Few Shot Generative Domain Adaptation Via Inference-Stage Latent Learning in GANs

Arnab Kumar Mondal
IIT Delhi

Piyush Tiwary
IISc, Bengaluru

Parag Singla
IIT Delhi

Prathosh AP
IISc, Bengaluru

Abstract

In this study, we adapt generative models trained on large source datasets to scarce target domains. We adapt a pre-trained Generative Adversarial Network (GAN) without retraining the generator, avoiding catastrophic forgetting and over-fitting. Starting from the observation that target images can be 'embedded' onto the latent space of a pre-trained source-GAN, our method finds the latent code corresponding to the target domain on the source latent manifold. Optimizing a latent learner network during inference generates a novel target embedding that is supplied to the source-GAN generator to generate target samples. Our method, albeit simple, can be used to generate data from multiple target distributions using a generator trained on a single source distribution.

1 Introduction

1.1 Few Shot Image Generation - Background

Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] form a family of highly successful deep generative models for generation of high-quality realistic data such as images [Karras et al., 2018, 2019, 2020a]. However, one of their major caveats is that they require a large number of images for proper training [Sushko et al., 2021, Shermin et al., 2021, Tran et al., 2021]. This problem poses practical restrictions on their applications [Rao et al., 2021], as the number of training data is often limited to the order of hundreds or even tens at times [Zhao et al., 2020]. Therefore, it is crucial to adapt GANs for few-shot settings.

One way to accomplish this objective is to utilise a generative model built on a larger, but 'close' source dataset [Wang et al., 2018, 2020]. Several ideas, such as learning latent transformations [Wang et al., 2020], retraining generators on target data with Elastic-weight consolidation regularization [Li et al., 2020], and cross-domain correspondence [Ojha et al., 2021], have been proposed. All of them adapt the generator of a GAN trained on a large source dataset to the target dataset so that the re-trained generator imbibes the target's "style" while keeping the source domain's "variability." While the above methods demonstrate progress in adapting a pretrained GAN, they have flaws such as lack of diversity owing to over-fitting. De-novo retraining on each new target may lead to forgetting source information. In this paper, we'll answer the question **Can a GAN trained on a single large source dataset be adapted to various target domains with few instances without retraining the source generator?**

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

1.2 Motivation and Contributions

Recently, it has been observed that out-of-distribution data can be embedded [Abdal et al., 2019, 2021, Richardson et al., 2021, Tov et al., 2021] on to the latent space of high-fidelity GANs such as StyleGAN [Karras et al., 2019, 2020a]. Further, traversal in the latent space of such high-fidelity GANs have shown to yield a variety of images [Karras et al., 2020a]. This motivates us to hypothesize the existence of a target-data manifold in the latent space of the pretrained source-GAN.
Towards this end, we propose solving an inference-time optimization problem on the latent space of a pretrained GAN to locate the latent vectors that create the target data during inference without retraining the source-generator. This work's contributions are: (a) We propose a simple procedure to utilize a GAN trained on large-scale source-data to generate samples from a target domain with very few (1-10) examples. (b) Our procedure is shown to be capable of generating data from multiple target domains using a single source-GAN without the need for re-training or fine-tuning it. (c) Extensive experimentation shows that our method generates diverse and high-quality target samples with very few examples surpassing the performance of the baseline methods.

Even though our method is aimed at finding the latent manifold corresponding to the target data, it is unique in multiple respects as compared to other latent-transformation methods such as MineGAN [Wang et al., 2020]. Specifically, (a) Proposed method finds point estimates via inference-time optimization compared to [Wang et al., 2020] where a distribution-level transformation is sought, leading to overfitting with few samples. (b) We train a latent learner every time we desire to generate a batch of images. This contributes to the diversity. (c) This method does not retrain the source generator with target data which preserves the diversity imbibed via source-data and enables generation from multiple targets at once.

2 Proposed Method

2.1 Overview

Given a few target samples and a GAN trained on a source dataset, our method optimizes a new multi-layer perceptron (which we refer to as latent learning network) to generate "novel" target latent vectors. The source-generator uses them to create new target domain images. In our method, the generator is fixed and not trainable. Latent learning network and discriminator are our only trainable module.

2.2 Background

2.2.1 Generative Adversarial Networks (StyleGAN)

We use StyleGAN2 [Karras et al., 2020a] as the backbone for generator. StyleGAN2 has two latent spaces: (i) initial latent space, $\mathcal{Z} \subseteq \mathbb{R}^{512}$, and (ii) intermediate latent space, $\mathcal{W} \subseteq \mathbb{R}^{512}$. An 8-layer MLP is used to map $\mathbf{z} \in \mathcal{Z}$ to $\mathbf{w} \in \mathcal{W}$. To begin with, Manipulating an image requires finding the corresponding latent code. Previous research [Abdal et al., 2019] suggests that it may not be possible to embed a given image directly into the $\mathcal{Z}$ or $\mathcal{W}$ space by using a common latent representation; the results improve if a separate code is chosen for each layer of the StyleGAN. We call this latent space as extended intermediate latent space and denote it as $\mathcal{W}^+$.

2.2.2 Problem Formulation

Consider a StyleGAN2 model with generator G_{θ_G} having parameters θ_G and discriminator D_{θ_D} with parameters θ_D. Let $p_S(\boldsymbol{s})$ be the underlying distribution of the source data $\mathcal{S}$ on which it was trained. Let the extended intermediate latent space of G_{θ_G} be represented by $\mathcal{W}^+$. We are given few samples from target data $\mathcal{T} \sim p_T(\boldsymbol{t})$. Our objective is to use the trained networks G_{θ_G}, D_{θ_D} and target images $\mathcal{T}$ to generate samples $\mathcal{I} \sim p_T(\boldsymbol{t})$. Note, $G_{\theta_G}(\boldsymbol{w})$ represents the images generated by the generator G_{θ_G}, and $\boldsymbol{w} \in \mathcal{W}^+$ as the input from the extended latent space.

2.3 Latent Learning Network

During inference/generation, our objective is to find a latent vector $\boldsymbol{w}_L \in \mathcal{W}^+$ that lies on the target manifold. To achieve this, we use a feed forward network with ReLU activations which we call the latent learner L_{θ_L} with parameters θ_L. The input to this network is a random vector $\mathcal{X}$ where $\mathcal{X} \sim \mathcal{N}(0, I)$ is sampled from the normal distribution with arbitrary dimensions. Therefore, the objective is to obtain a $\boldsymbol{w}_L = L_{\theta_L}(\mathcal{X})$ such that $G_{\theta_G}(\boldsymbol{w}_L) \sim p_T(\boldsymbol{t})$.

To train L_{θ_L}, we enforce target domain attributes on $G_{\theta_G}(\boldsymbol{w}_L)$ using two-component losses. First, standard style loss [Gatys et al., 2016] between $\boldsymbol{t}$ and $G_{\theta_G}(\boldsymbol{w}_L)$ captures target domain image characteristics. The second component of the loss function is an adversarial loss: the discriminator, D_{θ_D}, of StyleGAN2 acts as a critic that aims to discriminate between the generated image, $G_{\theta_G}(\boldsymbol{w}L)$, and the original target image, $\boldsymbol{t} \in \mathcal{T}$, whereas L_{θ_L} aims to fool the discriminator. We are fine-tuning D_{θ_D} and L_{θ_L} while leaving G_{θ_G} untouched. Our training objective is as follows:

$$\theta_L^*, \theta_D^* = \arg\min_{\theta_L}\max_{\theta_D}\left(\mathcal{L}_{style} + \mathcal{L}_{adv}\right) \tag{1}$$

$$\mathcal{L}_{style} = \mathop{\mathbb{E}}_{\mathcal{X},\boldsymbol{t}}\left[\sum_l \frac{\beta_l}{R_l C_l}||A^{l;G_{\theta_G}(L_{\theta_L}(\mathcal{X}))} - A^{l;\boldsymbol{t}}||_2^2\right] \tag{2}$$

$$\mathcal{L}_{adv} = \mathop{\mathbb{E}}_{\boldsymbol{t}}\left[\log D_{\theta_D}(\boldsymbol{t})\right] + \mathop{\mathbb{E}}_{\mathcal{X}}\left[\log(1 - D_{\theta_D}(G_{\theta_G}(L_{\theta_L}(\mathcal{X}))))\right] \tag{3}$$

Here, $\mathcal{L}_{style}$ is the style loss, R_l and C_l are dimensions of feature map of l^{th}-layer, A^l denotes gram matrix of corresponding feature maps as shown in [Gatys et al., 2016]. We have used VGG-19 model to get the feature maps. β_l is the hyperparameter corresponding to l^{th}-layer, which is set to 1 for all l. $\mathcal{L}_{adv}$ is the adversarial loss explained above. While training, generator parameters θ_G are kept constant, but the gradient is propagated to the latent learner. To update the latent learner, L_{θ_L}, we use non-saturating adversarial loss as outlined in [Goodfellow et al., 2014]. Once the latent learner converges, its output $\boldsymbol{w}_L^* = L_{\theta_L^*}(\mathcal{X})$ is used as the input to the generator to obtain the final generated target images $\mathbf{t}^* = G_{\theta_G}(\boldsymbol{w}_L^*)$. Figure 1 presents an illustration of the proposed method.

2.3.1 Inference-stage latent learner retraining

We train the latent learner afresh during inference-stage, whenever we desire to generate a batch of images from the target domain. Note that the same set of target samples are used for training L_{θ_L} every time, albeit it is trained afresh with different initialization. This would ensure that L_{θ_L} would not overfit on the few target data points (unlike in [Wang et al., 2020]), thus inducing diversity in the generated samples. This procedure nevertheless, comes at the cost of an increased inference time which is manageable in many practical settings since the latent learner is a simple 3-layer MLP.

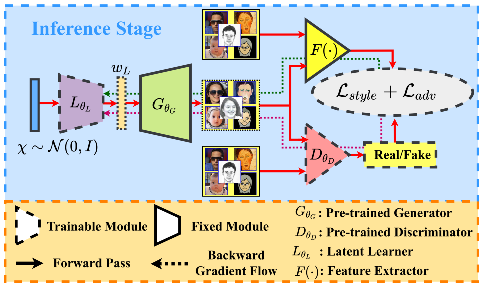

Inference Stage
w_L
L_{θ_L}
G_{θ_G}
$\chi \sim \mathcal{N}(0, I)$
$F(\cdot)$
$\mathcal{L}_{style} + \mathcal{L}_{adv}$
D_{θ_D}
Real/Fake
Trainable Module
Fixed Module
Forward Pass
Backward Gradient Flow
G_{θ_G}: Pre-trained Generator
D_{θ_D}: Pre-trained Discriminator
L_{θ_L} : Latent Learner
$F(\cdot)$: Feature Extractor

Figure 1: Given a trained StyleGAN [Karras et al., 2019, 2020a] on a source dataset and a few examples from the target domain, our method trains a latent-learner L_{θ_L} (a MLP) to generate a new latent code starting from a random Gaussian noise vector. The generated latent code is then fed to the source Generator to generate a 'novel' target domain sample.

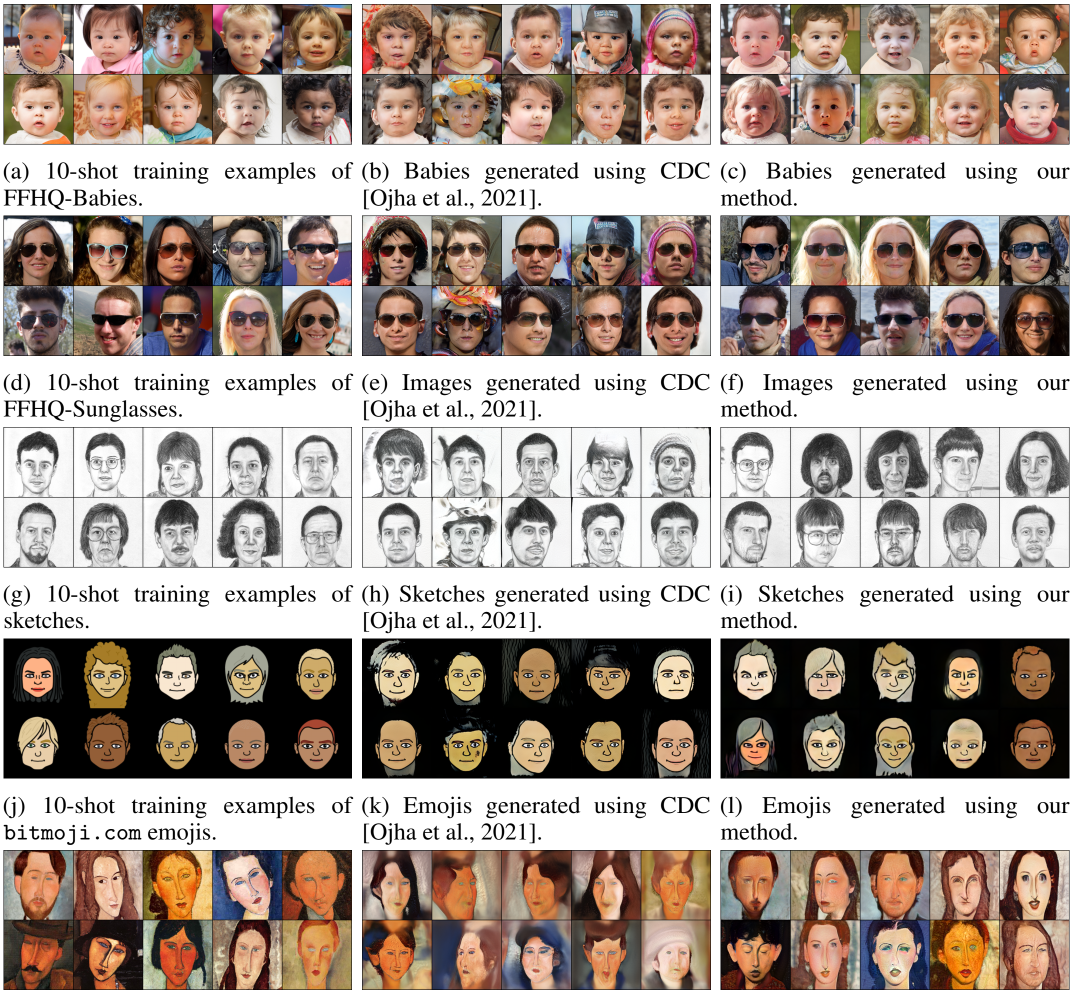
2.4 Implementation Details

The proposed method is inference time optimization. During inference, we optimise the latent learner's parameter space using style (Eq. 2) and adversarial loss (Eq. 3). We use VGG-19 model to calculate style loss; the layers utilised to extract feature maps are - {conv1_2, conv2_2, conv3_4, conv4_4, conv5_4}. Unless otherwise specified, the latent learner is a 3-layer MLP with 512 neurons per layer. The hidden layers use ReLU activation, while the last layer has none. We use a batch size of 8.

3 Experiments and Results

3.1 Datasets

Following previous work [Li et al., 2020, Ojha et al., 2021], we consider Flickr Faces HQ (FFHQ) [Karras et al., 2019] as one of the source domain datasets and adapt to the following target domains: (i) FFHQ-Babies [Ojha et al., 2021], (ii) FFHQ-Sunglasses [Ojha et al., 2021], (iii) face sketches



(a) 10-shot training examples of FFHQ-Babies.

(b) Babies generated using CDC [Ojha et al., 2021].

(c) Babies generated using our method.

(d) 10-shot training examples of FFHQ-Sunglasses.

(e) Images generated using CDC [Ojha et al., 2021].

(f) Images generated using our method.

(g) 10-shot training examples of sketches.

(h) Sketches generated using CDC [Ojha et al., 2021].

(i) Sketches generated using our method.

(j) 10-shot training examples of `bitmoji.com` emojis.

(k) Emojis generated using CDC [Ojha et al., 2021].

(l) Emojis generated using our method.

(m) 10-shot training examples of Amedeo's painting.

(n) Amedeo's painting generated using CDC [Ojha et al., 2021].

(o) Amedeo's painting generated using our method.

Figure 2: Source domain: FFHQ. Each row represents one target domain among — Babies, Sunglasses, Sketches, Emoji and Amedeo Modigliani's face paintings. The first column presents 10-shot training examples from the target domains. The second column presents 10 images generated using CDC [Ojha et al., 2021]. The third column presents 10 images generated using our proposed method.

[Wang and Tang, 2009], (iv) emoji faces from `bitmoji.com` API [Taigman et al., 2016, Hua et al., 2017], and (v) portrait paintings from the artistic faces dataset [Yaniv et al., 2019].

Next, we employ LSUN Church [Yu et al., 2015] as a source domain and adapt to haunted houses [Ojha et al., 2021] and Van Gogh's house paintings [Ojha et al., 2021]. Both source and target domain datasets use 256x256 images. Unless otherwise noted, all our tests use 10 randomly picked target samples.

Table 1: FID scores (↓) for domains with ample data.

Method	Babies	Sunglasses	Sketches	Bitmoji
TGAN	104.79	55.61	53.41	66.69
TGAN + ADA	102.58	53.64	66.99	68.71
BSA	140.34	76.12	69.32	105.56
FreezeD	110.92	51.29	46.54	71.16
MineGAN	98.23	68.91	64.34	86.44
EWC	87.41	59.73	71.25	74.87
CDC	74.39	42.13	45.67	69.54
Proposed Method	**63.31**	**35.64**	**35.59**	**64.50**

3.2 Methodology, Metrics and Baselines

Methodology: Even though very few (1, 5, or 10) target examples are used in the method for adaptation, the evaluation is conducted on a larger target set. For example, there are approximately 300, 2500, 2700 and unlimited examples in the sketches [Wang and Tang, 2009], FFHQ-Babies [Ojha et al., 2021], FFHQ-Sunglasses [Ojha et al., 2021], and emoji [Taigman et al., 2016, Hua et al., 2017] datasets, respectively.

Table 2: Density (↑) and Coverage (↑) scores.

Method	**Babies**		**Sketches**	
	Density	**Coverage**	**Density**	**Coverage**
TGAN	0.379	0.250	0.221	0.401
TGAN + ADA	0.434	0.285	0.193	0.374
FreezeD	0.418	0.217	0.415	0.436
MineGAN	0.803	0.125	0.394	0.263
EWC	0.301	0.325	—	—
CDC	0.690	0.467	0.149	0.492
Proposed Method	**1.118**	**0.611**	**0.445**	**0.716**

Table 3: Intra-cluster pairwise LPIPS distance (↑)

Method	**Amedeo's Paintings**	**Sketches**
TGAN	0.41	0.39
TGAN + ADA	0.51	0.41
BSA	0.39	0.35
FreezeD	0.40	0.39
MineGAN	0.42	0.40
CDC	0.60	0.45
Proposed Method	**0.61**	**0.48**

Metrics: We compute and report Fréchet Inception Distance (FID) [Heusel et al., 2017] and density and coverage metrics [Naeem et al., 2020] to quantify quality and diversity, respectively. Further, we report Learned Perceptual Image Patch Similarity (LPIPS) [Zhang et al., 2018] metric, which gives an idea about overfitting on the small amount of target data as shown in [Ojha et al., 2021].
Baselines: We compare our method against the following baselines - Transferring GAN (TGAN) [Wang et al., 2018], Batch Statistics Adaptation (BSA) [Noguchi and Harada, 2019], MineGAN [Wang et al., 2020], Freeze-D [Mo et al., 2020], Non-leaking Adaptive Data Augmentation [Karras et al., 2020b, Zhao et al., 2020] (TGAN + ADA), Elastic Weight Consolidation (EWC) [Li et al., 2020], and Few-shot Image Generation via Cross-domain Correspondence (CDC) [Ojha et al., 2021].

3.3 Results

As shown in Table 1, our strategy outperforms state-of-the-art methods on all four datasets. From Table 2, it's clear that the proposed strategy generates diverse, high-quality samples. As in [Ojha et al., 2021], we allocate 1000 generated images to one of $k = 10$ possible clusters based on the lowest LPIPS distance [Zhang et al., 2018]. Next, we compute the average the pair-wise LPIPS metric among cluster members. Finally, we average over k clusters. A lower value of this measure indicates less diversified images. It can be seen from Table 3, our technique provides the best intra-cluster pair-wise LPIPS distance for Amedeo Modigliani's paintings and sketches datasets. Figure 2 presents a few samples generated using CDC [Ojha et al., 2021] and our proposed method when adapting a FFHQ generator to various target domains — Babies, Sunglasses, Sketches, and Amedeo Modigliani's face painting. It can be seen that the images generated using proposed method are crisp and diverse. Refer to the supplementary material for church to haunted house and church to Van Gogh's house painting adaptation.

Table 4: Effect of k in k-shot adaptation on generation quality as measured using FID score (↓).

Method	**1-shot**		**5-shot**		**10-shot**	
	Babies	**Sketches**	**Babies**	**Sketches**	**Babies**	**Sketches**
CDC	105.58	81.95	73.63	51.01	74.39	45.67
Proposed Method	**105.13**	**79.20**	**65.47**	**41.88**	**62.14**	**35.59**

3.4 Ablation Studies

Number of Target Examples: So far, we have used 10 target domain images. Table 4 shows how number of training data affects generation quality. We compare the proposed approach to SOTA CDC [Ojha et al., 2021]. Generation quality improves with k, and the proposed technique performs better in almost all cases. For qualitative analysis, see supplementary.

4 Conclusion

In this study, we demonstrate that existing source models can be used as-is (without retraining or fine-tuning) to model new distributions with less data. We feel our work is an important step towards few-shot 'generative domain adaptation,' where the same source generator can generate source domain samples and many target domain samples. Our method can also be seen as a step toward continual learning for generative tasks where the same generator can generate data from diverse domains.

References

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Proc. of NeuRIPS*, 2014.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *Proc. of ICLR*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proc. of CVPR*, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *Proc. of CVPR*, 2020a.

Vadim Sushko, Jürgen Gall, and Anna Khoreva. One-Shot GAN: Learning to Generate Samples from Single Images and Videos. In *Proc. of CVPRW*, 2021.

Tasfia Shermin, Shyh Wei Teng, Ferdous Sohel, Manzur Murshed, and Guojun Lu. Bidirectional mapping coupled gan for generalized zero-shot learning. *IEEE Transactions on Image Processing*, 31:721–733, 2021.

Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30:1882–1897, 2021.

Ning Rao, Tao Lu, Qiang Zhou, Yanduo Zhang, and Zhongyuan Wang. Seeing in the dark by component-gan. *IEEE Signal Processing Letters*, 28:1250–1254, 2021.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable Augmentation for Data-Efficient GAN Training. In *Proc. of NeurIPS*, 2020.

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

Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. MineGAN: effective knowledge transfer from GANs to target domains with few images. In *Proc. of CVPR*, 2020.

Yijun Li, Richard Zhang, Jingwan (Cynthia) Lu, and Eli Shechtman. Few-shot Image Generation with Elastic Weight Consolidation. In *Proc. of NeurIPS*, 2020.

Utkarsh Ojha, Yijun Li, Cynthia Lu, Alexei A. Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proc. of CVPR*, 2021.

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proc. of ICCV*, pages 4432–4441, 2019.

Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3), 2021. ISSN 0730-0301. doi: 10.1145/3447648.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proc. of CVPR*, 2021.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Trans. Graph.*, 40(4), 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459838.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *Proc. of CVPR*, 2016.

Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1955–1967, 2009. doi: 10.1109/TPAMI.2008.222.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised Cross-Domain Image Generation. *arXiv preprint arXiv:1611.02200*, 2016.

Xinru Hua, Davis Rempe, and Haotian Zhang. Unsupervised Cross-Domain Image Generation. *Stanford Course, CS229 Project*, 2017.

Jordan Yaniv, Yael Newman, and Ariel Shamir. The face of art: Landmark detection and geometric style in portraits. *ACM Trans. Graph.*, 38(4), 2019. doi: 10.1145/3306346.3322984.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, 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.

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 *Proc. of NeurIPS*, 2017.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proc. of ICML*, 2020.

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

A. Noguchi and T. Harada. Image Generation From Small Datasets via Batch Statistics Adaptation. In *Proc. of ICCV*, 2019.

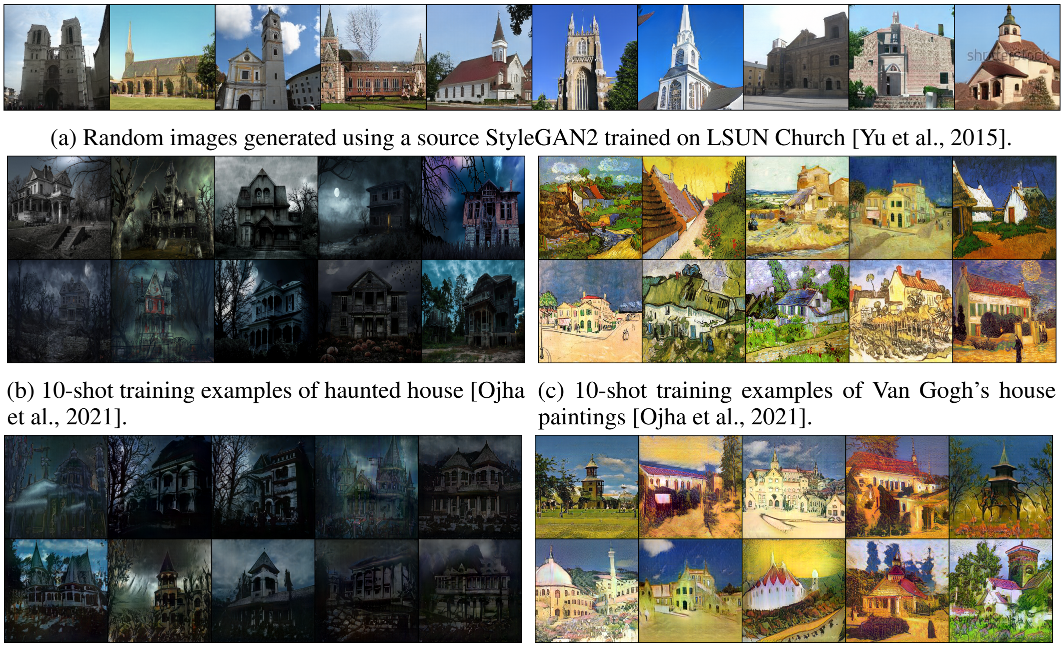
Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the Discriminator: a Simple Baseline for Fine-Tuning GANs. In *CVPR AI for Content Creation Workshop*, 2020.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. of NeurIPS*, 2020b.

Appendix

A Qualitative Analysis of Impact of k on Generation for k-shot Adaptation

In the main paper, we have provided quantitative comparison of generation quality as k in k-shot generation is varied from 1 to 10. Here, we present some visual samples of 1-shot and 5-shot adaptation for FFHQ → Babies. Figure 4c and 5c presents newly generated baby samples using the proposed method under 1-shot and 5-shot settings respectively. As can be seen, variation in the generated images increases in the 5-shot adaptation. As compared to the current SOTA CDC [Ojha et al., 2021], our method generates images that are more crisp and exhibits more variety.



(a) Random images generated using a source StyleGAN2 trained on LSUN Church [Yu et al., 2015].

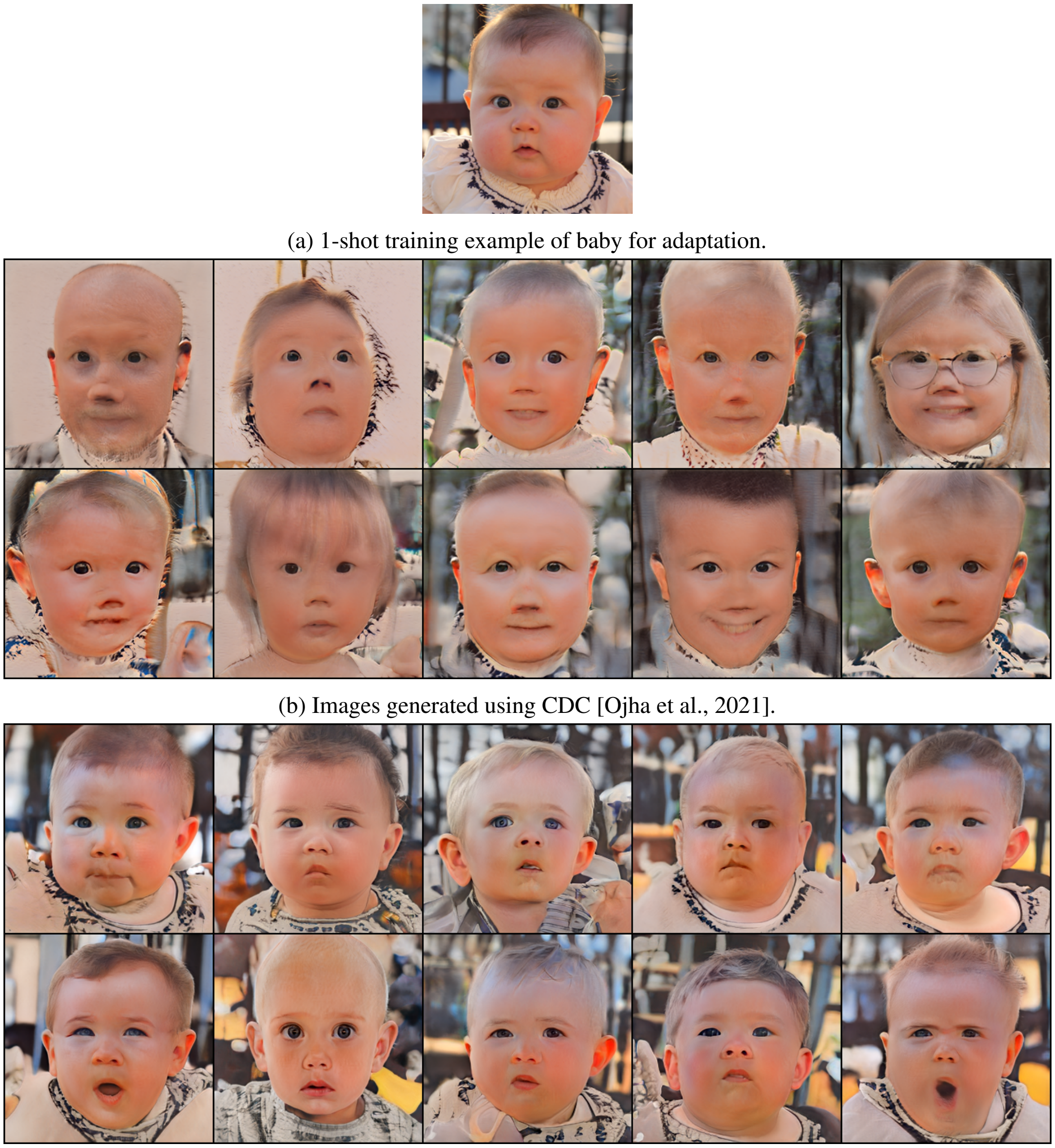
(b) 10-shot training examples of haunted house [Ojha et al., 2021].

(c) 10-shot training examples of Van Gogh's house paintings [Ojha et al., 2021].

(d) Haunted houses generated by our proposed method.

(e) Van Gogh style house paintings generated by proposed method.

Figure 3: Source domain: LSUN Church [Yu et al., 2015]. Target Domain: Haunted house [Ojha et al., 2021] and Van Gogh's house paintings [Ojha et al., 2021].

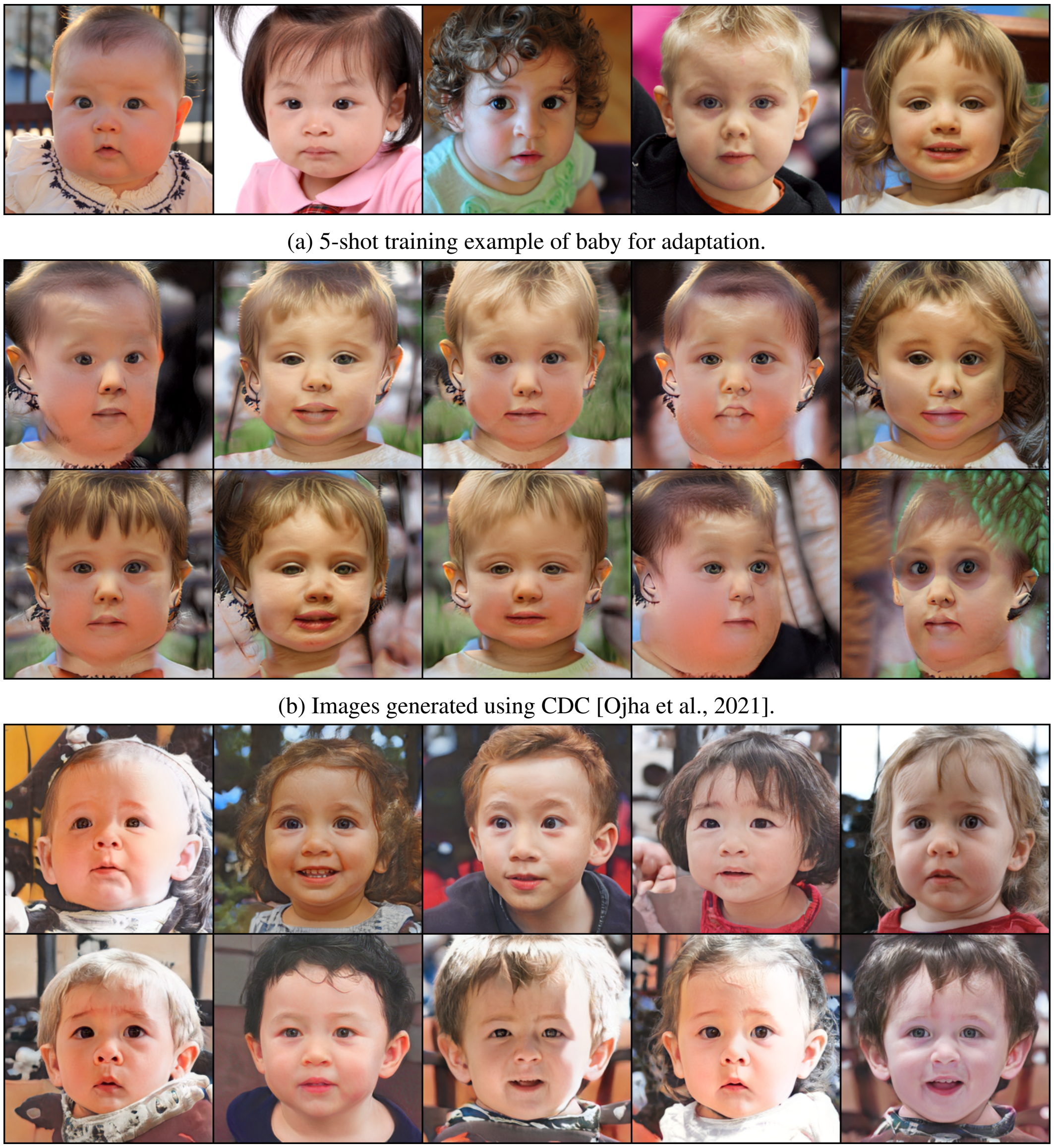


(a) 1-shot training example of baby for adaptation.

(b) Images generated using CDC [Ojha et al., 2021].

(c) Images generated using proposed method.

Figure 4: Examples of images generated after 1-shot adaptation.



(a) 5-shot training example of baby for adaptation.

(b) Images generated using CDC [Ojha et al., 2021].

(c) Images generated using proposed method.

Figure 5: Examples of images generated after 5-shot adaptation.