Data-Efficient Learning in Image Synthesis and Instance Segmentation

Esther A. Robb

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Jia-Bin Huang, Chair
Ruoxi Jia
Hoda Eldardiry

July 2, 2021
Blacksburg, Virginia

Data-Efficient Learning in Image Synthesis and Instance Segmentation

Esther A. Robb

(ABSTRACT)

Modern deep learning methods have achieve remarkable performance on a variety of computer vision tasks, but frequently require large, well-balanced training datasets to achieve high-quality results. Data-efficient performance is critical for downstream tasks such as automated driving or facial recognition. We propose two methods of data-efficient learning for the tasks of image synthesis and instance segmentation. We first propose a method of high-quality and diverse image generation from finetuning to only 5-100 images. Our method factors a pretrained model into a small but highly expressive weight space for finetuning, which discourages overfitting in a small training set. We validate our method in a challenging few-shot setting of 5-100 images in the target domain. We show that our method has significant visual quality gains compared with existing GAN adaptation methods. Next, we introduce a simple adaptive instance segmentation loss which achieves state-of-the-art results on the LVIS dataset. We demonstrate that the rare categories are heavily suppressed by *correct background predictions*, which reduces the probability for all foreground categories with equal weight. Due to the relative infrequency of rare categories, this leads to an imbalance that biases towards predicting more frequent categories. Based on this insight, we develop DropLoss – a novel adaptive loss to compensate for this imbalance without a trade-off between rare and frequent categories.

Data-Efficient Learning in Image Synthesis and Instance Segmentation

Esther A. Robb

(GENERAL AUDIENCE ABSTRACT)

Many of the impressive results seen in modern computer vision rely on learning patterns from huge datasets of images, but these datasets may be expensive or difficult to collect. Many applications of computer vision need to learn from a very small number of examples, such as learning to recognize an unusual traffic event and behave safely in a self-driving car. In this thesis we propose two methods of learning from only a few examples. Our first method generates novel, high-quality and diverse images using a model fine-tuned on only 5-100 images. We start with an image generation model that was trained a much larger image set (70K images), and adapts it to a smaller image set (5-100 images). We selectively train only part of the network to encourage diversity and prevent memorization. Our second method focuses on the instance segmentation setting, where the model predicts (1) what objects occur in an image and (2) their exact outline in the image. This setting commonly suffers from long-tail distributions, where some of the known objects occur frequently (e.g. "human" may occur 1000+ times) but most only occur a few times (e.g. "cake" or "parrot" may only occur 10 times). We observed that the "background" label has a disproportionate effect of suppressing the rare object labels. We use this to develop a method to balance suppression from background classes during training.

Dedication

This work is dedicated to my family, thank you to my parents for supporting me in anything I want to do and teaching me to value education and asking questions. Thank you to my older brother and sister, who have been my closets friends and unconditional supporters my entire life.

Acknowledgments

I would like to thank my advisor, Jia-Bin Huang, for all of his guidance and support in the last few years. Thank you for your clarity and good judgement in helping with important decisions, and for inspiring me to try for many new opportunities. I would also like to thank my intern mentors for supporting and guiding me in my summer projects and beyond, and everyone in the VT Computer Vision lab for many good discussions and support.

Contents

List of Figures ix

List of Tables xiv

1 Introduction 1

2 Few-shot Adaptation for Generative Adversarial Networks 2

2.1 Introduction . . . 2

2.2 Related Work . . . 4

2.2.1 Generative Adversarial Networks (GANs) . . . 4

2.2.2 Sample-efficient Image Synthesis . . . 5

2.2.3 One-shot Image Re-synthesis. . . . 5

2.2.4 Singular Value Decomposition (SVD). . . . 6

2.3 Approach . . . 7

2.3.1 Overview . . . 7

2.3.2 Adaptation Procedure . . . 8

2.3.3 Training & Inference . . . 10

2.3.4 Evaluation in Few-Shot Synthesis . . . 10

2.4 Results . . . 12

2.4.1 Settings . . . 12
2.4.2 Near-domain adaptation . . . 14
2.4.3 Far-domain adaptation . . . 14
2.4.4 N-shot Settings . . . 15
2.5 Conclusions . . . 16

3 DropLoss for Long-Tail Instance Segmentation 19

3.1 Introduction . . . 19
3.2 Related Work . . . 22
3.2.1 Object Detection and Instance Segmentation. . . . 22
3.2.2 Learning Long-tailed Distributions. . . . 22
3.2.3 Resampling Methods. . . . 22
3.2.4 Reweighting and Cost-sensitive Methods. . . . 23
3.2.5 Feature Manipulation Methods. . . . 23
3.2.6 Long-tail Learning Settings. . . . 24
3.3 Approach . . . 25
3.3.1 Revisiting the Equalization Loss . . . 25
3.3.2 Equalization Loss. . . . 25
3.3.3 Background Equalization Loss. . . . 27
3.3.4 DropLoss . . . 29

3.4 Results . . . 31
3.4.1 Dataset. . . . 32
3.4.2 Implementation Details. . . . 32
3.4.3 Incorporating with Resampling Methods. . . . 34
3.4.4 Measuring the Frequent-rare Category Performance Tradeoff. . . . 35
3.4.5 Quantitative Comparison Between the DropLoss and Our Proposed Baseline BEQL. . . . 36
3.4.6 Visual Results. . . . 36
3.4.7 Ablation Study . . . 37
3.5 Conclusions . . . 39

4 Summary **40**

Bibliography **41**

List of Figures

2.1 **Few-shot image generation.** Our method generates novel and high-quality samples in a new domain with a small amount of training data. (*Top*) Diverse random samples from adapting a FFHQ-pretrained StyleGAN2 to toddler images from the CelebA dataset (with **only 30 images**) using our method. (*Bottom*) Smooth latent space interpolation between two random seeds shows that our method produces novel samples instead of simply memorizing the 30 images. Please see the supplementary video for more results. 3

2.2 **Comparing methods for GAN adaption.** Learnable parameters are denoted in red. **(a)** TransferGAN (TGAN for simplicity) [72] and FreezeD [53] retrain all weights W in a layer. SSGAN [54] and FSGAN train significantly fewer parameters per layer. Note FSGAN adapts both conv and FC layers, while SSGAN adapts only conv layers. *#params* is the number of learnable paramaters per conv layer; *Count* gives parameter counts over the full StyleGAN2 generator and discriminator. **(b)** FSGAN (ours) adapts singular values $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$ of pretrained weights W_0 to obtain adapted weights W_Σ. 7

2.3 **Effects of singular values.** We visualize FSGAN's adaptation space by magnifying the top 3 singular values $\sigma_0, \sigma_1, \sigma_2$ from SVD performed on style and conv layers of a StyleGAN2 [34, 36] pretrained on FFHQ. In mapping layer 4 (style_4), the leading σs change the age, skin tone, and head pose. In synthesis layer 2 ($\text{conv}_{8\times8}$), face dimensions are modified in term of face height/size/width. In synthesis layer 9 ($\text{conv}_{1024\times1024}$), the face appearance changes in finer pixel stats such as saturation, contrast, and color balance. . 9

2.4 **Problem with FID as a few-shot metric.** TGAN [72] adaptation from English characters to 10-shot Kannada characters (*Bottom*) [8]. The adaptation process is illustrated by interpolating two random latent vectors at different timesteps (t=20 means 20K images seen during training). We measure FID against a 2K-image Kannada set, from which the 10 images was sampled. The interpolation shows larger timesteps (t) tend to memorize the 10-image training set while yielding lower FID, revealing that FID favors overfitting and is not suitable for the few-shot setting. 10

2.5 **Close-domain adaptation** (FFHQ→CelebA). Models adapted from a pretrained StyleGAN2 using ∼30 target images (left-most column) of **(a)** CelebA ID 4978 and **(b)** CelebA ID 3719. The proposed FSGAN generates more natural face images without noticeable artifacts. Comparison methods include TGAN [72], FD [53], SSGAN [54], trained with a limited number of timesteps to prevent overfitting or quality degradation. 13

2.6 **Far-domain adaptation** (Photo→Art). Comparing FSGAN with alternative GAN adaptation methods in the photo-to-art setting. **(a)** FSGAN more effectively alters building layouts and adds landscape in the foreground to match the Van Gogh paintings, maintaining better spatial coherency. **(b)** FSGAN adopts features from the Portraits dataset (hats, beards, artistic backgrounds), while other methods primarily alter image textures. **(c)** FSGAN transforms natural hair and facial features to imitate the anime target while retaining spatial consistency. Note the occurrence of pink hair in our generated images, which does not exist in the few-shot target but is visually consistent. 17

2.7 **N-shot settings** (FFHQ→Portraits): **(a)** Mo et al. [53] with limited timesteps preserves diversity at all n-shots, but produces undesired artifacts and limited adaptation (*e.g.* sunglasses remain). **(b)** Mo et al. [53] with increased timesteps produces quality adaptation with 100 shots, but degenerates at ≤50 shots. **(c)** FSGAN (ours) is robust to n-shot settings, producing high-quality adaptation even at N=5. **(d)** Pretrained FFHQ images. 18

3.1 Motivation. (a) Percentage of gradient updates from incorrect foreground classification (blue) and ground-truth background anchors (orange) on LVIS [16]. We divide the categories into "frequent" (white shading), "common" (orange shading), and "rare" (yellow shading). For rare categories, background gradients occupy a disproportionate percentage of total gradients. (b) The distribution of average foreground class prediction scores for ground-truth background bounding boxes at earlier (red) and later (blue) training stages. We find that, for background bounding boxes, the prediction scores of rare categories are more severely suppressed, and the training is biased towards predicting more frequent categories. 20

3.2 Background equalization loss. We present an extension of the equalization loss that specifically focuses on the background classification. (a) The curves of the $\mathcal{L}_{\text{BEQL}}$ weights in (3.4) with different choices for the logarithm base b. Smaller values of the logarithm base b reduce the effects of background more. (b) The experimental result of applying different logarithm bases shows a tradeoff between the mean average precision (mAP) of rare categories and the mAP of frequent categories with respect to different logarithm base settings. Note that the background equalization loss $\mathcal{L}_{\text{BEQL}}$ with a large base b reduces to the existing equalization loss $\mathcal{L}_{\text{EQL}}$. 26

3.3 Visual results comparison. Qualitative results of the Mask R-CNN trained with standard cross-entropy loss (*Top*) and the proposed DropLoss (*Bottom*). Instances with scores larger than 0.5 are shown. 34

3.4 Comparison between rare, common, and frequent categories AP for baselines and our method. We visualize the trade-off for 'rare vs. frequent' and 'common vs. frequent' as a Pareto frontier, where the top-right position indicates an ideal trade-off between objectives. DropLoss achieves an improved trade-off between object categories, resulting in higher overall AP. 37

3.5 Qualitative results of a Mask R-CNN baseline and b the proposed DropLoss. Instances with score > 0.5 are shown. DropLoss adaptively removes background proposal losses of rare and common categories to reduce bias towards misclassifying these objects as background. In this case, the *common* category 'goose' is misclassified as background in a, and correctly identified in b. . . 38

List of Tables

2.1 **Quantitative comparisons** in three metrics: FID [26], Face Quality Index (FQI) [24], and sharpness [38]. See Fig 2.5 for illustrations. FQI and Sharpness are evaluated on 1,000 images randomly generated with the same set of seeds. Bracketed/bold numbers indicated the best/second best results, respectively. 13

3.1 Comparison between architecture and backbone settings, evaluated on LVIS v0.5 validation set. We compare BCE (binary cross-entropy), EQL (equalization loss) and Drop (DropLoss). AP/AR refers to mask AP/AR, and subscripts 'r', 'c', and 'f' refer to rare, common, and frequent categories. . . . 29

3.2 Evaluation on LVIS v0.5 (top) and LVIS v1.0 (bottom) validation sets *with* and *without* Repeat Factor Sampling (RFS). Here we use Mask-RCNN and ResNet-50. DropLoss achieves the best overall AP across both settings. . . 33

3.3 Comparison between DropLoss and two top-performing results from our another proposed method BEQL. Evaluation on LVIS v0.5 validation set. DropLoss offers overall better performance (in both AP and AR) compared with the BEQL baseline. 36

Chapter 1

Introduction

Performance on rare modes is critical in downstream applications of computer vision and deep learning such as automated driving or facial recognition. However, many of the impressive recent advances in computer vision depend on large, diverse, and well-balanced datasets. Collecting large datasets may be exceedingly expensive or impossible in practice, which limits real-world applications. Therefore data-efficient approaches are essential to make these techniques practical and safe in real-world applications. In this thesis we present data-efficient approaches to learning in image generation and instance segmentation.

Few-shot Generative Adversarial Networks (Chapter 2) presents a method of low-shot generative adversarial model (GAN) finetuning. Our method factors a pretrained weight space into a small, semantically meaningful set of weights. Limiting the weight space allows for fine-tuning to very small datasets while discouraging overfitting or quality degredation.

DropLoss for Long-Tail Instance Segmentation (Chapter 3) introduces a method of adaptively rebalancing losses based on class distribution for instance segmentation. We observe that during training, rare class predictions are suppressed disproportionately by *correct background class predictions* from the model. Because the background class makes up a majority of bounding boxes during training, this has a significant effect on long-tail performance. We introduce an adaptive method to rebalance the background losses based on the batch statistics. Our method DropLoss achieves state-of-the-art performance on the LVIS dataset.

Chapter 2

Few-shot Adaptation for Generative Adversarial Networks

2.1 Introduction

Recent years have witnessed rapid progress in Generative Adversarial Networks (GAN) [14] with improvements in architecture designs [33, 34, 57, 79], training techniques [33, 52, 63], and loss functions [1, 15]. Training these models, however, typically requires large, diverse datasets in a target visual domain. While there have been significant advancements in improving training stability [33, 52], adversarial optimization remains challenging because the optimal solutions lie at saddle points rather than a minimum of a loss function [75]. Additionally, GAN-based models may suffer from the inadequate generation of rare modes in the training data because they optimize a mode-seeking loss rather than the mode-covering loss of standard likelihood maximization [55]. These difficulties of training GANs become even more severe when the number of training examples is scarce. In the low-data regime (e.g., less than 1,000 samples), GANs frequently suffer from memorization or instability, leading to a lack of diversity or poor visual quality.

Several recent efforts have been devoted to improving the sample efficiency of GANs through transfer learning. The most straightforward approaches are finetuning a pre-trained generator and discriminator on the samples in the target domain [53, 72]. When the number of

training examples is severely limited, however, finetuning the network weights often leads to poor results, particularly when the source and target domains are distant. Instead of finetuning the entire network weights, the method in [54] focuses on adapting batch norm statistics, constraining the optimization problem to a smaller set of parameters. The authors report that this method achieves better results using MLE-based optimization but fails for GAN-based optimization. Although their quality outperforms GAN-based methods in the low-shot setting, the images are blurry and lack details due to maximum likelihood optimization. Invertible flow-based models have shown promising results in data-efficient adaptation [10], but require compute- and memory-intensive architectures with high-dimensional latent spaces.

Figure 2.1: **Few-shot image generation.** Our method generates novel and high-quality samples in a new domain with a small amount of training data. (*Top*) Diverse random samples from adapting a FFHQ-pretrained StyleGAN2 to toddler images from the CelebA dataset (with **only 30 images**) using our method. (*Bottom*) Smooth latent space interpolation between two random seeds shows that our method produces novel samples instead of simply memorizing the 30 images. Please see the supplementary video for more results.

In this paper, we propose a method for adapting a pre-trained GAN to generate novel, high-quality sample images with a small number of training images from a new target domain (Figure 2.1). To accomplish this, we restrict the space of trainable parameters to a small number of highly-expressive parameters that modulate orthogonal features of the pre-trained weight space. Our method first applies singular value decomposition (SVD) to the network weights of a pretrained GAN (generator + discriminator). We then adapts the singular

values using GAN optimization on the target few-shot domain, with *fixed* left/right singular vectors. We show that varying singular values in the weight space corresponds to semantically meaningful changes of the synthesized image while preserving natural structure. Compared with methods that finetune all weights of the GAN [72], individual layers [53], or only adapt batch norm statistics [54], our method demonstrates higher image quality after adaptation. We additionally highlight problems with the standard evaluation practice in the low-shot GAN setting.

2.2 Related Work

2.2.1 Generative Adversarial Networks (GANs)

GANs [14] use adversarial training to learn a mapping of random noise to the distribution of an image dataset, allowing for sampling of novel images. GANs optimize a competitive objective where a generator $G(Z)$ maximizes the classification error of a discriminator $D(X)$ trained to distinguish real data $p(X)$ from fake data $G(Z)$. The GAN [14] objective is expressed formally as:

$$\max_G \min_D \mathbb{E}_{x\sim p(X)}[\log\ D(x)] - \mathbb{E}_{x\sim G(X)}[1 - \log\ D(x)] \tag{2.1}$$

Recent research reformulated this objective to address instability problems [1, 15, 25]. Improved architecture and training has led to remarkable performance in synthesis [3, 36]. Compared to pixel-reconstruction losses [2, 27, 37] GANs typically produce sharper images, although strong priors over the latent space can offer competitive quality [58]. A high-quality generation has relied on large datasets of high-quality images (10K+) that may be expensive or infeasible to collect in many scenarios. Additionally, GANs can suffer from a lack of

diversity, even when large training sets are used because the objective does not penalize the absence of outlier modes [55]. Data-efficient GAN methods are, therefore, of great utility.

2.2.2 Sample-efficient Image Synthesis

Sample-efficient image synthesis methods encourage diverse and high-quality generation in the low-data regime, most commonly through pretraining [54, 72] or simultaneous training [76] on large image datasets. The main differences among these methods lie in the choice of learnable parameters used for adaptation. Examples include adapting all weights of the generator and discriminator Wang et al. [72], freezing only lower layers of the discriminator [53], or changing only channel-wise batch statistics Noguchi and Harada [54]. Flow-based methods [10] show promising results in few-shot adaptation, but their architecture is compute- and memory-intensive and requires latent space of the same dimensionality as the data. Our method uses a *smaller but more expressive set of parameters* (Figure 2.2), resulting in more natural adapted samples.

2.2.3 One-shot Image Re-synthesis.

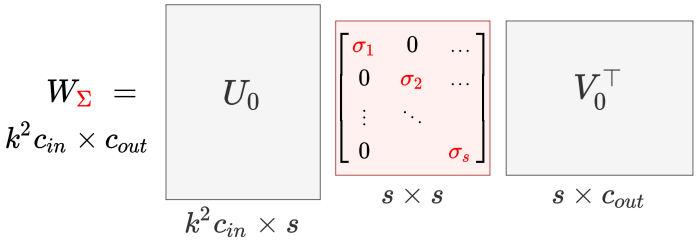
Recent work in one-shot image synthesis has demonstrated high-quality and diverse results by modeling the *internal* distribution of features from a single image without pretraining [66, 68]. Our work differs as we transfer *external* knowledge from a pretrained GAN to a new domain and, therefore, can generate drastically more diverse samples.

2.2.4 Singular Value Decomposition (SVD).

SVD factorizes any matrix $M \in \mathbb{R}^{m \times n}$ into unitary matrices $U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{n \times n}$ and diagonal matrix Σ such that $M = U\Sigma V^\top$, where U, V contain the left and right singular vectors respectively and Σ contains the singular values along the diagonal entries. SVD can be interpreted as a decomposition of a linear transformation $x \to Mx$ into three separate transformations: a rotation/reflection U, followed by rescaling Σ, followed be another rotation/reflection $V^\top$. The transformation defined by the maximum singular value $\sigma_0 = \Sigma^{(1,1)}$ and its corresponding normalized singular vectors represent the maximal axis of variation in the matrix M. This interpretation is commonly used in data science for dimensionality reduction via PCA [39]. PCA can be obtained via SVD on a column-normalized matrix [13]. SVD is also used for a wide number of other applications, including regularization [65], and quantification of entanglement [51], and has also been used to build theoretical background for semantic development in neural networks [64]. The work most closely related to ours is GANSpace [18] for image synthesis editing. GANSpace applies PCA within the *latent feature* space of a pretrained GAN to discover semantically-interpretable directions for image editing in the latent space. In contrast, our work performs SVD on the *weight* space of a GAN to discover meaningful directions for domain adaptation. Performing SVD on the weight space enables two critical differences between our work and Härkönen et al. [18]: (i) we edit the entire output *distribution* rather than one image, and (ii) rather than manual editing, we adapt the GAN to a new domain.

Method	Conv layer	#params	Count
Pretrain	$conv(x, W_0)$	–	–
TGAN	$conv(x, W)$	$k^2 c_{\text{in}} c_{\text{out}}$	59M
FreezeD	$conv(x, W)$	$k^2 c_{\text{in}} c_{\text{out}}$	47M
SSGAN	$conv(x, W_0) \cdot \gamma + \beta$	$2c_{\text{out}}$	23K
FSGAN (Ours)	$conv(x, W_\Sigma)$	c_{out}	16K

(a) Adaptation method formulations.



(b) FSGAN singular value adaptation.

Figure 2.2: **Comparing methods for GAN adaption.** Learnable parameters are denoted in red. **(a)** TransferGAN (TGAN for simplicity) [72] and FreezeD [53] retrain all weights W in a layer. SSGAN [54] and FSGAN train significantly fewer parameters per layer. Note FSGAN adapts both conv and FC layers, while SSGAN adapts only conv layers. *#params* is the number of learnable paramaters per conv layer; *Count* gives parameter counts over the full StyleGAN2 generator and discriminator. **(b)** FSGAN (ours) adapts singular values $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$ of pretrained weights W_0 to obtain adapted weights W_Σ.

2.3 Approach

2.3.1 Overview

Our goal is to improve GAN finetuning on small image domains by discovering a more effective and constrained parameter space for adapting the pretrained weights. We are inspired by prior work in GAN adaptation showing that constraining the space of trainable parameters can lead to improved performance on target domain [53, 54, 59]. In contrast to identifying the parameter space within the model architecture, we propose to discover a parameter space based on the pretrained weights. Specifically, we apply singular value decomposition to the pretrained weights and uncover a basis representing orthogonal directions of maximum variance in the weight space. To explore the interpretation of the SVD representation, we visualize the top three singular values of synthesis and style layers of StyleGAN2 [36]. We

observe that varying the singular values corresponds to natural and semantically-meaningful changes in the output image as shown in Figure 2.3. Changing the singular values can be interpreted as changing the entanglement between orthogonal factors of variation in the data (singular vectors), providing an expressive parameterization of the pretrained weights, which we leverage for adaptation as described in the following section.

2.3.2 Adaptation Procedure

Our method first performs SVD on both the generator and discriminator of a pretrained GAN and adapts the singular values to a new domain using standard GAN training objectives. A generator layer $G^{(\ell)}$ or a discriminator layer $D^{(\ell)}$ may consist of either 2D ($c_{\text{in}} \times c_{\text{out}}$) fully-connected weights or 4D ($k \times k \times c_{\text{in}} \times c_{\text{out}}$) convolutional filter weights. We apply SVD separately at every layer of the generator $G^{(\ell)}$ and discriminator $D^{(\ell)}$. Next, we describe the decomposition process for a single layer of pretrained weights $W_0^{(\ell)}$. For fully-connected layer $W_0^{(\ell)}$, we can apply SVD directly on the weight matrix. For 4D convolution weights $W_0^{(\ell)} \in \mathbb{R}^{k \times k \times c_{in} \times c_{out}}$ this is not feasible because SVD operates only on a 2D matrix. We therefore reshape the 4D tensor by flattening across the spatial and input feature channels before performing SVD to obtain a 2D matrix $W_0^{(\ell)} \in \mathbb{R}^{k^2 c_{in} \times c_{out}}$. Our intuition is that the spatial-feature relationship in the pretrained model should be preserved during the adaptation. We apply SVD over each set of flattened convolutional weights or fully convolution weights to obtain the decomposition:

$$W_0^{(\ell)} = (U_0 \, \Sigma_0 \, V_0^\top)^{(\ell)}. \tag{2.2}$$

After decomposing the pretrained weights, we perform domain adaptation by freezing pre-trained left/right singular vectors in $(U_0, V_0)^{(\ell)}$ and optimizing the singular values $\Sigma = \lambda \Sigma_0$

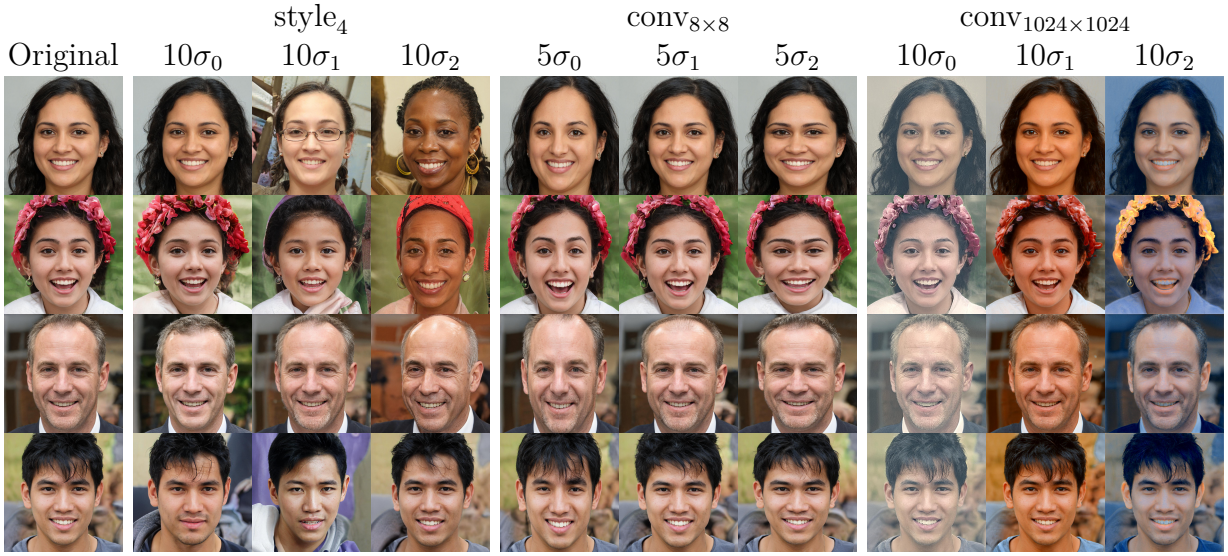


Figure 2.3: **Effects of singular values.** We visualize FSGAN's adaptation space by magnifying the top 3 singular values $\sigma_0, \sigma_1, \sigma_2$ from SVD performed on style and conv layers of a StyleGAN2 [34, 36] pretrained on FFHQ. In mapping layer 4 (style$_4$), the leading σs change the age, skin tone, and head pose. In synthesis layer 2 (conv$_{8\times8}$), face dimensions are modified in term of face height/size/width. In synthesis layer 9 (conv$_{1024\times1024}$), the face appearance changes in finer pixel stats such as saturation, contrast, and color balance.

using a standard GAN objective to obtain transferred weights (Figure 2.2):

$$W_{\Sigma}^{(\ell)} = (U_0 \, \Sigma \, V_0^{\top})^{(\ell)} \tag{2.3}$$

Effectively, our GAN domain adaptation aims to find a new set of singular values in each layer of a pretrained model so that the generated outputs match the distribution of the target domain.

During forward propagation, we reconstruct weights W_{Σ} using the finetuned singular values at each convolution or fully-connected layer of the generator and discriminator before applying the operation.

t	FID	Interpolation
0	121.21	
20	154.25	
40	134.22	
80	102.87	
120	93.65	
180	**92.94**	
Train set (10-shot):		

Figure 2.4: **Problem with FID as a few-shot metric.** TGAN [72] adaptation from English characters to 10-shot Kannada characters (*Bottom*) [8]. The adaptation process is illustrated by interpolating two random latent vectors at different timesteps (t=20 means 20K images seen during training). We measure FID against a 2K-image Kannada set, from which the 10 images was sampled. The interpolation shows larger timesteps (t) tend to memorize the 10-image training set while yielding lower FID, revealing that FID favors overfitting and is not suitable for the few-shot setting.

2.3.3 Training & Inference

Our experiments use the StyleGAN2 [36] training framework, which optimizes a logistic GAN loss (Equation 2.1) with latent space gradient regularization and a discriminator gradient penalty. We retrain the singular values Σ for a fixed number of timesteps (20K images or 16K for 5-shot). We find limiting the training time is essential for quality and diversity in the low-shot setting, as longer training often leads to overfitting or quality degradation (examples in Figure 2.4 & 2.7). Like Noguchi and Harada [54], we use the truncation trick [3] during inference, but our method works with a less-restrictive truncation parameter of $\psi = 0.8$, which enables more diversity in the generated images.

2.3.4 Evaluation in Few-Shot Synthesis

A common adverse outcome in few-shot image generation is overfitting to the target set, such that all generated images look similar to the training data. Evaluation metrics should reflect the diversity of generated images, so that memorization is penalized. The standard evaluation practice used in prior low-shot GAN adaptation work [53, 54, 72] is to estimate FID [26] using a large held-out *test set* with 1K+ images, from which the low-shot *training*

set was sampled. Standard GAN evaluation typically measures FID with respect to the *training set*, but in the low-shot setting, this is not desirable because the generator may simply memorize the training set. However, we find that even when measuring FID against a held-out test set, this evaluation still favors overfitted or poor-quality models, as shown in Figure 2.4. FID between real and fake images is calculated as the Frechet distance between perceptual features $p_r(X)$ and $p_f(Z)$:

$$||\mu_r - \mu_f||^2 + \text{Tr}(C_r + C_f - 2\sqrt{C_r C_f}). \tag{2.4}$$

where it is assumed features are Gaussian *i.e.* $p_f(Z) = N(\mu_f, C_f)$ and $p_r(X) = N(\mu_r, C_r)$. In the few-shot setting, our n-shot training set $T = (x_1, x_2, ..., x_n)$ is sampled from our test set $p_r(X)$. Assuming T is chosen at random, its sample mean and variance $\hat{\mu}, \hat{\sigma^2}$ are unbiased estimators of μ_r, C_r. Therefore if the generator *memorizes* T, its statistics approximate μ_r, C_r. This artificially decreases the FID of an overfit model (Figure 2.4). Consequently, we suggest that FID should be supplemented with additional metrics and extensive qualitative results in the low-shot setting. In high-data settings, a very large number of parameters would be required to memorize the images, so this problem is less likely to occur. Based on these observations, throughout our evaluation, we limit training timesteps rather than select the step with the best FID as we find the latter approach gives more inferior qualitative results. To address the limitations of standard metrics for GAN evaluation, we also report sharpness [38] and face quality index [24] for human face transfer.

2.4 Results

2.4.1 Settings

We adapt a pretrained model to a new target domain using only 5-100 target images, as we focus on scenarios with 1-2 orders fewer number of training samples than standard data-efficient GAN adaptation methods [53, 54, 72]. As discussed in Section 2.3.4, we find that the FID score is unsuitable in the low-shot regime due to overfitting bias. However, we still report the FID scores of our experiments for completeness. In addition, we report additional quality metrics and extensive qualitative results.

Adaptation Methods. We compare the proposed FSGAN with Transfer GAN (TGAN) [72], FreezeD (FD) [53], and the Scale & Shift GAN (SSGAN) baseline of Noguchi and Harada [54]. For a fair comparison in the GAN setting, we choose the GAN baseline of SSGAN [54] instead of their GLO-based variant. We implement all methods using the StyleGAN2 [36] codebase.[1] We follow the training setting of StyleGAN, but change the learning rate to 0.003 to stabilize training and reduce the number of training steps to prevent overfitting in the low-shot setting. Figures 2.4, 2.7 show comparisons of different training times.

[1] https://github.com/NVlabs/stylegan2

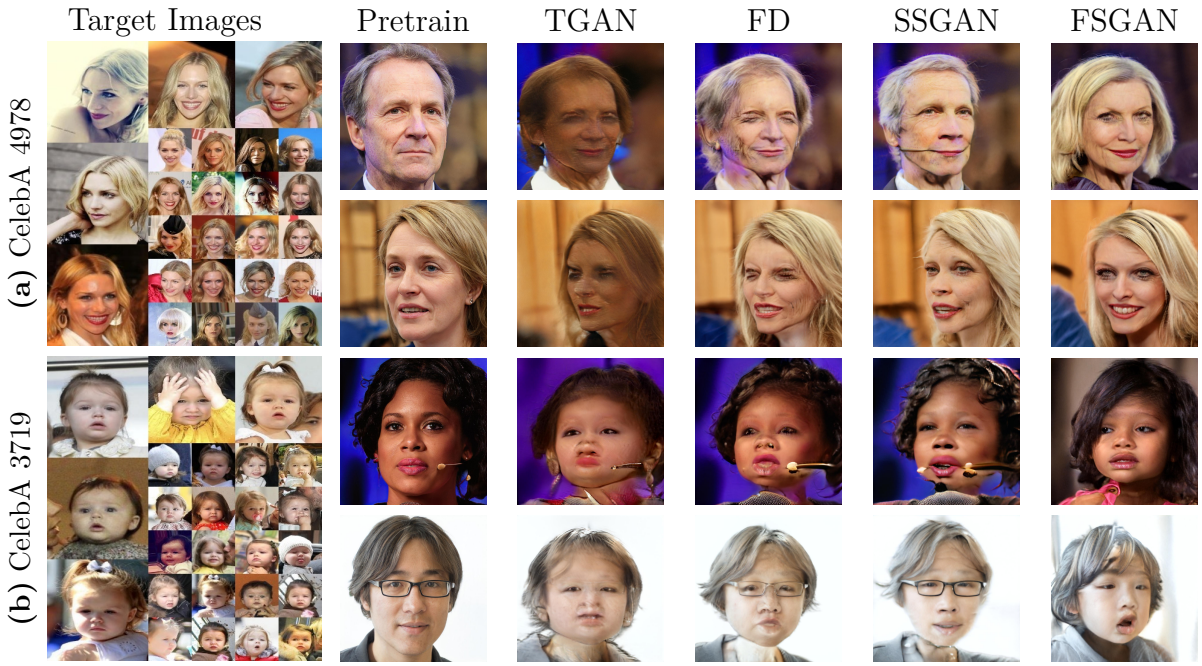


Figure 2.5: **Close-domain adaptation** (FFHQ→CelebA). Models adapted from a pre-trained StyleGAN2 using ∼30 target images (left-most column) of **(a)** CelebA ID 4978 and **(b)** CelebA ID 3719. The proposed FSGAN generates more natural face images without noticeable artifacts. Comparison methods include TGAN [72], FD [53], SSGAN [54], trained with a limited number of timesteps to prevent overfitting or quality degradation.

Table 2.1: **Quantitative comparisons** in three metrics: FID [26], Face Quality Index (FQI) [24], and sharpness [38]. See Fig 2.5 for illustrations. FQI and Sharpness are evaluated on 1,000 images randomly generated with the same set of seeds. Bracketed/bold numbers indicated the best/second best results, respectively.

	CelebA 4978			CelebA 3719		
Method	FID	FQI	Sharpness	FID	FQI	Sharpness
Pretrain	–	0.40±0.11	0.91±0.06	–	0.37±0.12	0.92±0.06
TransferGAN	75.41	0.30±0.07	0.61±0.05	178.31	0.26±0.09	**0.61±0.04**
FreezeD	75.30	**0.33±0.09**	0.58±0.04	143.83	**0.27±0.09**	0.56±0.05
SSGAN	87.79	0.32±0.08	**[0.67±0.05]**	147.14	0.27±0.10	0.58±0.05
FSGAN (ours)	78.90	**[0.36±0.07]**	**0.65±0.05**	170.00	**0.27±0.08**	**[0.68±0.07]**

Datasets. We used FFHQ [34] and LSUN Churches [78] pretrained checkpoints from StyleGAN2 [35], and transferred to few-shot single-ID CelebA (30 or 31 images) [48], Portraits (5-100 images) [40], Anime ID "Rem" (25 images) [2], and Van Gogh landscapes (25 images) [81]. We evaluate FID against a large test set (10K for CelebA) following the evaluation method of Wang et al. [72]. We also evaluate face quality index [24] and image sharpness [38] for face domain adaptation, using 1000 images from each method generated using identical seeds. Full few-shot target sets are shown in Figures 2.5 & 2.6, and we will make all few-shot sets available online.

2.4.2 Near-domain adaptation

We first show a *near domain* transfer setting (adapting FFHQ to single-ID CelebA datatset [48]). As both source and target domains contain faces, the pretrained model has useful features for the transfer domain. Figure 2.5 shows that existing GAN adaptation methods produce artifacts around the eyes/chin and low overall structural consistency. In contrast, our method generates more natural face images with characteristics similar to the training samples (e.g., the head size, position of the faces). Comparing Figure 2.5 and Table 2.1 shows that the FID correlates poorly with qualitative evaluation for this setting. In light of this, we report additional metrics of face quality [24] and sharpness [38]. On these metrics, our method achieves competitive performance across adaptation settings.

2.4.3 Far-domain adaptation

We show *far-domain* 25-shot transfer, where we define "far" as differing significantly in the distribution of image features such as textures, proportions, and semantics. 1) *LSUN*

[2] https://www.gwern.net/Danbooru2019

Churches→ Van Gogh paintings: The two domains differ in the foreground, building shapes, and textural styles. 2) *FFHQ→Art portraits*: The main differences between the two domains are low-level styles and facial features. 3) *FFHQ→Anime Rem ID*: A challenging setting with exaggerated facial proportions and lack of texture details. Figure 2.6 shows visual comparisons with three state-of-the-art methods. We find that the proposed FSGAN can adapt the model to produce more dramatic changes to match the target distributions in terms of semantics, proportions, and textures while maintaining high image quality.

2.4.4 N-shot Settings

We test the sensitivity of both FSGAN (ours) and FreezeD [53] to differing n-shot settings and show the results in Figure 2.7. We find that FSGAN is more robust to n-shot setting compared to FreezeD. To show this better, we compare two variations of FreezeD. The first FreezeD variant (FD) is limited in timesteps (20K images / 16K on 5-shot) to match FSGAN and the results reported in Figures 2.5 & 2.6. Limiting timesteps prevents degradation that occurs at later iterations in the few-shot settings. However, the time-limited FD produces low quality and limited adaptation of textures and semantic features. The second FreezeD variant (FD-FT) is trained for longer (60K images) to demonstrate (1) degradation in fewer n-shot and (2) improvements in quality/adaptation in higher n-shot as seen in [53]. In contrast, our method (FSGAN) effectively transfers semantic features while preserving quality across all n-shot settings tested in Figure 2.7. We note variance across n-shot settings for all methods as the data distribution changes.

2.5 Conclusions

We presented Few-shot GAN, a simple yet effective method for adapting a pre-trained GAN based model to a new target domain where the number of training images is scarce. Our core idea lies in factorizing the weights of convolutional/fully-connected layers in a pretrained model using SVD to identify a semantically meaningful parameter space for adaptation. Our strategy preserves the capability of a pre-trained model of generating diverse and realistic samples while provides the flexibility for adapting the model to a target domain with few examples. We demonstrate the effectiveness of the proposed method with close-domain and far-domain adaptation experiments and across various n-shot settings. We show favorable results compared with existing data-efficient GAN adaptation methods.

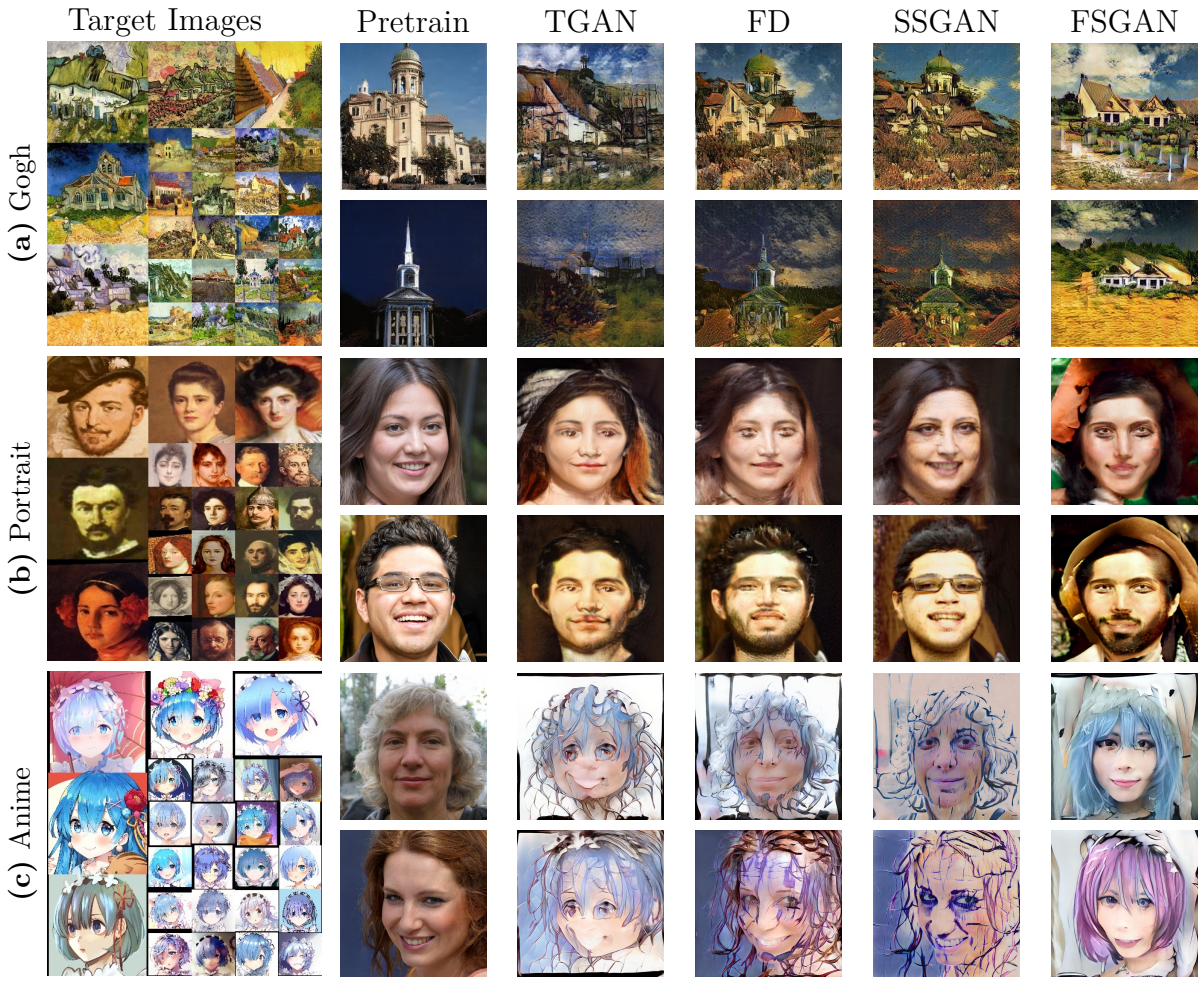


Figure 2.6: **Far-domain adaptation** (Photo→Art). Comparing FSGAN with alternative GAN adaptation methods in the photo-to-art setting. **(a)** FSGAN more effectively alters building layouts and adds landscape in the foreground to match the Van Gogh paintings, maintaining better spatial coherency. **(b)** FSGAN adopts features from the Portraits dataset (hats, beards, artistic backgrounds), while other methods primarily alter image textures. **(c)** FSGAN transforms natural hair and facial features to imitate the anime target while retaining spatial consistency. Note the occurrence of pink hair in our generated images, which does not exist in the few-shot target but is visually consistent.

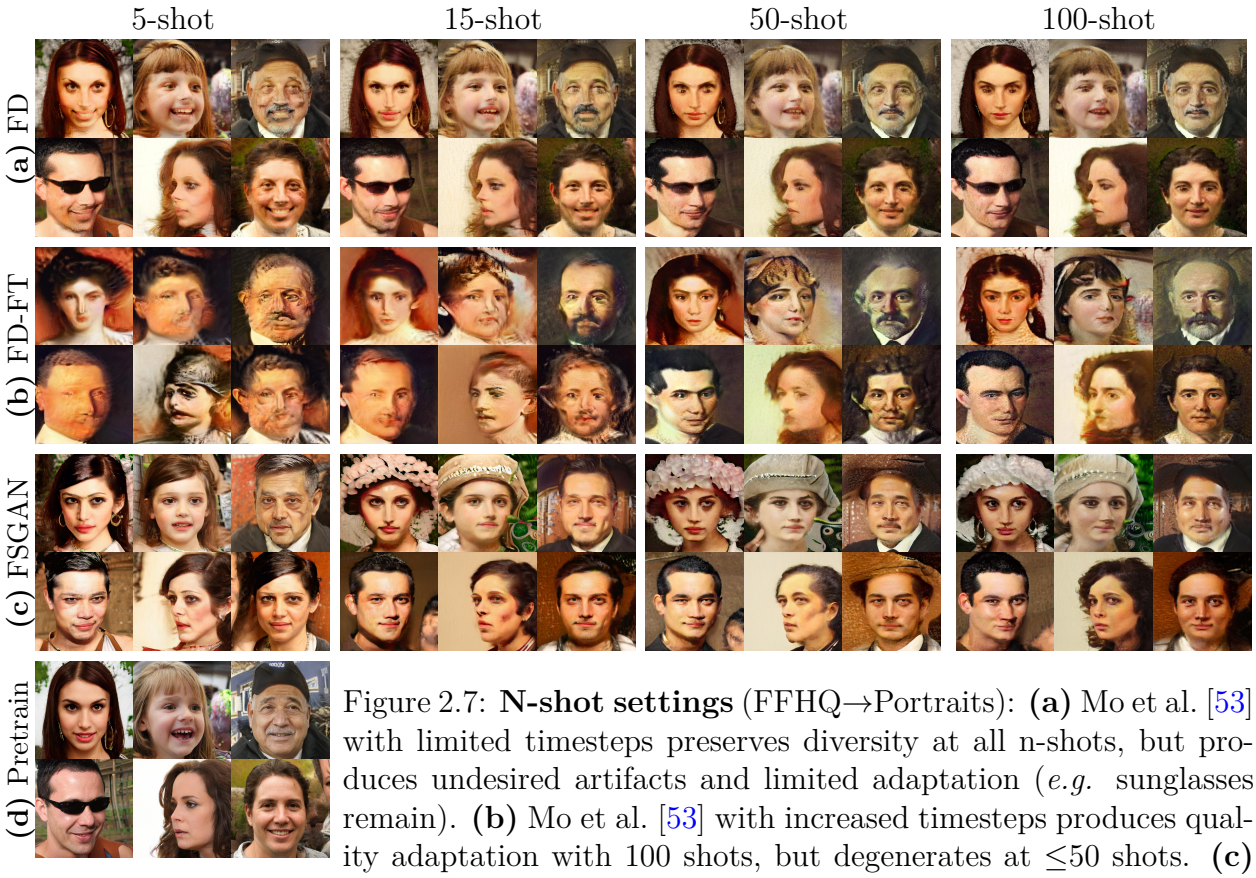


Figure 2.7: **N-shot settings** (FFHQ→Portraits): **(a)** Mo et al. [53] with limited timesteps preserves diversity at all n-shots, but produces undesired artifacts and limited adaptation (*e.g.* sunglasses remain). **(b)** Mo et al. [53] with increased timesteps produces quality adaptation with 100 shots, but degenerates at ≤50 shots. **(c)** FSGAN (ours) is robust to n-shot settings, producing high-quality adaptation even at N=5.
(d) Pretrained FFHQ images.

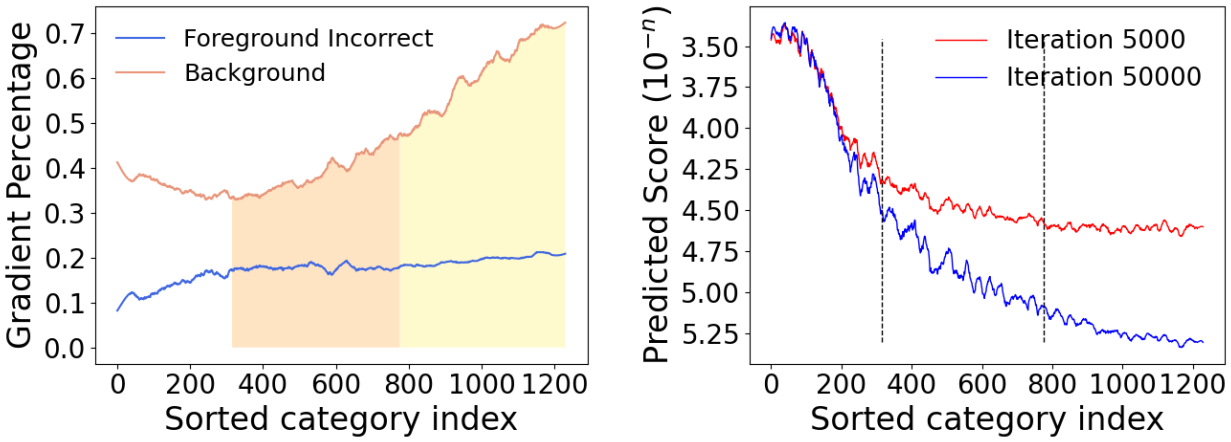
Chapter 3

DropLoss for Long-Tail Instance Segmentation

3.1 Introduction

Object detection and instance segmentation have a wide array of practical applications. State-of-the-art object detection methods adopt a multistage framework [11, 12, 61] trained on large-scale datasets with abundant examples for each object category [42]. However, datasets used in real-word applications commonly fall into a long-tailed distribution over categories, i.e., the majority of classes have only a small number of training examples. Training a model on these datasets inevitably induces an undesired bias towards frequent categories. The limited diversity of rare-category samples further increases the risk of overfitting. Methods for addressing the issues involving long-tailed distributions commonly fall into several groups: *i*) resampling to balance the category frequencies, *ii*) reweighting the losses of rare and frequent categories, and *iii*) specialized architectures or feature transformations.

The instance segmentation problem presents unique challenges for learning long-tailed distributions, as it contains multiple training objectives to supervise region proposal, bounding box regression, mask regression, and object classification. Each of these losses contributes to the overall balance of model training. The prior state-of-the-art in long-tail instance segmentation [70] discovered a phenomenon where the predictions for rare categories are suppressed



(a) Gradient percentage

(b) Prediction scores

Figure 3.1: Motivation. (a) Percentage of gradient updates from incorrect foreground classification (blue) and ground-truth background anchors (orange) on LVIS [16]. We divide the categories into "frequent" (white shading), "common" (orange shading), and "rare" (yellow shading). For rare categories, background gradients occupy a disproportionate percentage of total gradients. (b) The distribution of average foreground class prediction scores for ground-truth background bounding boxes at earlier (red) and later (blue) training stages. We find that, for background bounding boxes, the prediction scores of rare categories are more severely suppressed, and the training is biased towards predicting more frequent categories.

by *incorrect foreground class predictions*. To reduce these "discouraging gradients" and allow the network to explore the solution space for rare categories, the EQL method [70] removes losses to rare categories from incorrect foreground classification. However, we observe that most "discouraging gradients" in fact originate from *correct background classification* (where a bounding box does not contain any labeled objects). In the background case, the classification branch receives losses to suppress all foreground class prediction scores.

In Figure 3.1, we study the effect of such *discouraging gradients* on the different categories of a long-tail dataset, categorized by number of training images into *rare* (1-10 images), *common* (11-100), and *frequent* ($>$ 100) categories. We find that these losses disproportionately affect rare and common categories, due to the infrequency of "encouraging gradients" in which a bounding box contains the correct category label. Specifically, Figure 3.1(a) shows that

50-70% of discouraging gradients for rare categories originate from background predictions, compared with only 30-40% of discouraging gradients for frequent categories. Discouraging gradients from background classification (orange curve) contribute a much higher percentage of total discouraging gradients compared to that of incorrect foreground prediction (blue curve) as used in EQL [70]. Figure 3.1(b) shows that using a ground-truth background anchor, a trained model predicts scores for rare categories with several orders-of-magnitude lower confidence than for frequent categories. This demonstrates a bias towards predicting more frequent categories.

Based on these observations, we develop a simple yet effective method to *adaptively* rebalance the ratio of background prediction losses between rare/common and frequent categories. Our proposed method *DropLoss* removes losses for rare and common categories from background predictions based on sampling a Bernoulli variable with parameters determined by batch statistics. DropLoss prevents suppression of rare and common categories, increasing opportunities for correct predictions of infrequent classes during training and reducing frequent class bias.

The contributions of this work are summarized as follows:

1. We provide an analysis of the unique characteristics of long-tailed distributions, particularly in the context of instance segmentation, to pinpoint the imbalance problem caused by disproportionate discouraging gradients from background predictions during training.

2. We develop a methodology for alleviating imbalances in the long-tailed setting by leveraging the ratio of rare and frequent classes in a sampled training batch.

3. We present state-of-the-art instance segmentation results on the challenging long-tail LVIS dataset [16].

3.2 Related Work

3.2.1 Object Detection and Instance Segmentation.

Two-stage detection architectures [11, 12, 44, 61] have been successful in the object detection setting, where the first stage proposes a "region of interest" and the second stage refines the bounding box and performs classification. This decomposition was initially proposed in R-CNN [12]. Fast R-CNN [11] and Faster R-CNN [61] improve efficiency and quality for object detection. Mask R-CNN later adapts Faster R-CNN to the instance segmentation setting by adding a mask prediction branch in the second stage [21]. Mask R-CNN has proven effective in a wide variety of instance segmentation tasks. Our work adopts this architecture. In contrast with two-stage methods, single-stage methods provide faster inference by eliminating the region proposal stage and instead predicting a bounding box directly from anchors [46, 47, 60]. However, two-stage architectures generally provide better localization.

3.2.2 Learning Long-tailed Distributions.

Techniques for learning long-tailed distributions generally fall into three groups: resampling, reweighting and cost-sensitive learning, and feature manipulation. We discuss each in the following sections.

3.2.3 Resampling Methods.

Oversampling methods [6, 17, 19, 23, 29, 50, 82] duplicate rare class samples to balance out the class frequency distribution. However, oversampling methods tend to overfit to the rare categories, as this type of method does not address the fundamental lack of data. Several

oversampling methods aim to address this by augmenting the available data [6, 17], but undersampling methods are often preferred [9]. Undersampling methods [9, 31, 71] remove frequent class samples from the dataset to balance the class frequency distribution. The loss of information from removing these samples can be mitigated through careful selection using statistical techniques [31, 71]. It can be beneficial to combine the advantages of undersampling and oversampling [6]. Dynamic methods adjust the sampling distribution throughout training based on loss or metrics [56]. Class balance sampling [32, 67] uses class-aware strategies to rebalance the data distribution for learning classifiers and representations. In the context of the dense instance segmentation problem, it is difficult to apply the above resampling methods because the number of class examples per image may vary.

3.2.4 Reweighting and Cost-sensitive Methods.

Rather than rebalancing the sampling distribution, reweighting methods seek to balance the *loss weighting* between rare and frequent categories. Class frequency reweighing methods commonly use the inverse frequency of each class to weight the loss [7, 29, 73]. Cost-sensitive methods [41, 45] aim to balance the model loss magnitudes between rare and frequent categories. An existing meta-learning method [69] explicitly learns loss weights based on the data. Our method provides a simple way to combine class frequency-aware sampling and cost-sensitive learning.

3.2.5 Feature Manipulation Methods.

In contrast to resampling methods and reweighting methods that focus on modifying the loss based on class frequency, feature manipulation methods aim to design specific architectures or feature relationships to address the long-tail problem. Normalization can be used to control

the distribution of deep features, preventing frequent categories from dominating training [32]. Metric learning methods [32, 80] learn maximally-distant "prototypes" of deep features to improve performance on data-scarce categories, effectively transferring knowledge between head and tail categories. Similarly, knowledge transfer in feature space can be accomplished using memory-based prototypes [49] or transfer of intra-class variance [77].

3.2.6 Long-tail Learning Settings.

Several methods have been proposed to handle the problem of learning from imbalanced datasets in other settings such as object classification [5, 7, 30, 70]. In the long-tail object recognition setting, the prior state-of-the-art method [70] uses selective reweighting. Their work observed that rare categories receive significantly more "discouraging gradients" compared with frequent categories, and develop a method for rebalancing discouraging gradients from foreground misclassifications. Their method uses a binary 0 or 1 reweighting based on whether the class is rare or frequent. Unlike this work, our method focuses on the much more prevalent background classification losses in the instance segmentation setting, and we develop a new adaptive resampling and reweighting method which accounts for this imbalance and for the distribution of classes within a sample. Note that most of the methods for learning long-tailed distributions focus on the *image classification* setting where this is no background class (and therefore no losses associated with background). In object detection and instance segmentation, however, the background class plays a very dominant role in the loss. This inspires our design of a reweighting mechanism which specifically considers background class.

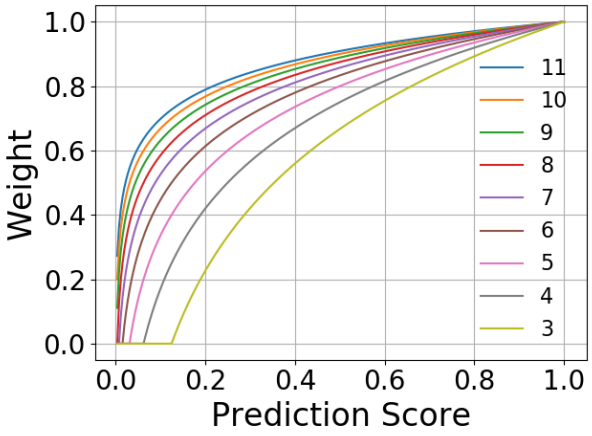
3.3 Approach

Based on the observation that rare and common categories receive disproportionate discouraging gradients from background classifications (compared with frequent categories), the goal of our method is to prevent rare categories from being overly suppressed by reducing the imbalance of discouraging gradients. We are inspired by work in one-stage object detection [41, 45], which encounters a similar problem of large gradients from negative anchors inhibiting learning. We first construct a baseline which modifies the equalization loss [70] to rebalance gradients from foreground and *background* region proposals for rare/common and frequent categories. We show that this baseline leads to improved results over [70] but requires careful hyperparameter selection and exhibits a clear tradeoff between rare/common and frequent categories. To alleviate this problem, we propose a stochastic *DropLoss* which improves the overall frequent-rare category performance as well as improving the tradeoff as measured by a Pareto frontier 3.4.

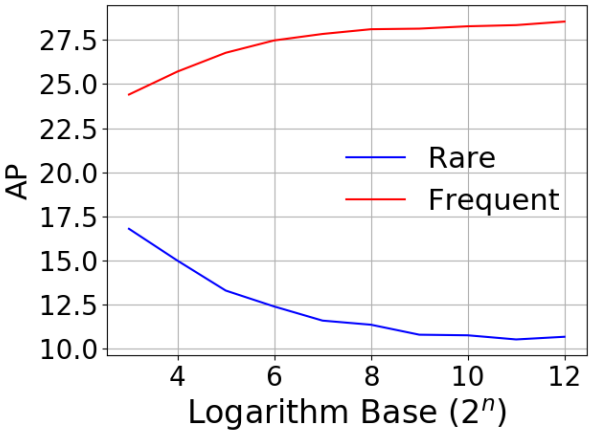
3.3.1 Revisiting the Equalization Loss

3.3.2 Equalization Loss.

We start with a review of the equalization loss [70]. Equalization loss modifies sigmoid cross-entropy to alleviate discouraging gradients from incorrect foreground predictions. Note that, in sigmoid cross-entropy, the ground-truth label y_j represents only a binary distribution for the foreground category j, and no extra class label for the background is included. That is, we have $y_j = 1$ if the ground-truth category of a region is j. On the other hand, if a region belongs to the background, we have $y_j = 0$ for all the categories. During training, a region proposal is labeled as "background" if its IoU with any ground-truth region of a foreground



(a) Weight w_j as a function of the logarithm base b.



(b) Frequent-rare category performance tradeoff

Figure 3.2: Background equalization loss. We present an extension of the equalization loss that specifically focuses on the background classification. (a) The curves of the $\mathcal{L}_{\text{BEQL}}$ weights in (3.4) with different choices for the logarithm base b. Smaller values of the logarithm base b reduce the effects of background more. (b) The experimental result of applying different logarithm bases shows a tradeoff between the mean average precision (mAP) of rare categories and the mAP of frequent categories with respect to different logarithm base settings. Note that the background equalization loss $\mathcal{L}_{\text{BEQL}}$ with a large base b reduces to the existing equalization loss $\mathcal{L}_{\text{EQL}}$.

class is lower than 50%.

Given a region proposal r, the equalization loss is formulated as follows:

$$\mathcal{L}_{\text{EQL}} = -\sum_{j=1}^{C} w_j \log(\hat{p}_j)\,, \;\; \hat{p}_j = \begin{cases} p_j, & \text{if } y_j = 1\,, \\ 1 - p_j, & \text{otherwise,} \end{cases} \tag{3.1}$$

$$w_j = 1 - E(r)T_\lambda(f_j)(1 - y_j)\,, \tag{3.2}$$

where C is the number of categories, p_j is predicted logit, and f_j is the frequency of category j in the dataset. The indicator function $E(r)$ outputs 1 if r is a foreground region and 0 if it belongs to the background. More specifically, for a region proposal r that is considered a background region with all y_j being zero, we have $E(r) = 0$. $T_\lambda(f)$ is also a binary indicator

function that, given a threshold λ, outputs 1 if $f < \lambda$ to indicate the category is of low frequency. It can be verified from 3.2 that, for a foreground region r (*i.e.*, $E(r) = 1$), the weight w_j is either 1 or $1 - T_\lambda(f_j)$, depending on the frequency of the ground-truth category j. Further, if a category j is of low frequency (rare category), the weight $w_j = 1 - T_\lambda(f_j)$ becomes zero and thus no penalty is given to incorrect foreground predictions. On the other hand, for frequent categories, the weight is 1 and the penalty of incorrect prediction remains $-\log(1 - p_j)$. By removing discouraging gradients to rare/common categories from incorrect foreground predictions, the equalization loss achieved state-of-the-art on the LVIS Challenge 2019. Foreground class label prediction is selected using the maximum logit, so entirely removing the loss allows the network to optimize rare categories without penalties, as long as the prediction logit is less than ground truth for the frequent categories. This approach removes large penalties for non-zero confidences in rare categories, which otherwise imbalance the training to suppress rare categories.

3.3.3 Background Equalization Loss.

In contrast to the mechanism of equalization loss that prevents large penalties for non-zero confidences in rare categories, cost-sensitive learning methods only reduce [45] or remove [41] discouraging gradients if the magnitude of the loss falls below some threshold. Our core insight is that foreground and background categories require different approaches due to the differences in prediction criteria. For background categories, the network predicts background class if *all* logits p_j fall below a threshold. For foreground categories, the prediction is selected using the *maximum* logit p_j. Inspire by cost-sensitive loss and equalization loss,

we present the *background equalization loss* as an extension to the original equalization loss:

$$\mathcal{L}_{\text{BEQL}} = -\sum_{j=1}^{C} w_j \log(\hat{p}_j)\,,\ \ \hat{p}_j = \begin{cases} p_j, & \text{if } y_j = 1\,, \\ 1 - p_j, & \text{otherwise,} \end{cases} \tag{3.3}$$

$$w_j = \begin{cases} 1 - T_\lambda(f_j)(1 - y_j), & \text{if } E(r) = 1\,, \\ 1 - T_\lambda(f_j) \cdot \min\{-\log_b(p_j), 1\}, & \text{otherwise.} \end{cases} \tag{3.4}$$

By comparing (3.2) and (3.4), we can see that the background equalization loss differs from the equalization loss in the weights for background regions. The equalization loss always penalizes a background region ($E(r) = 0$ and thus $w_j = 1$) even if the category is of low frequency. In contrast, our background equalization loss gives smaller weight to background predictions as long as their confidences are low. We use a logarithm base b to control the sensitivity of the weight concerning the confidence of background prediction. Figure 3.2(a) shows the curves of the $\mathcal{L}_{\text{BEQL}}$ weights in (3.4) by varying the value of the logarithm base b. For example, suppose we would want to focus on the performance of the rare category, we can set the value of $b = 2$. The main idea here is to alleviate the accumulation of small but non-negligible discouraging gradients from the background. When applying the proposed background equalization loss with different logarithm bases, however, we see a clear performance tradeoff between frequent and rare categories (see Figure 3.2(b)). The results show that the average precision of the rare categories behaves in the *opposite* way as the average precision of the frequent categories for different choices of logarithm bases.

Architecture	Backbone	Loss	AP (%)	AP_{50}	AP_{75}	AP_r	AP_c	AP_f	AR	AP_{bbox}
Mask R-CNN	R-50-FPN	BCE	21.5	33.4	22.9	4.7	21.2	**28.6**	28.3	21
		EQL [70]	23.8	36.3	25.2	8.5	25.2	28.3	31.5	23.5
		DropLoss (Ours)	**25.5**	**38.7**	**27.2**	**13.2**	**27.9**	27.3	**34.8**	**25.1**
Mask R-CNN	R-101-FPN	BCE	23.6	36.5	25.1	5.6	24.2	**30.1**	30.9	23.3
		EQL [70]	26.2	39.5	27.9	11.9	27.8	29.8	33.8	26.2
		DropLoss (Ours)	**26.9**	**40.6**	**28.9**	**14.8**	**29.7**	28.3	**36.4**	**26.8**
Cascade R-CNN	R-50-FPN	BCE	21.4	32	23.1	3.4	20.4	**29.8**	27.6	22.8
		EQL [70]	24.2	35.9	25.8	7.8	25	29.7	31.4	26
		DropLoss (Ours)	**25**	**37**	**26.9**	**9.1**	**27.2**	28.7	**34**	**26.9**
Cascade R-CNN	R-101-FPN	BCE	23	34.4	24.7	3.5	22.8	**31.2**	29.9	24.9
		EQL [70]	25.4	37.3	27.3	7.2	26.6	31	33.1	27.2
		DropLoss (Ours)	**26.4**	**39**	**28.1**	**11.5**	**28.5**	29.7	**35.5**	**28.6**

Table 3.1: Comparison between architecture and backbone settings, evaluated on LVIS v0.5 validation set. We compare BCE (binary cross-entropy), EQL (equalization loss) and Drop (DropLoss). AP/AR refers to mask AP/AR, and subscripts 'r', 'c', and 'f' refer to rare, common, and frequent categories.

3.3.4 DropLoss

While suppressing discouraging gradients from the background shows improvement for the rare categories, the background equalization loss has a drawback. The performance often sensitively depends on the choice of the logarithm base. It is difficult to choose an appropriate logarithm base that works for different long-tailed distributions without suffering from a tradeoff between frequent and rare categories. In light of this, we propose a new stochastic method, called *DropLoss*, which dynamically balances the influence of background discouraging gradients for rare/common/frequent categories.

Similar to the design of the background equalization loss, we seek to adjust weights on the logits of low-frequency categories for background region proposals. In DropLoss, we introduce a Bernoulli distribution and sample a binary value from the distribution as the weight w_j if a region belongs to the background. Further, we determine the parameter of the Bernoulli distribution by a beta sampling distribution over the occurrence ratios of rare, common, and frequent categories for the regions generated by the Region Proposal Network

[61] during training. The Bernoulli distribution with a Beta prior is suitable because we aim to model binary outcomes with varying biases in a stochastic manner.

Given a batch of region proposals, we compute the ratio between the occurrences of 'rare + common' categories to all foreground occurrences (*i.e.*, 'rare + common + frequent' categories). In other words, we treat a batch of region proposals as a sample of occurrence ratio that is drawn from a beta distribution to provide the parameter of the Bernoulli distribution. Our intuition behind such a scheme is simple: For region proposals of rare and common categories, their occurrences in a batch are of low frequency. Therefore, the discouraging gradients from the background predictions should be accordingly discounted for rare and common categories.

We formulate DropLoss as follows:

$$\mathcal{L}_{\text{Drop}} = -\sum_{j=1}^{C} w_j \log(\hat{p}_j)\,,\ \hat{p}_j = \begin{cases} p_j, & \text{if } y_j = 1\,, \\ 1 - p_j, & \text{otherwise,} \end{cases} \tag{3.5}$$

$$w_j = \begin{cases} 1 - T_\lambda(f_j)(1 - y_j), & \text{if } E(r) = 1\,, \\ w \sim \text{Ber}(\mu_{f_j}), & \text{otherwise,} \end{cases} \tag{3.6}$$

where a random sample $w \in \{0, 1\}$ is drawn from Bernoulli distribution $\text{Ber}(\mu_{f_j})$ if the region proposal r belongs to the background, *i.e.*, $E(r) = 0$. The parameter μ_{f_j} of the Bernoulli distribution is determined by the occurrence ratio of low-frequency ('rare + common') categories in the current batch of region proposals. We compute the parameter by

$$\mu_{f_j} = \begin{cases} (n_{\text{rare}} + n_{\text{common}})/n_{\text{all}}, & \text{if } T_\lambda(f_j) = 1\,, \\ n_{\text{frequent}}/n_{\text{all}}, & \text{otherwise,} \end{cases} \tag{3.7}$$

where n_{rare}, n_{common}, and n_{frequent} are the numbers of occurrences of rare, common, and frequent categories in the current training batch of foreground region proposals. The total number of foreground occurrences is $n_{\text{all}} = n_{\text{rare}} + n_{\text{common}} + n_{\text{frequent}}$. Implementation of the above DropLoss scheme is straightforward: For each batch, we derive the parameter μ_{f_j} depending on whether category j is rare/common or frequent. We can then simulate a flip of a biased coin with head probability μ_{f_j} and assign $w_j = 1$ if we get a head.

A region proposal is annotated as a background region if it does not overlap with any ground-truth foreground region, or if the IoU is lower than 50%. If the number of rare category occurrences in a given batch is large, discouraging gradients to that rare category are more likely to be kept (with a higher chance to get $w_j = 1$). On the other hand, if a rare category does not appear very often in a batch, it is highly probable that discouraging gradients to the rare category will be dropped. Therefore, our dropping strategy tends to neglect unrelated non-overlapping background proposals but would be inclined to keep more related ($0 <$ IoU < 0.5) background proposals.

3.4 Results

In this section, we present the implementation details and experimental results. We compare DropLoss with the state-of-the-art long-tail instance segmentation baselines on the challenging LVIS dataset [16]. To validate the effectiveness of this approach, we compare across different architectures and backbones and integrate with additional long-tail resampling methods. We find that DropLoss demonstrates consistently improved results in AP and AR across all these experimental settings.

3.4.1 Dataset.

Following the previous work *equalization loss* [70], we train and evaluate our model on LVIS benchmark dataset. LVIS is a large vocabulary instance segmentation dataset, containing 1,230 categories. In LVIS dataset, categories are sorted into three groups based on the number of images in which they appear: *rare* (1-10 images), *common* (11-100), and *frequent* (> 100). We report AP for each bin to quantify performance in the long-tailed distribution setting. We train our model on the 57K-image LVIS v0.5 training set and evaluate it on the 5K-image LVIS v0.5 validation set.

3.4.2 Implementation Details.

For our experiments, we adopt the Mask R-CNN [22] architecture with Feature Pyramid Networks [43] as a baseline model. We train the network using stochastic gradient descent with a momentum of 0.9 and a weight decay of 0.0001 for 90K iterations, with batch size 16 on eight parallel NVIDIA 2080 Ti GPUs. We initialize the learning rate to 0.2 and decay it by a ratio of 0.1 at iterations 60,000 and 80,000. We use the Detectron2 [74] framework with default data augmentation. The data augmentation includes scale jitter with a short edge of (640, 672, 704, 736, 768, 800) pixels and a long edge no more than 1,333 pixels horizontal flipping. In the Region Proposal Network (RPN), we sample 256 anchors with a 1:1 ratio between foreground and background to compute the RPN loss and choose 512 ROI-aligned proposals per image with a 1:3 foreground-background ratio for later predictions. Based on LVIS [16], the prediction threshold is reduced from 0.05 to 0.0, and we set the top 300 bounding boxes as prediction results. This setting is widely used in LVIS training and evaluation. For all the experiments, we report the average results of three independent runs of model training. The variances in AP are generally small (approximately 0.1-0.2).

Method	Use RFS	AP (%)	AP_{50}	AP_{75}	AP_r	AP_c	AP_f	AP_s	AP_m	AP_L	AP_{bbox}
Sigmoid	-	21.5	33.4	22.9	4.7	21.2	**28.6**	15.6	29.3	39	21
Softmax	-	21.3	33.1	22.6	3	21.2	**28.6**	15.8	28.5	39.2	21
EQL [70]	-	23.8	36.3	25.2	8.5	25.2	28.3	17.1	31.4	41.7	23.5
DropLoss (Ours)	-	**25.5**	**38.7**	**27.2**	**13.2**	**27.9**	27.3	**17.7**	**32.7**	**43.2**	**25.1**
Sigmoid	✓	23.8	36.3	25.2	8.5	25.2	**28.3**	17.1	31.4	41.7	23.5
Softmax	✓	24.3	37.8	25.9	14.1	24.3	**28.3**	16.5	31.6	41.2	23.8
EQL [70]	✓	25.5	39	27.2	16.7	26.3	28.1	17.5	33	43	25
DropLoss (Ours)	✓	**26.4**	**40.3**	**28.4**	**17.3**	**28.7**	27.2	**17.9**	**33.1**	**44**	**25.8**

Method	Use RFS	AP (%)	AP_{50}	AP_{75}	AP_r	AP_c	AP_f	AP_s	AP_m	AP_L	AP_{bbox}
Baseline	-	16.2	25.9	16.9	0.7	12.6	27	10.5	22.7	32.7	16.6
EQL [70]	-	18.4	28.6	19.4	2.5	16.5	**27.4**	11.9	25.4	35.6	18.9
DropLoss (Ours)	-	**19.8**	**30.9**	**20.9**	**3.5**	**20**	26.7	**12.9**	**27.5**	**37.1**	**20.4**
Baseline	✓	18.8	29.6	19.9	5.6	16.6	27.1	11.6	25.6	35.7	19.2
EQL [70]	✓	21	32.7	22.3	9.1	20.1	**27.3**	13.1	28.5	39.2	21.7
DropLoss (Ours)	✓	**22.3**	**34.5**	**23.6**	**12.4**	**22.3**	26.5	**13.9**	**29.9**	**40**	**22.9**

Table 3.2: Evaluation on LVIS v0.5 (top) and LVIS v1.0 (bottom) validation sets *with* and *without* Repeat Factor Sampling (RFS). Here we use Mask-RCNN and ResNet-50. DropLoss achieves the best overall AP across both settings.

Comparisons with state-of-the-art methods. In our experiments, we use Mask R-CNN [21] as our architecture and compare it with two baseline training methods: standard Mask R-CNN and the equalization loss [70]. To verify that DropLoss is effective across different settings, we validate on several different architectures and backbones. We test ResNet50 and ResNet101 [20] as backbones, and compare the Cascades R-CNN [4] as an alternative architecture to Mask R-CNN [22]. Table 3.1 reports the results, where all methods are tested using the same experiment settings and environment. We find that DropLoss achieves improved performance (in terms of overall AP) compared with both baselines across all backbones and architectures. We are most interested in the AP_r, AP_c, AP_f and AR. Although the AP_f (frequent) decreases slightly in our method, our AP_r (rare) and AP_c(*common*) increase significantly. Our method improves the AP and AR by a large margin, indicating the overall performance across all categories is improved. In particular, using Mask R-CNN with ResNet-50 as the backbone, we achieve a 1.7 AP improvement over the state-of-the-art

Figure 3.3: Visual results comparison. Qualitative results of the Mask R-CNN trained with standard cross-entropy loss (*Top*) and the proposed DropLoss (*Bottom*). Instances with scores larger than 0.5 are shown.

method [70] (winner of the LVIS 2019 challenge). Across all the settings, compared with the baselines, DropLoss can more successfully balance the tradeoff between rare and frequent categories, resulting in better performance in the long-tailed distribution dataset.

3.4.3 Incorporating with Resampling Methods.

Here we show that our approach can be combined with state-of-the-art resampling methods to improve learning long-tailed distribution further. Specifically, we adopt the Repeat Factor Sampling (RFS) [16] that uses the number of images per category to determine the sampling frequency. Table 3.2 shows the quantitative comparisons of different loss function choices on the LVIS v0.5 validation set.[1] We find that applying RFS generally improves the performance of all the methods. The proposed DropLoss compares favorably against other

[1]Note that the EQL results (25.5 AP) are not consistent with the reported results (26.1 AP). We use the public implementation without changes and report the average over 3 runs. The difference may be due to number of GPUs used for training, resulting in different batch normalization. For fair comparisons, we use the same hardware and experimental setting to train all models.

baseline methods either with or without using RFS. Note that the RFS method rebalances based on *overall* data distribution, while DropLoss reweights the loss based on statistics in the *each batch*. The complementary nature of the two methods may explain why integrating RFS and the DropLoss leads to improved results.

3.4.4 Measuring the Frequent-rare Category Performance Tradeoff.

Methods for learning long-tail distribution often involve a tradeoff between accuracy on rare, common, and frequent categories. Here we wish to quantify this tradeoff for various methods. We compare our proposed DropLoss against three baselines: equalization loss [70], background equalization loss, and fixed drop ratio.

Equalization loss and DropLoss have no tunable hyperparameters. Background equalization loss has the log base as a tunable hyperparameter. A fixed drop ratio has the drop ratio as a hyperparameter. These methods may be adjusted to measure the tradeoff between object categories. We can use the Pareto Frontier from multi-objective optimization to visualize this tradeoff, as seen in Figure 3.4. We observe that for reweighting methods with tunable hyperparameters, improvement in rare AP_r or common AP_c generally leads to a rapid decrease in frequent AP_f. Our proposed DropLoss does not have tunable hyperparameters, but Figure 3.4 demonstrates that DropLoss balances more effectively between AP_r, AP_c and AP_f, resulting in higher overall AP than other baselines.

DropLoss adapts to the sampling distribution so that if a rare category appears in a given batch, its loss is less likely to be dropped. However, if a rare category does not appear in a batch, the chance of its loss being dropped is very high. This allows the network to *dynamically* attend to the categories that it sees in a given batch, decreasing drop loss

probability selectively for only those categories. We postulate that this allows the network to achieve a better overall balance between frequent and infrequent categories.

3.4.5 Quantitative Comparison Between the DropLoss and Our Proposed Baseline BEQL.

To validate that the DropLoss provides better pareto-efficiency over BEQL, in Table 3.3, we compare the DropLoss with two *best-performing* results (in term of overall AP) from BEQL. DropLoss still offers overall better performance (not only in AP but also in AR) despite not sweeping the parameters on the validation set to find the best performance as in BEQL.

Method	AP (%)	AP_r	AP_c	AP_f	AP_{bbox}	AR
BEQL ($b = 4$)	25.2	**14.6**	**28.1**	25.9	24.9	34.1
BEQL ($b = 5$)	25.1	13.4	27.4	26.9	24.8	34.3
DropLoss	**25.5**	13.2	27.9	**27.3**	**25.1**	**34.8**

Table 3.3: Comparison between DropLoss and two top-performing results from our another proposed method BEQL. Evaluation on LVIS v0.5 validation set. DropLoss offers overall better performance (in both AP and AR) compared with the BEQL baseline.

3.4.6 Visual Results.

Figure 3.3 demonstrates the results on a dense instance segmentation example containing common/rare category. For example, the goose in the first image is a 'common' object category. We demonstrate the suppression of these less-frequent categories, as most of the geese in this image are classified as background or with low confidence. In contrast, training the model with the proposed loss correctly identifies all geese as foreground, and predicts

category "goose" with high confidence and other waterbirds with lower confidence. Despite the stochastic removal of rare and common category losses for background proposals, we find that the network does not misclassify background regions as foreground. The distinction between background and foreground is likely less difficult to learn than the distinction between foreground image categories, so reducing background gradients does not appear to significantly affect background/foreground classification. By reducing the suppression of rare and common categories via background predictions, our method allows for rare and common categories to improve prediction scores, decreasing bias towards frequent categories.

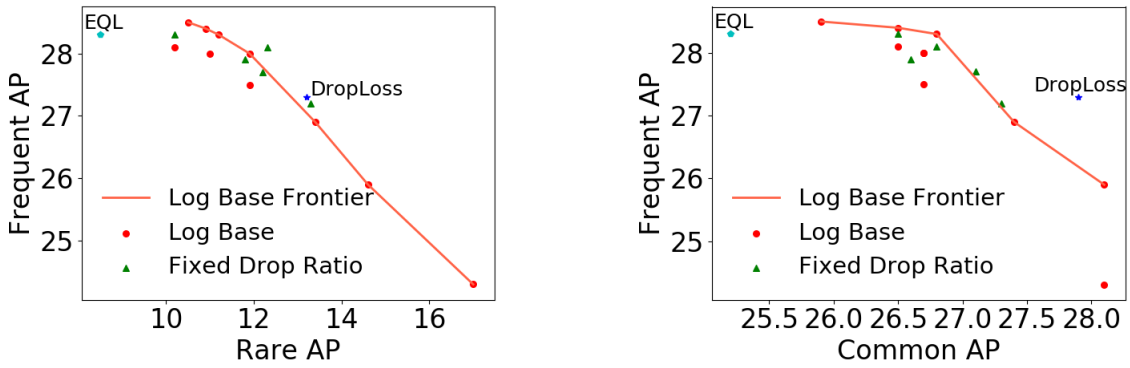
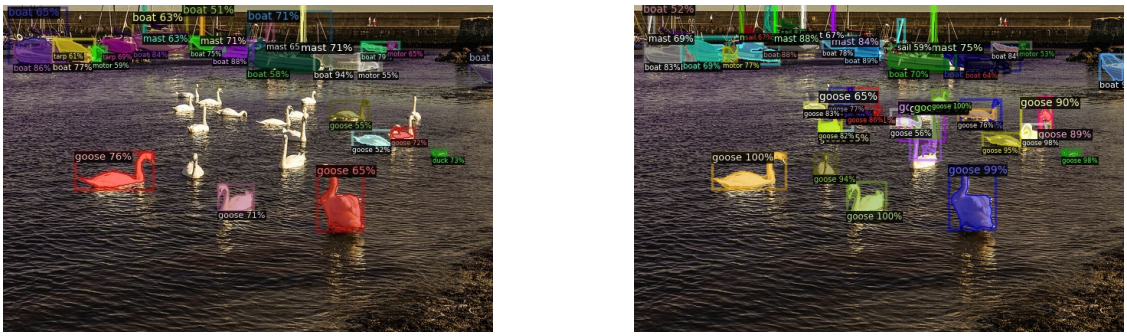


Figure 3.4: Comparison between rare, common, and frequent categories AP for baselines and our method. We visualize the trade-off for 'rare vs. frequent' and 'common vs. frequent' as a Pareto frontier, where the top-right position indicates an ideal trade-off between objectives. DropLoss achieves an improved trade-off between object categories, resulting in higher overall AP.

3.4.7 Ablation Study

Measuring the balance. Long-tail distribution methods often involve a trade-off between accuracy on rare, common, and frequent categories, so we wish to quantify this trade-off for various methods. We compare our proposed DropLoss against three baselines: equalization loss [70], background equalization loss, and fixed drop ratio. Equalization loss and DropLoss have no tunable hyperparameters. Background equalization loss has the log base as a tun-



(a) Mask R-CNN (b) DropLoss (Ours)

Figure 3.5: Qualitative results of a Mask R-CNN baseline and b the proposed DropLoss. Instances with score > 0.5 are shown. DropLoss adaptively removes background proposal losses of rare and common categories to reduce bias towards misclassifying these objects as background. In this case, the *common* category 'goose' is misclassified as background in a, and correctly identified in b.

able hyperparameter. A fixed drop ratio has the drop ratio as a hyperparameter, so these methods may be adjusted to measure the trade-off between object categories. We can use the Pareto Frontier graph from multi-objective optimization to visualize this trade-off, as seen in Figure 3.4. We observe that for the reweighting methods with tunable hyperparameters, improvement in rare AP_r or common AP_c generally leads to a decrease in frequent AP_f. Our proposed DropLoss does not have tunable hyperparameters, but Figure 3.4 demonstrates that DropLoss balances more effectively between AP_r, AP_c and AP_f, resulting in a higher overall AP than our baselines. DropLoss adapts to the sampling distribution so that if a rare category appears in a given batch, its loss is less likely to be dropped. However, if a rare category does not appear in a batch, the chance of its loss being dropped is very high. This allows the network to dynamically attend to the categories that it sees in a given batch, decreasing drop loss probability selectively for only those categories. We postulate that this allows the network to achieve a better overall balance.

Qualitative results. Figure 3.5 demonstrates the results on a dense instance segmentation example containing common category "goose". Figure 3.5a demonstrates the suppression of less-frequent categories, as most of the geese in this image are classified as background or with low confidence. In contrast, Figure 3.5b correctly identifies all geese as foreground, and guesses category "goose" with high confidence and other waterbirds with lower confidence. Despite the stochastic removal of rare and common category losses for background proposals, we find that the network does not misclassify background regions as foreground. The distinction between background and foreground is likely less difficult to learn than the distinction between foreground image categories, so reducing background gradients does not appear to significantly affect background/foreground classification. By reducing the suppression of rare and common categories via background predictions, our method allows for rare and common categories to improve prediction scores, decreasing bias towards frequent categories.

3.5 Conclusions

Through analysis of the loss gradient distributions over rare, common, and frequent categories, we discovered that disproportionate background gradients suppress less-frequent categories in the long-tailed distribution problem. To address this problem, we propose DropLoss, which balances the background loss gradients between different categories via random sampling and reweighting. Our method provides a sizable performance improvement across different backbones and architectures by improving the balance between object categories in the long-tail instance segmentation setting. We focus on the challenging problem of instance segmentation, but we expect that DropLoss may be applicable to other perception problems with long-tailed distributions. We leave this exploration to future work.

Chapter 4

Summary

In this thesis, we investigated methods of improving data efficiency for image generation and instance segmentation. We presented a method of adapting pre-trained GANs to new domains while maintaining diversity (originally published in Robb et al. [62]). We use SVD of factorized weights to extract a small, semantically meaningful set of weights for fine-tuning, and show that this strategy allows for flexible adaptation with preservation of diversity and realism. We also demonstrated a method of improving long-tail performance for instance segmentation (originally published in Hsieh et al. [28]). We observed that the background class predictions disproportionately suppress less-frequent categories, and develop an adaptive method of loss rebalancing based on random sampling and reweighting. Our method demonstrates a significant performance improvement across different architecture setting in the long-tail instance segmentation setting. We believe that developing new methods for data-efficient vision and learning is essential for safe, practical application of state-of-the-art research methods to real-world scenarios. We leave this exploration to future work.

Bibliography

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

[2] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. In *ICML*, 2018.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

[6] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

[8] Teófilo Emídio De Campos, Bodla Rakesh Babu, and Manik Varma. Character recognition in natural images. *VISAPP (2)*, 7, 2009.

[9] Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pages 1–8. Citeseer, 2003.

[10] Andrew Gambardella, Atılım Günes Baydin, and Philip H. S. Torr. TransFlow Learning: Repurposing flow models without retraining. In *arXiv*, 2019.

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[13] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer, 1971.

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *NeurIPS*, pages 5767–5777, 2017.

[16] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[17] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-

sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[18] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable gan controls. In *ECCV*, 2020.

[19] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE, 2008.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[23] Paulina Hensman and David Masko. The impact of imbalanced training data for convolutional neural networks. *Degree Project in Computer Science, KTH Royal Institute of Technology*, 2015.

[24] Javier Hernandez-Ortega, Javier Galbally, Julian Fierrez, Rudolf Haraksim, and Laurent Beslay. Faceqnet: quality assessment for face recognition based on deep learning. In *2019 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2019.

[25] 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 *NeurIPS*, 2017.

[26] 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 *NeurIPS*, 2017.

[27] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

[28] Ting-I Hsieh, Esther Robb, Hwann-Tzong Chen, and Jia-Bin Huang. Droploss for long-tail instance segmentation. In *AAAI*, 2021.

[29] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[30] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[31] Herman Kahn and Andy W Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

[32] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng,

and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[33] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

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

[35] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. *CoRR*, 2019.

[36] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020.

[37] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

[38] Jayant Kumar, Francine Chen, and David Doermann. Sharpness estimation for document and scene images. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3292–3295. IEEE, 2012.

[39] Nojun Kwak. Principal component analysis based on l1-norm maximization. *TPAMI*, 30(9):1672–1680, 2008.

[40] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.

[41] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8577–8584, 2019.

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[43] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[44] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[45] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[46] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[47] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[48] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[49] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019.

[50] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[51] John Martyn, Guifre Vidal, Chase Roberts, and Stefan Leichenauer. Entanglement and tensor networks for supervised image classification. *arXiv preprint arXiv:2007.06082*, 2020.

[52] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[53] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze Discriminator: A simple baseline for fine-tuning GANs. *arXiv preprint arXiv:2002.10964*, 2020.

[54] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *ICCV*, 2019.

[55] Ben Poole, Alexander A Alemi, Jascha Sohl-Dickstein, and Anelia Angelova. Improved generator objectives for gans. *arXiv preprint arXiv:1612.02780*, 2016.

[56] Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, et al. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 112–117. IEEE, 2018.

[57] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2015.

[58] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae2. In *NeurIPS*, 2019.

[59] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NeurIPS*, 2017.

[60] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[61] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[62] Esther Robb, Wen-Sheng Chu, Abhishek Kumar, and Jia-Bin Huang. Few-shot adaptation of generative adversarial networks. *arXiv preprint*, 2020.

[63] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NeurIPS*, 2016.

[64] Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

[65] Hanie Sedghi, Vineet Gupta, and Philip M Long. The singular values of convolutional layers. *arXiv preprint arXiv:1805.10408*, 2018.

[66] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: Learning a generative model from a single natural image. In *ICCV*, 2019.

[67] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pages 467–482. Springer, 2016.

[68] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. InGAN: Capturing and retargeting the "dna" of a natural image. In *ICCV*, 2019.

[69] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pages 1917–1928, 2019.

[70] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11662–11671, 2020.

[71] Chih-Fong Tsai, Wei-Chao Lin, Ya-Han Hu, and Guan-Ting Yao. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Information Sciences*, 477:47–54, 2019.

[72] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring GANs: generating images from limited data. In *ECCV*, 2018.

[73] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7029–7039, 2017.

[74] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[75] Abhay Yadav, Sohil Shah, Zheng Xu, David Jacobs, and Tom Goldstein. Stabilizing adversarial nets with prediction methods. *arXiv preprint arXiv:1705.07364*, 2017.

[76] Shin'ya Yamaguchi, Sekitoshi Kanai, and Takeharu Eda. Effective data augmentation with multi-domain learning GANs. *AAAI*, 2019.

[77] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5704–5713, 2019.

[78] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[79] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019.

[80] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5409–5418, 2017.

[81] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[82] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.