1 describes this entire training procedure:

Algorithm 1: Iterative pseudo-label refinement

Data: Images X, initial noisy labels $\hat{Y}$
$C \leftarrow$ **Classifier** pretrained with $(X, \hat{Y})$
$G \leftarrow$ **Conditional GAN** pretrained with $(X, \hat{Y})$
for *Batch* $\boldsymbol{x}$ *from* X **do**
  $\mathbf{y} \leftarrow$ Pseudo-labels inferred using $C(\mathbf{x})$
  $G \leftarrow$ Gradient update from $\mathbf{x}, \mathbf{y}$
  $\mathbf{x}', \mathbf{y}' \leftarrow$ Generate from G
  $C \leftarrow$ Gradient update from $\mathbf{x}', \mathbf{y}'$
end

4.2 Artificial Pseudo Label Training

First, we train an oracle classifier on the original MNIST dataset which we only use to evaluate the accuracy of the generator. After this step, we inject noise into the labels by randomly relabelling a certain percentage of the labels. Thus, the original, correct labels are never used from this stage onwards in the training procedure.

We then seperatively pretrain the classifier and cGAN on the noisy dataset. This is a noteworthy distinction to the original framework proposed by Morerio et al., as they utilize a classifier originally pretrained on a source dataset. This separate pretraining step is critical for the algorithm to work correctly because the classifier is trained on generated data and the generator is trained on labels from the classifier, so starting with a randomly initialized classifier and generator would result in non-informative pseudo-labels and noisy images.

After separately pretraining the classifier and cGAN we then iteratively train the two models together as outlined in the technical approach above. Note that in our setup the models are trained in a joint stochastic manner, meaning that in the same loop, for the same batch of images, both the classifier and generator are updated. Thus the classifier and generator are updated together in an online manner, as opposed to repetitively training the classifier and generator separately for their own set of batches.

4.3 Few-Shot Methods for Initial Pseudo Label Generation

Labeled image data may be rare in real-world settings, so we also explore the use of few-shot methods to generate the initial labels, where we only need one or a few labeled examples per class. Since the iterative pseudo-label refinement process is robust to noise, we may use these noisy labels to initialize the process. We therefore also test our method using a few-shot model to label all of the images in the MNIST dataset, given only a few labelled images per class.

To generate our pseudo labels we use MetaOptNet [7], a meta-learning optimization based model which has achieved state of the art results on many of the most common meta-learning benchmarks. MetaOptNet's objective is to learn feature embeddings that generlize well under a linear classification rule for novel categories. Thus, the model is optimized to differentiate between images of unseen categories by updating a linear classifier which has effectively "learned to learn" how to distinguish feature embeddings of different classes.

We take a pretrained MetaOptNet model on the tieredImageNet dataset [2] and utilize it to create a 1-shot and 20-shot model for the MNIST dataset. We achieve this by giving the pretrained model 1 example per class to adapt itself to in the 1-shot learning case, and 20 examples per class in the 20-shot case. The entire dataset can then be labelled using the 1-shot and 20-shot models respectively.

5 Results

5.1 Experimental Setup

We run our initial experiments on the MNIST dataset. The cGAN uses a 128-dimensional latent vector, with a concatenated one-hot class vector representing the digit type, and the classifier is a simple convolutional neural network featuring two convolutional layers.

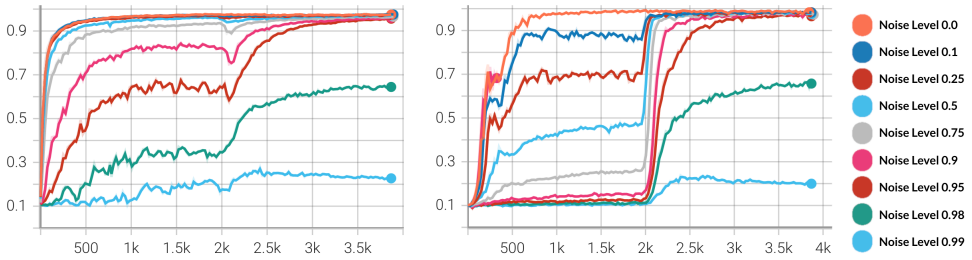

Figure 2: Classifier (left) and conditional generator (right) accuracy on MNIST over the course of training for varying noise levels. We first independently pre-train each network for 2,000 steps, then begin the pseudo-label refinement process. The generator is evaluated using an "oracle" classifier. We observe that this process significantly improves both the generator and classifier performance across noise levels, even recovering from 95% label noise.

In order to accurately evaluate the robustness of our method to different levels of noise, we generate the initial pseudolabels by randomly changing some proportion of training labels. This replicates the effect of having labeled the target distribution with a classifier trained on a source distribution.

We begin by training the classifier and generator independently for 5 epochs (or approximately 2,000 steps) on the noisy data. Afterward, we perform the pseudo-label refinement process where the classifier and generator each create data to train the other. We perform this process for another 5 epochs (or approximately another 2,000 steps). Every 25 step we log the classifier and generator accuracy.

5.2 Robustness to Noise

We evaluate the performance of the pseudo-label refinement method across a variety of initial label noise levels from 0% to 99%, where the noise level denotes the proportion of labels that are assigned random values by a random uniform distribution. This experiment aims to show how the model recovers from high initial noise levels. We evaluate both the classifier accuracy and generated label accuracy as the model trains.

Figure 2 shows the accuracy of the classifier and generated labels over the course of training. We first pre-train each model independently for 2,000 steps on the noisy dataset, then begin pseudo-label refinement. We observe that at the end of pre-training, the cGAN accuracy approximately matches the data noise level, while the classifier accuracy is generally higher. After beginning pseudo-label refinement, all generators markedly improve, with most converging to near perfect accuracy. However, extremely high noise levels (in this case, above 0.95) prevent optimal convergence as expected. This result showcases the robustness of our method to label noise, and also suggests that the classifier's robustness to noise is crucial to optimal convergence.

5.3 Label Noise Distribution

Classification performance depends on the ability of the classifier to distinguish between similar images with different class labels. Since classification accuracy is crucial for pseudo-label refinement, we evaluate this method for different noise distributions, and we use entropy as our measure of how unevenly the noisy labels are distributed amongst classes. For our experiment we fix the noise level at 0.97, since at this value the pseudo-label refinement process improves accuracy but does not converge optimally. Figure 3 shows classifier and generator performance at different entropy levels. There is not an obvious correlation between entropy and performance, suggesting that the actual placement of noisy labels is important. For example, 1s and 5s are easy to tell apart while 1s and 7s are harder, so label noise between more similar images may more significantly affect performance. The proceeding experiment with few-shot labels reinforces this idea.

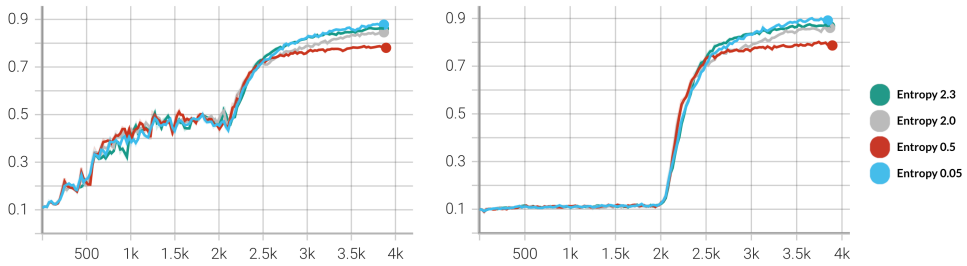

Figure 3: Classifier (left) and conditional generator (right) accuracy on MNIST over the course of training, with pseudo-label refinement beginning 2k steps in, at different noise entropies. We fix the noise level at 0.97, since at this value the pseudo-label refinement process improves accuracy but does not converge optimally. There is not obvious correlation between entropy and performance, suggesting that the placement of noisy labels is important (i.e. 1s and 5s are easy to tell apart, but not 1s and 7s)

.

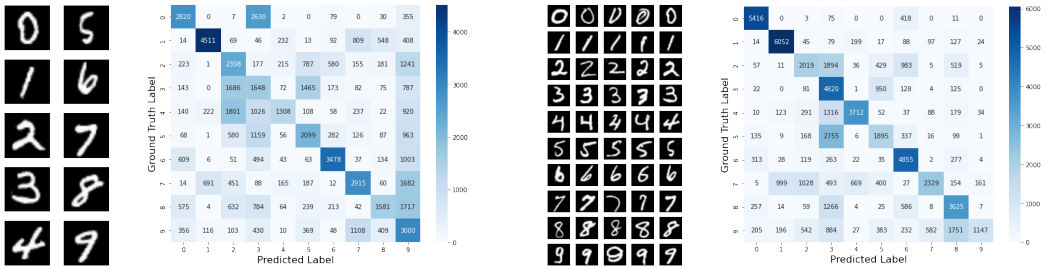

Figure 4: We try 1-shot and 20-shot learning to label the training set. The "known" examples used are chosen randomly and pictured above (a subset for 20-shot), along with the confusion matrices of training set labels for 1-shot (left) and 20-shot (right) prediction. The 1-shot prediction achieves an accuracy of 42.93%, while 5-shot prediction achieves an accuracy of 59.78%.

5.4 Few-Shot Initial Label Generation

While the previous experiments used artificial label noise to precisely control noise level, we now demonstrate the use of few-shot learning to infer the initial labels with as low as a single example per class.

We experiment with both 1-shot and 20-shot learning. We randomly select 1 or 20 digits per class, respectively, and use the few-shot method to label the entire training dataset. We then use these noisy labels to proceed with iterative pseudo-label refinement. Figure 4 shows the few-shot examples used to label the data, as well as the confusion matrix of the training set labels. The confusion matrix reveals that specific digit pairs are harder to tell apart. For instance, there is a high level of confusion for the pairs (3,5), (1,7), and (5,9), which makes sense given the perceptual similarities in these digits. Using just this few-shot prediction method, the 1-shot prediction achieves an accuracy of 42.93%, while 5-shot prediction achieves an accuracy of 59.78% in labelling the training set before proceeding with pseudo-label refinement.

Figure 5 shows the accuracy curve for pseudo-label refinement for the 1-shot and 20-shot method. We observe that both trials achieve a final classifier and generator accuracy near 88%, improving significantly over the initial label accuracy generated by the few-shot learning method.

5.5 Discussion

Our results show the impressive robustness of iterative pseudo-label refinement to large amounts of noise, enabling us to train an accurate conditional GAN with as low as a single example per class with the use of few-shot learning methods.

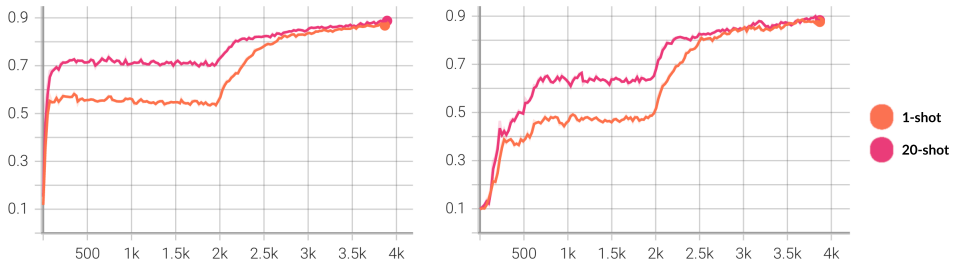

Figure 5: Classifier (left) and generator (right) accuracy with initial labels generated using 1-shot and 20-shot learning, with pseudo-label refinement beginning 2k steps in. We observe that both trials reach a high level of accuracy, with a final classifier and generator accuracies both near 88%.

.

We observe from Figures 2 and 3 that while the generated label accuracy will tend to approximate the noise in the labeled data, the classification accuracy remains high even in the initial stages when label noise is significant. Recent work on evaluating the robustness of deep neural networks to label noise [10] supports this result, showing that classifiers generally have an impressive robustness to uniform label noise. This result suggests that while the cGAN itself may not be robust to label noise, the high quantity of samples it generates to train the classifier are important to improve classification accuracy, and thus conditional GAN label accuracy after finetuning. At the same time, the robustness of the classifier to noise provides high quality image-label pairs to train the cGAN.

In Figure 2 we also observe a small, temporary drop in classifier accuracy right at the start of the pseudo-label refinement stage (at 2,000 steps, approximately halfway through the training process). This drop corresponds to when the classifier begins training using just data generated by the cGAN. At this stage the cGAN generates data with about the same noise level as the initial noisy labels, but perhaps with a slightly lower image quality, thus providing worse quality data to traing the classifier. This then leads to a slight drop in classifier performance until the cGAN accuracy begins to increase.

Our novel use of a few-shot label to provide the initial noisy labels further develops this method by allowing for conditional GAN training with as low as 1 example per class. Figures 4 and 5 show that using few-shot methods provides a good starting point to train high-quality conditional generative models. While the final accuracy (near 88%) is not perfect, we believe that it is a significant achievement given that we are provided just a single example per class. Furthermore, we believe that using better few-shot learning methods will even further improve this accuracy beyond the current capability, possibly achieving near-perfect performance as with our other experiments.

6 Conclusion

Our work provides a deep analysis of the pseudo-label refinement strategy, meaningfully quantifying the robustness of the method to different levels and types of label noise. Furthermore, we improve on previous works by relaxing the requirement of a small labelled dataset [9], demonstrating that using pseudo-label refinement together with few-shot learning methods allows conditional GAN to be trained with as little as a single example per class. We believe that this novel result is an important step toward training conditional GANs without needing large labelled datasets.

7 Future work

Our progress up to this point has shown the effectiveness of this method on a toy dataset, including robustness to varying noise levels and dataset sizes. Now we plan to apply the method to more complex datasets, as well as extend the method with ideas from representation learning.

After validating out method on the MNIST dataset, we will now move on to larger and more complex data distributions. We plan to use the CelebA dataset, which features over 200k face images with 40 attribute labels. The high number of attributes will make the task of conditional generation difficult in general, so by showing that we are able to robustly train a cGAN with noisy attribute labels, we will show how our method works even in difficult scenarios. We additionally plan to evaluate the

generated image quality using standard metrics such as Fréechet Inception Distance and Learned Perceptual Image Patch Similarity (LPIPS) [13].

8 Additional Information

Both authors contributed equally to this project, sharing an equal workload for running experiments and writing the final report.

This project is not being supported by a lab and it is not being shared with other AI classes at Stanford.

References

[1] Sergey Bartunov and Dmitry Vetrov. Few-shot generative modelling with generative matching networks. In *International Conference on Artificial Intelligence and Statistics*, pages 670–678. PMLR, 2018.

[2] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9062–9071, 2021.

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains, 2020.

[4] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020.

[5] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

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

[7] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. *CoRR*, abs/1904.03758, 2019.

[8] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[9] Pietro Morerio, Riccardo Volpi, Ruggero Ragonesi, and Vittorio Murino. Generative pseudo-label refinement for unsupervised domain adaptation, 2020.

[10] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

[11] David Rolnick, Andreas Veit, Serge J. Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *CoRR*, abs/1705.10694, 2017.

[12] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2c: Diffusion-denoising models for few-shot conditional generation, 2021.

[13] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.