MineGAN: effective knowledge transfer from GANs to target domains with few images

Vladimir Fedoseev
s6vlfedo@uni-bonn.de

Institute of Informatics,
University of Bonn

Transfer learning is researched less in GANs than in discriminative models. However, it is promising: training GANs from scratch is costly. Collecting vast training data is often hard for real problems, and knowledge transfer from a GAN pre-trained on a 'source' dataset to a domain represented by a *small* 'target' set is the focus of the paper.

GANs are powerful models for generating high-quality images that resemble the training ones. We follow the WGAN-GP realisation [1]. *Generator* G samples noise vectors z and transforms them into outputs, while *critic* D gives the outputs and the training images 'reality' scores. G tries to output images to score as much as the real ones, and D competes and improves as a critic. The loss (gradient penalty omitted) is as follows:

$$L = E_{\tilde{x}\sim P_g}[D(\tilde{x})] - E_{x\sim P_r}[D(x)] \tag{1}$$

The first attempt at transfer learning in GANs, TransferGAN [3], just fine-tuned all parameters, requiring target sets of thousands of images and leading to mode collapse. BSA [2] adds scale and shift parameters to the convolution filters and learns only them and batch statistics. It reduces overfitting and allows as few as 25 target images, but blurs the outputs.

This paper proposes a new approach, *mining* GANs before fine-tuning them, as well as an extension of this method for transferring knowledge from multiple GANs, which has not been done yet. Our experiments show significant improvement in quality and variance. We distinguish *on-manifold* and *off-manifold* generation, depending on whether the output distribution of the original GAN p_g has significant or negligible overlap with the target distribution p^{τ}_{data}.

The mining step narrows the output distribution p_g to outputs that resemble the target dataset D_τ. It learns a transformation of $p_z(z)$, so the model samples only from regions which lead to images as close to the target p^{τ}_{data} as the GAN can produce. This way, a new prior $p^{\tau}_z(z)$ and a resulting output distribution p^{τ}_g are learned. Transformation of $p_z(z)$ is done by *miner*, a multilayer perceptron M, which takes a new latent variable $u \sim p_z(u)$ and gives z values to the GAN. By mining we mean learning M: training the model on D_τ, while fixing all the weights except ones in the miner. D and G losses then are written as:

$$L^M_D = E_{u\sim p_z(u)}[D(G(M(u)))] - E_{x\sim p^{\tau}_{data}(x)}[D(x)] \tag{2}$$

$$L^M_G = -E_{u\sim p_z(u)}[D(G(M(u)))] \tag{3}$$

We show that mining alone performs on par with state of the art, and treat it as a separate method, *MineGAN (w/o FT)*. The full MineGAN method includes subsequent fine-tuning, when learning on D_τ continues, but D and G are released to be trainable together with M. Fine-tuning benefits from the mining: it reduces the divergence between the output distribution and the target, making the fine-tuning more efficient and less prone to overfitting.

MineGAN also extends to **mining conditional GANs**, which condition the output by an additional input, class label c. Conditional MineGAN, however, does not require labels: it learns c values corresponding to each z with another miner that takes the same u values.

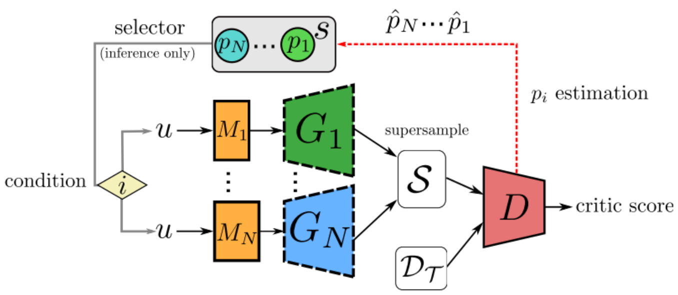

Figure 1: MineGAN model for multiple pre-trained GANs

Our approach to the general case of **mining multiple GANs** (Figure 1) uses the relevant knowledge of N pre-trained generators $\{G_i\}$. Each G_i has its own miner M_i. Just one critic from N pre-trained ones is used, and we pick one randomly, which is found to have no effect. D needs to observe the frequencies of generated images to encourage the right output distribution, so we train only the highest scoring G at each iteration. Minibatches consist of *supersamples* of outputs of each G_i, i.e. $S = \{G_i(z)|z \sim p^i_z(z); i = 1,...,N\}$.

The *selector* is added to learn the relevances of each G_i to the D_τ. They are used as probabilities $p_1...p_N$ of picking each G_i for inference. This emphasises that we do not combine knowledge of $\{G_i\}$ during an inference pass. Selector, however, ensures correct frequencies in the output: when D_τ is images from the manifolds of G_i, p_i reflect their proportion.

The paper contains detailed method and results for the following knowledge transfer experiments: (i) from a GAN to a D_τ of varying size (Figure 2); (ii) from a GAN to a D_τ of only 25 images (Figure 3, Table 1); (iii) from a conditional GAN to a D_τ with 500 images per category; (iv) from 2 GANs when the D_τ of 200 images lies on both manifolds, varying the ratio of these manifolds in D_τ; (v) from 4 GANs to 2 separate D_τ of 200 images, when 2 GANs generate far from D_τ (recognised by the selector, contributing little but significant knowledge). The results are compared to training from scratch, [2], [3], and iterative 'scratch' methods.

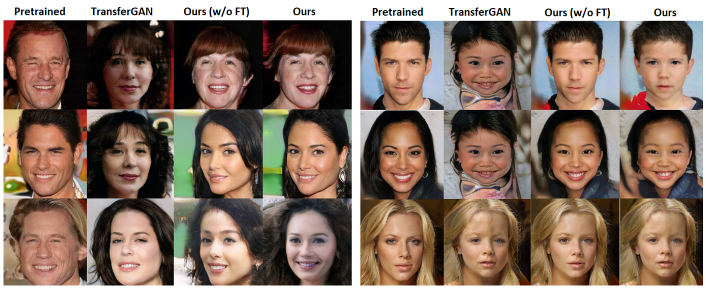

Figure 2: Results with D_τ of 100 images, on- and off-manifold

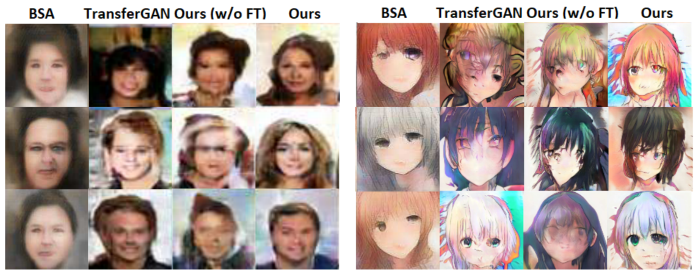

Figure 3: Results with D_τ of 25 images, on- and off-manifold

	On-manifold		Off-manifold	
	KMMD	MV	KMMD	MV
TransferGAN [3]	0.346	0.506	0.347	0.785
BSA [2]	0.345	0.785	0.342	0.908
MineGAN (w/o FT)	0.349	0.774	0.347	0.891
MineGAN	**0.337**	**0.812**	**0.334**	**0.934**

Table 1: Quantitative results with D_τ of 25 images. KMMD - resemblance 'distance' to D_τ, MV - variance of outputs

[1] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, 2017.

[2] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[3] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.