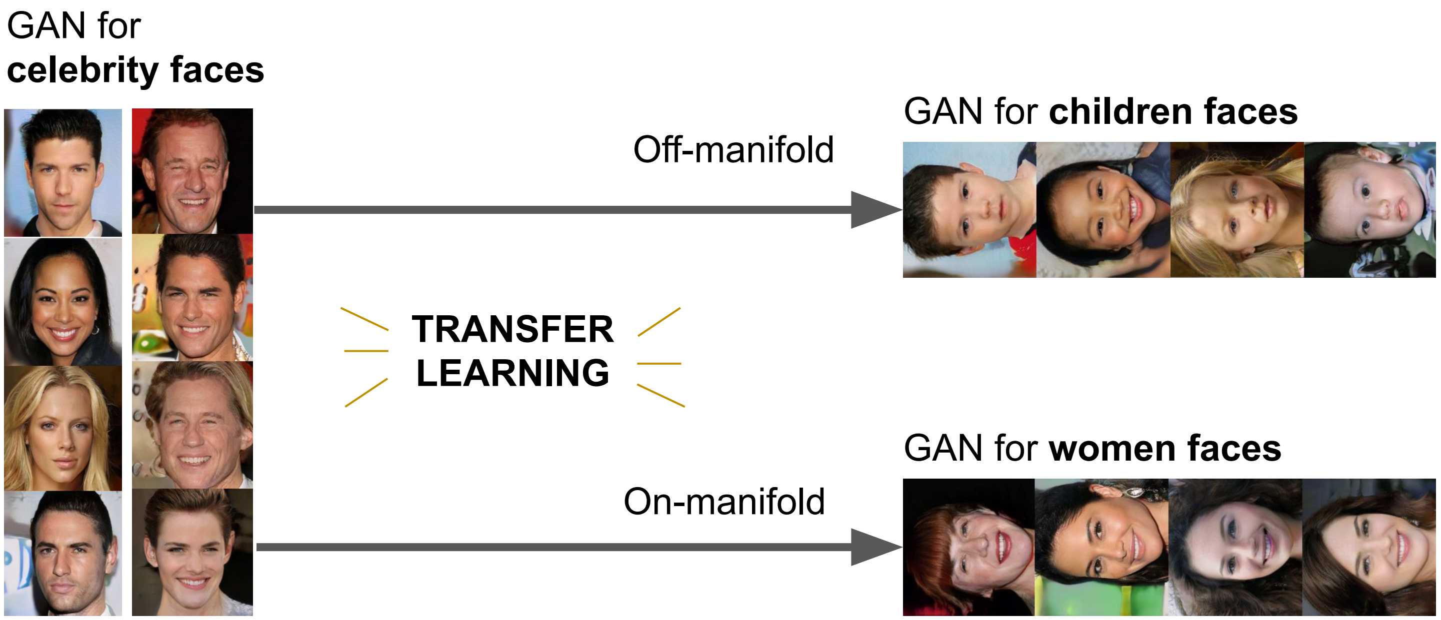


MineGAN: effective knowledge transfer from GANs to target domains with few images

(Wang et al., 2020)

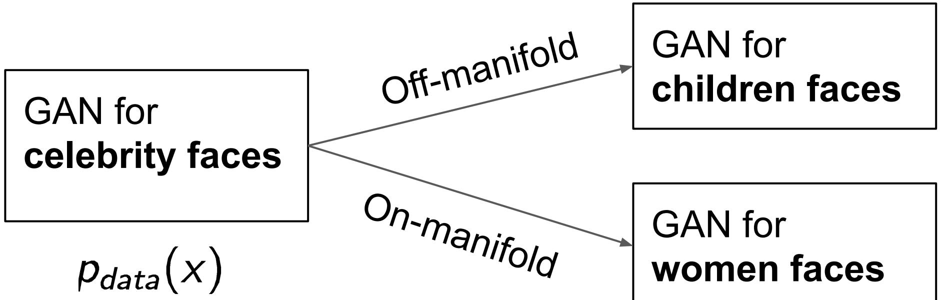
Vladimir Fedoseev
Computer Vision Seminar SS 2020

Problem - transfer from single GAN

Given:

- a target real images distribution $p^{\mathcal{T}}_{data}(x)$ from which a small set $\mathcal{D}_{\mathcal{T}}$ is sampled
- D and G trained to approximate a source real images distribution $p_{data}(x)$ with generative $p_g(x)$

Produce a new model to approximate $p^{\mathcal{T}}_{data}(x)$ with new $p^{\mathcal{T}}_g(x)$ using G, D, and $\mathcal{D}_{\mathcal{T}}$



$p^{\mathcal{T}}_{data}(x)$ has negligible overlap with $p_{data}(x)$

$p^{\mathcal{T}}_{data}(x)$ has significant overlap with $p_{data}(x)$

Problem - transfer from multiple GANs

Given:

- a target real images distribution $p^{\mathcal{T}}_{data}(x)$ from which a small set $\mathcal{D}_{\mathcal{T}}$ is sampled

- N generators $\{G_i\}$ and N discriminators $\{D_i\}$ pretrained

Produce a new model to approximate $p^{\mathcal{T}}_{data}(x)$ with new $p^{\mathcal{T}}_{g}(x)$ using $\{G_i\}$, $\{D_i\}$, and $\mathcal{D}_{\mathcal{T}}$

Contributions

1. New method of knowledge transfer for GANs for target distributions represented by a small training set

2. First method to transfer knowledge from several GANs to one GAN

3. Outperforming other approaches in transfer for different GANs

Motivation - GANs

Generative models to induce
complex, high-dimensional data distribution
(e.g. a set of images) and generate new samples

Variety of other tasks: super-resolution, style transfer, image inpaiting, text to image, ...

Motivation - GANs

images from thispersondoesnotexist.com (Karras et al, 2020)

Motivation - knowledge transfer for GANs

1. Training GANs from scratch require a lot of time and images
   (Progressive GANs - 30K images, a month on an NVIDIA Tesla V100)

- Sometimes images from target distribution are rare or costly to obtain
- Applications might require fast training on a specific target distribution

2. GAN training is prone to overfitting, mode collapse, not stable gradients ...

=> Need to quickly retrain well-trained GANs for new distributions represented by small sets

Motivation - knowledge transfer from several GANs

- Target distribution is a mixture or intersection of (parts of) distributions of available pre-trained GANs

GAN for Buses → GAN for **red** cars and buses
GAN for cars → GAN for **red** cars and buses

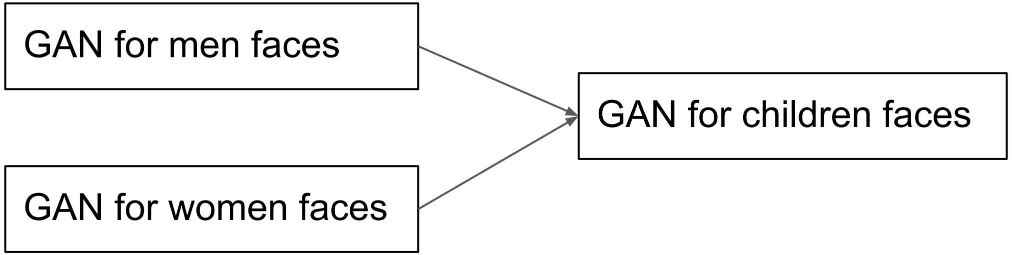
GAN for flowers → GAN for birds and flowers
GAN for birds → GAN for birds and flowers

- Target distribution includes distributions of pre-trained GANs

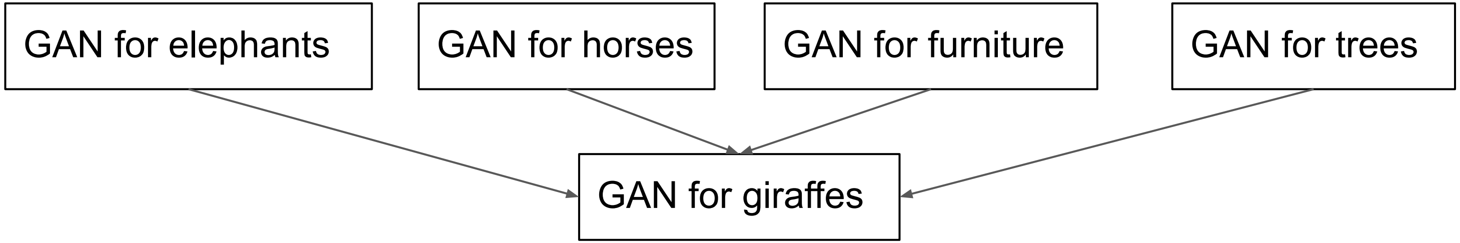
GAN for kitchens → GAN for living interior photos
GAN for bedrooms → GAN for living interior photos

Motivation - knowledge transfer from several GANs

- Target distribution is off-manifold for pre-trained GANs, but close



- Target distribution is mainly off-manifold for pre-trained GANs, but mixing them will increase the amount of potentially relevant knowledge



GAN - original formulation

D score for train image

expectations over minibatches

D score for generated image

$$\min_G \max_D \mathbb{E}_{x\sim\mathbb{P}_r}\left[\log\left(D\left(x\right)\right)\right] + \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}\left[\log\left(1 - D\left(\tilde{x}\right)\right)\right]$$

image drawn from
training ones

image drawn from
generated ones

$$\tilde{x} = G\left(z, \theta^g\right), z \sim \mathbb{P}_z$$

generator
network

$$G\left(z, \theta^g\right)$$

discriminator
network

$$\mathrm{D}(\mathrm{x}) = \mathrm{A}(\mathrm{f}\left(\mathrm{x}, \theta^d\right))$$

function to map to (0, 1), like sigmoid

latent noise vector
drawn from some distribution

Goodfellow et al. (2014). Generative Adversarial Nets

GAN - original formulation

$$\min_G \max_D \mathbb{E}_{x \sim \mathbb{P}_r} \left[\log\left(D\left(x\right)\right)\right] + \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} \left[\log\left(1 - D\left(\tilde{x}\right)\right)\right]$$

$G\left(z, \theta^g\right)$

find parameters θ^g to maximize $D\left(\tilde{x}\right)$

$\mathrm{D}(\mathrm{x}) = \mathrm{A}(\mathrm{f}\left(\mathrm{x}, \theta^d\right))$

find parameters θ^d to minimize $D\left(\tilde{x}\right)$ and maximize $D\left(x\right)$

Goodfellow et al. (2014). Generative Adversarial Nets

GAN - WGAN-GP used by MineGAN

$$\min_G \max_D \mathbb{E}_{x\sim\mathbb{P}_r}\left[\log\left(D\left(x\right)\right)\right] + \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}\left[\log\left(1 - D\left(\tilde{x}\right)\right)\right]$$

$$\lim_{\|D-D^*\|\to 0} \nabla_\theta \mathbb{E}_{z\sim p(z)}[\log(1 - D(\tilde{x}))] = 0$$ (Arjovsky & Bottou, 2017)

$$L = \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] + \underline{\lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]}$$

"Critic": not [0, 1] output, no logs

Critic gradient penalty
to enforce Lipschitz-1 constraint

Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks.
Gulrajani et al. (2017). Improved training of wasserstein gans

GAN - WGAN-GP used by MineGAN

$$L = \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r}[D(x)] + \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]$$

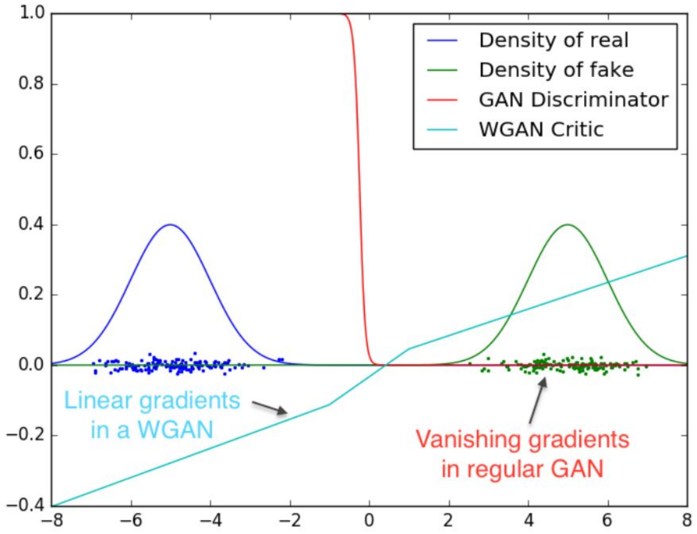
- Soft, two-sided penalty (not necessarily below 1)
- $\hat{x} \sim \mathbb{P}_{\hat{x}}$ is sampled uniformly along straight lines between pairs of points $\tilde{x} \sim \mathbb{P}_g$ —— $x \sim \mathbb{P}_r$
- Penalizing w.r.t each input individually, so no batch normalization in D (e.g. layer normalization instead)

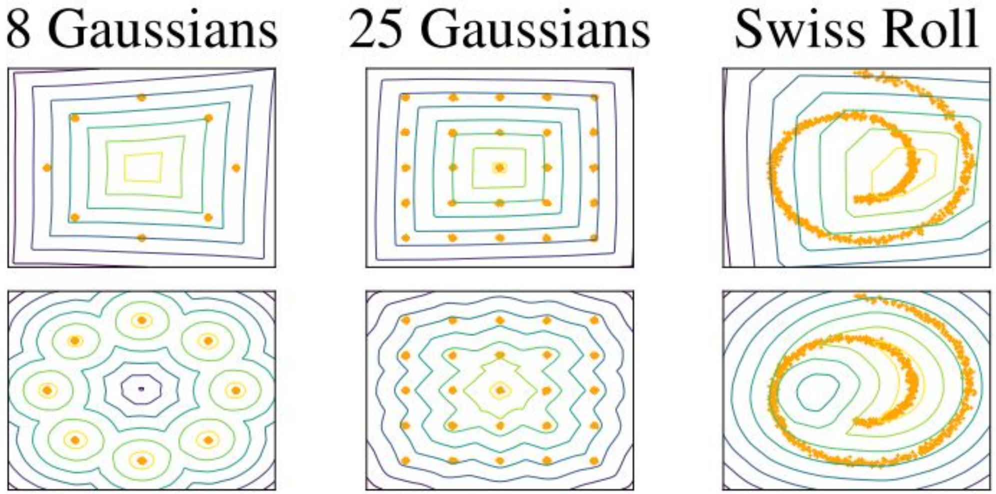
Gulrajani et al. (2017). Improved training of wasserstein gans

GAN - WGAN-GP used by MineGAN

WGAN:
Less gradient vanishing/exploding
No mode collapse

WGAN-GP:
Learning more complex functions
Even less gradient vanishing/exploding





Arjovsky, Chintala, & Bottou. (2017). Wasserstein gan

Gulrajani et al. (2017). Improved training of wasserstein gans

Other GANs - BigGAN (MineGAN is also applied to)

- Increase batch size (i.e. 8 times)
- Increase number of channels in each layer by 50%
- If conditional, project label embedding (and skip-z) to each BN layer
- "Truncation trick": trained with **z ~ N(0, I)**, during inference resample **z** if it falls outside a thresholded range.
  It increases image quality, but reduces variety.
  Threshold maintains the trade-off

need to make **G** smooth, so **z** maps to good output everywhere

Add regularizer to loss:

$$R_\beta(W) = \beta \left\| W^T W \odot (1 - I)) \right\|_F^2$$

Brock, Donahue & Simonyan. (2018). Large scale gan training for high fidelity natural image synthesis

Other GANs - Progressive GAN (MineGAN is also applied to)

- WGAN loss
- Start training with small res, iteratively add larger layers
- To increase variation, at the end of the D:
  - compute SD over a minibatch for every channel and location
  - average all these values to a single one
  - add one more channel with this mean SD in all locations
- For magnitudes of G and D not to grow out of control, normalize across features at each pixel in G after each conv layer:

$$b_{x,y} = \frac{a_{x,y}}{\sqrt{\frac{1}{N}\sum_{j=0}^{N-1}(a^j_{x,y})^2 + \epsilon}}$$

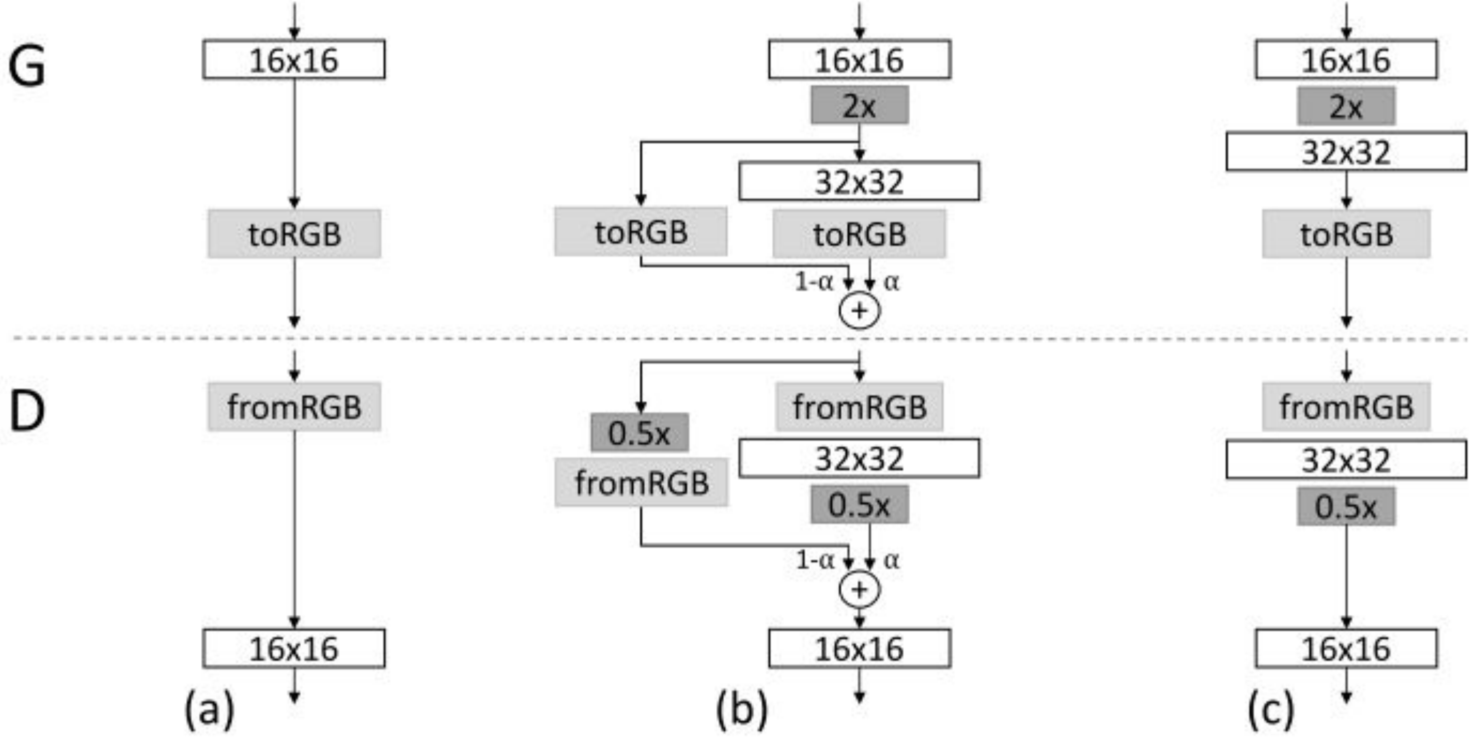
original feature vector

10^{-8}

features

Karras et al. (2017). Progressive growing of gans for improved quality, stability, and variation

Other GANs - Progressive GAN (MineGAN is also applied to)



Karras et al. (2017). Progressive growing of gans for improved quality, stability, and variation

Other GANs - SNGAN (MineGAN is also applied to)

- Spectral normalization to enforce Lipschitz-1 constraint on **D** and to improve the image quality (better than weight normalization and gradient penalty)

Normalized **D** weight matrix

$$\bar{W}_{SN}(W) = \frac{W}{\sigma(W)}$$

D weight matrix

$$max_{\|h\|_2 \leq 1} \|Wh\|_2$$

spectral norm, largest singular value

Miyato et al. (2018). Spectral normalization for generative adversarial networks

Related works - TransferGAN

- First knowledge transfer for GANs - **plainly fine-tune all parameters**
- Uses WGAN-GP
- Needs 2-5 times less images than scratch to get same scores
- GANs pretrained on narrow but dense domains perform better even when they are not so related to target domain
- Conditional: **D** has an 'auxiliary classifier', outputs P(C = y|x)

$$\mathcal{L}_{AC-GAN}(G) = \mathcal{L}_{GAN}(G) - \alpha_G \mathbb{E}\left[\log\left(P\left(C = y' | G(z, y')\right)\right)\right]$$
$$\mathcal{L}_{AC-GAN}(D) = \mathcal{L}_{GAN}(D) - \alpha_D \mathbb{E}\left[\log\left(P\left(C = y | x\right)\right)\right]$$

log likelihoods of the correct class

Wang et al. (2018). Transferring GANs: generating images from limited data

Related works - BSA

- To adapt prior knowledge, **fine-tune only these G parameters:** BN + introduced scale and shift on all channels of all conv layers

$$G^{(l)}_{Adapt} = G^{(l)} \cdot \gamma^{(l)} + \beta^{(l)}$$

$$\begin{aligned} & conv(x; W) \cdot \gamma + \beta \\ &= conv(x; W \cdot \gamma + \beta) \\ &= conv(x; \{\gamma_1 W_1 + \beta_1, ..., \gamma_{c_{out}} W_{c_{out}} + \beta_{c_{out}}\}) \end{aligned}$$

applying scale and shift on the convolution result

activation strength

activation threshold

Different filters of the layer

Noguchi & Harada. (2019). Image generation from small datasets via batch statistics adaptation

Related works - BSA

- Supervised, estimating **z** of sparse training samples + updating parameters

$$L = \sum_i \frac{1}{c_x h_x w_x} ||x_i - G_{Adapt}(z_i + \epsilon)||_1$$

Pixelwise MSE

Perceptual loss

train image

G output

$$+ \sum_i \sum_{l \in layers} \frac{\lambda_C^l}{c_l h_l w_l} ||C^{(l)}(x_i) - C^{(l)}(G_{Adapt}(z_i + \epsilon))||$$

Regularize **z**
to N(0, 1)

classifier CNN

$$+ \lambda_z \left(\sum_j^k \frac{1}{d_z} \min_i ||z_i - r_j||_2^2 + \sum_i^b \frac{1}{d_z} \min_j ||z_i - r_j||_2^2\right)$$

Regularize
parameters

~ N(0, I)

$$+ \lambda_{\gamma,\beta} \sum_l \frac{1}{d_{\gamma,\beta}^l} (||\gamma_l - 1||_2^2 + ||\beta_l||_2^2)$$

Noguchi & Harada. (2019). Image generation from small datasets via batch statistics adaptation

Related works - BSA

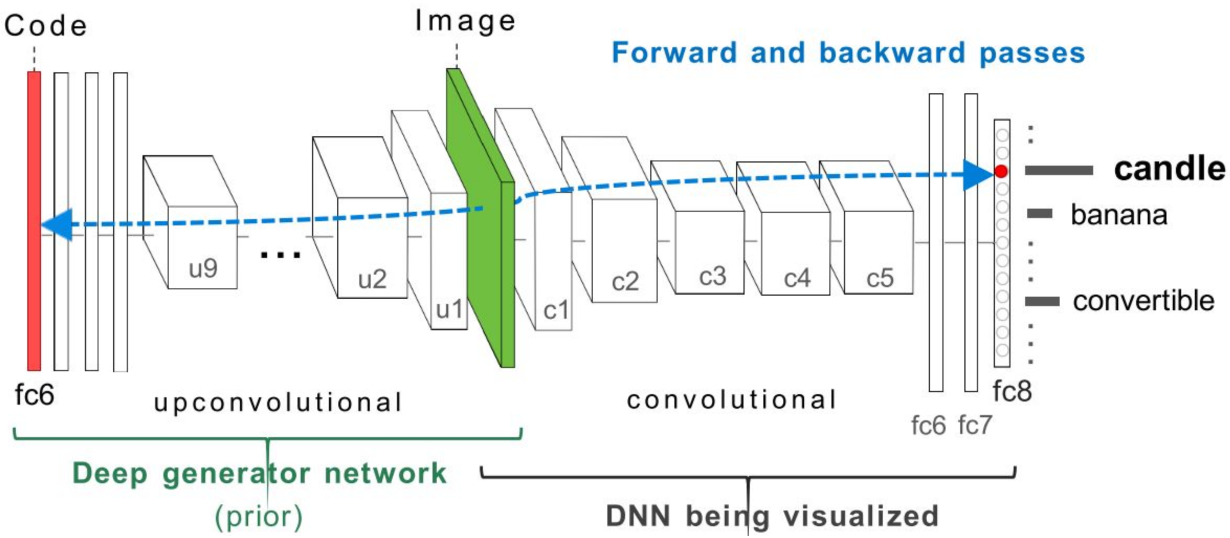
- Generator learns sparse $\mathbf{z}_i$ - $\mathbf{x}_i$ relationships.
  During inference - sample from truncated $\mathbf{z}$

- Small number of parameters avoids overfitting, suitable for very small (<100 images) target sets

- Blurred images because of MSE loss

Noguchi & Harada. (2019). Image generation from small datasets via batch statistics adaptation

Related works - iterative - DGN-AM, PPGN



$$\widehat{\mathbf{y}^l} = \arg\max_{\mathbf{y}^l}(\Phi_h(G_l(\mathbf{y}^l)) - \lambda\|\mathbf{y}^l\|)$$

DNN target neuron output Generated image Optimised code

Nguyen et al. (2017). Plug & play generative networks: Conditional iterative generation of images in latent space

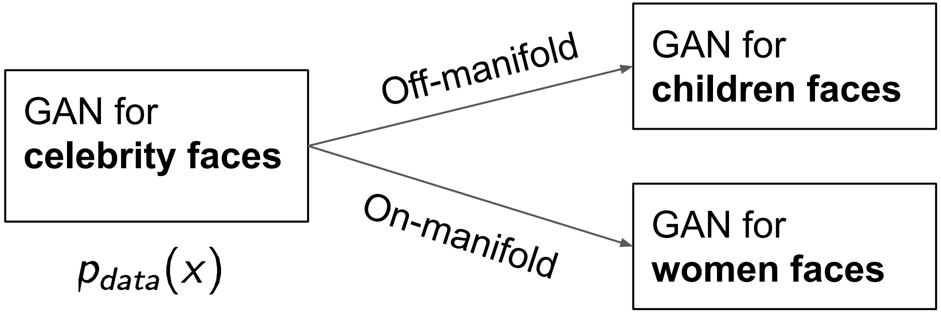
Nguyen et al. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks

Problem - transfer from single GAN

Given:

- a target real images distribution $p^{\mathcal{T}}_{data}(x)$ from which a small set $\mathcal{D}_{\mathcal{T}}$ is sampled
- D and G trained to approximate a source real images distribution $p_{data}(x)$ with generative $p_g(x)$

Produce a new model to approximate $p^{\mathcal{T}}_{data}(x)$ with new $p^{\mathcal{T}}_{g}(x)$ using G, D, and $\mathcal{D}_{\mathcal{T}}$



$p^{\mathcal{T}}_{data}(x)$ has negligible overlap with $p_{data}(x)$

$p^{\mathcal{T}}_{data}(x)$ has significant overlap with $p_{data}(x)$

MineGAN (w/o FT) - mining

W/o loss of generality, assume **z** ~ N(0, 1)

Learn $p_g^{\mathcal{T}}(x)$ by finding the regions in $p_g(x)$ that better approximate $p_{data}^{\mathcal{T}}(x)$

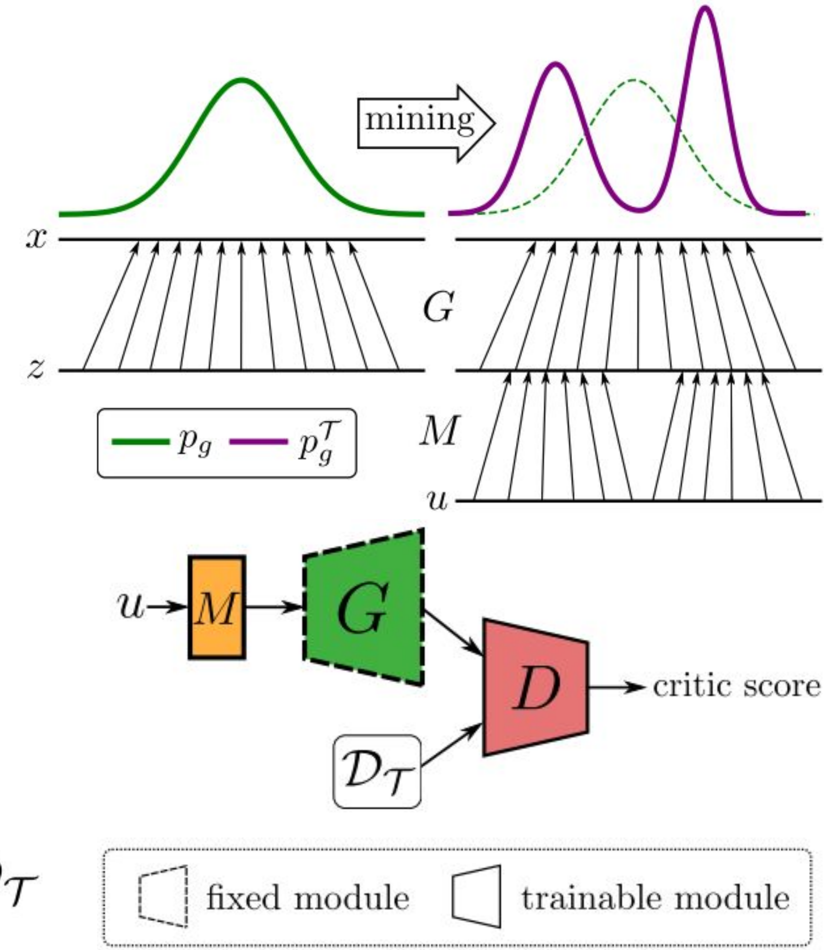
↓

Find new prior $p_z^{\mathcal{T}}(z)$ by transforming $p_z(z)$ with a miner M (MLP), training on $\mathcal{D}_{\mathcal{T}}$, G fixed

↓

Sampling **z** from promising regions of **z** with new learned multimodal distribution

Same for images from $p_g(x)$, will be closer to $\mathcal{D}_{\mathcal{T}}$



Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

MineGAN (w/o FT) - mining

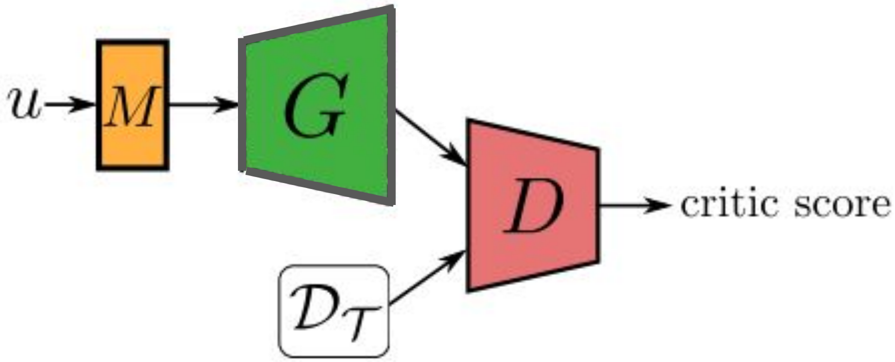
$$L_D^M = \mathbb{E}_{u \sim p_z(u)}[D(G(M(u)))] - \mathbb{E}_{x \sim p_{data}^{\tau}(x)}[D(x)]$$

$$L_G^M = -\mathbb{E}_{u \sim p_z(u)}[D(G(M(u)))]$$

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

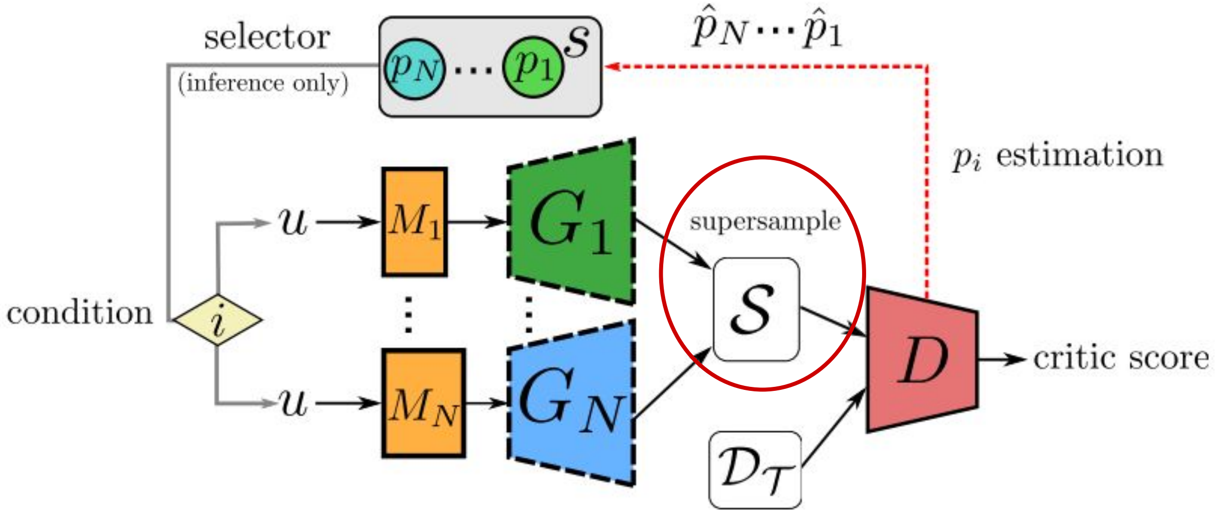
MineGAN - fine-tuning

Since mining step made output closer to $\mathcal{D}_{\mathcal{T}}$, G can be released and fine-tuned more efficiently and with more stable gradient and lower variance



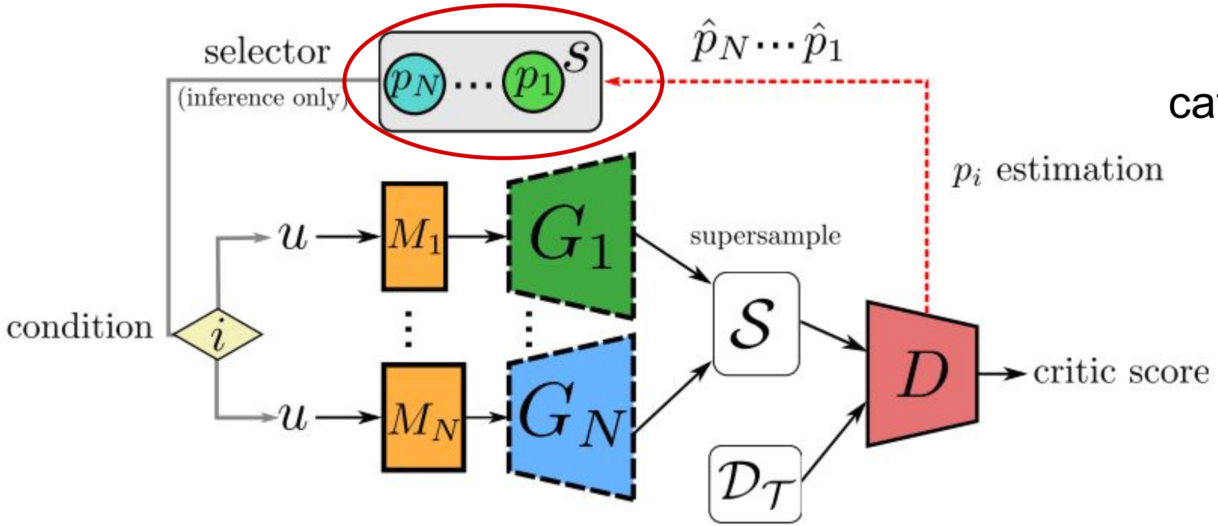
Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

MineGAN - mining multiple GANs - supersample



a set of samples, 1 per G_i : $\mathcal{S} = \{G_i(z) | z \sim p_z^i(z); i = 1, ..., N\}$

MineGAN - mining multiple GANs - selector



Training:

categorical distribution $p_1 \ldots p_N$

$\hat{p}_i$: for each S in minibatch,

acc. over K samples by i

$argmax_i D(G_i(z))$

normalize by K

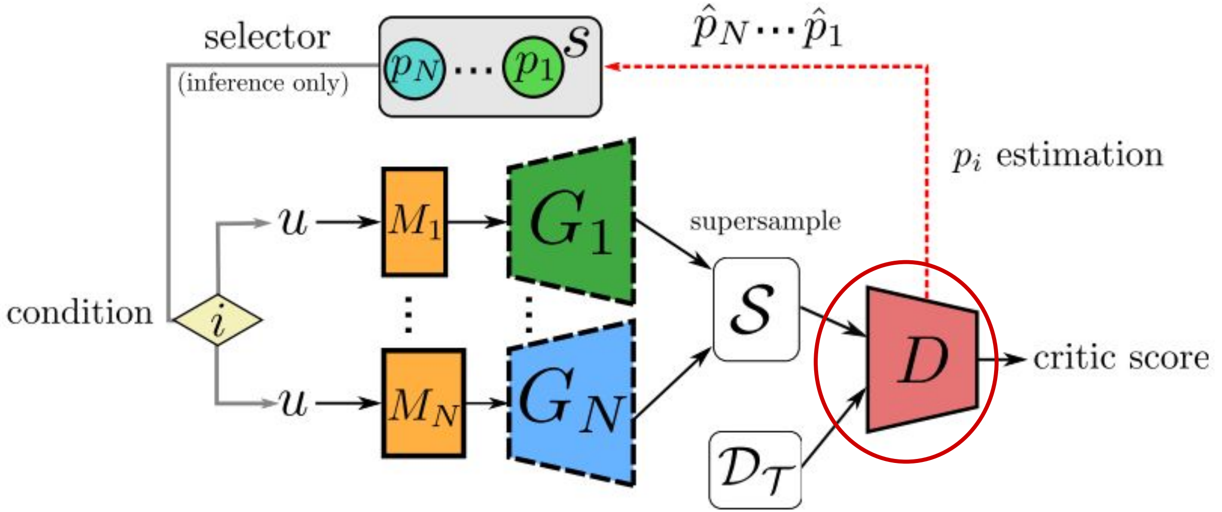
$$p_i = \frac{1}{1000} \sum_{last1000batches} \hat{p}_i$$

$$p_i > 0, \sum p_i = 1$$

Inference:
fix s, sample to get the index of the G to be used

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

MineGAN - mining multiple GANs - critic



weights from 1 random pretrained critic

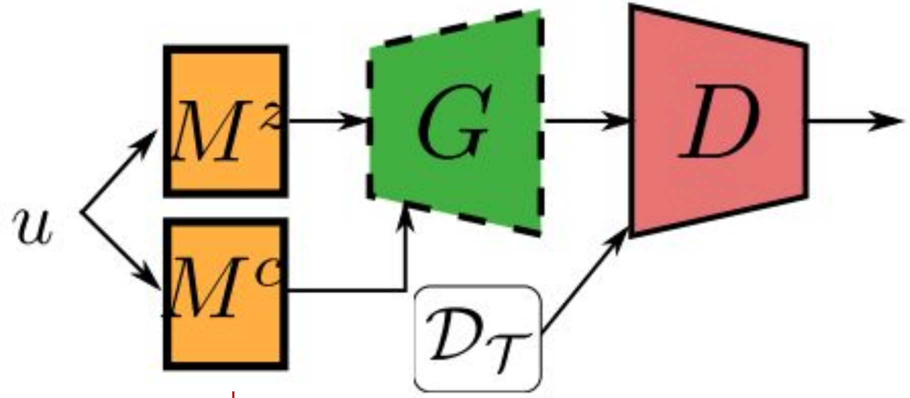
MineGAN - mining multiple GANs - loss

$$L_D^M = \mathbb{E}_{\{u^i \sim p_z^i(u)\}}[max_i \{D(G_i(M_i(u^i)))\}] - \mathbb{E}_{x \sim p_{data}^{\mathcal{T}}(x)}[D(x)]$$

$$L_G^M = -\mathbb{E}_{\{u^i \sim p_z^i(u)\}}[max_i \{D(G_i(M_i(u^i)))\}]$$

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

MineGAN - conditional GANs



During mining and FT correspondence between **c** and **z** is implicitly learned! (since both are mapped from u)

Second miner, maps **u** to a class embedding **c** (any label in target dataset)
c is projected to scale and shift parameters of each BN layer

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

MineGAN - overview

Mining - reducing the divergence between source and target distributions

- Less parameters (only M) to learn during mining - less overfitting
- Less adaptation during fine-tuning - less overfitting
- Fake images closer to target - more efficient training
- Transfer for conditional GANs does not need target labels
- Does not optimize **z,** but finds more relevant regions

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Evaluation measures - FID

sets of images embedded w/ a CNN to some layer

$$FID(\chi_1, \chi_2) = \underbrace{\|\mu_1 - \mu_2\|_2^2}_{\text{precision}} + \underbrace{Tr(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{\frac{1}{2}})}_{\text{mutual variance}}$$

- The **lower** the better
- Easy to compute
- Correlates with human perception
- Unstable on small datasets

Wang et al. (2018). Transferring GANs: generating images from limited data

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Evaluation measures - KMMD w/ Gaussian kernel

- Distance between means of CNN features of images from the 2 sets
- The **lower** the better

as in Noguchi & Harada (2019)

Noguchi & Harada. (2019). Image generation from small datasets via batch statistics adaptation

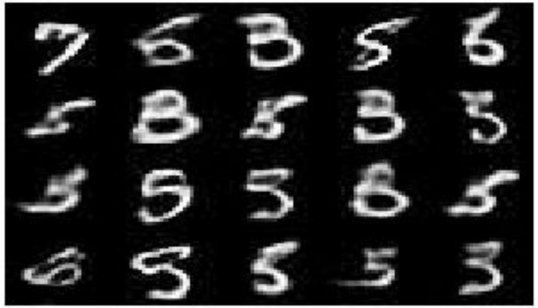
Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Evaluation measures - MV

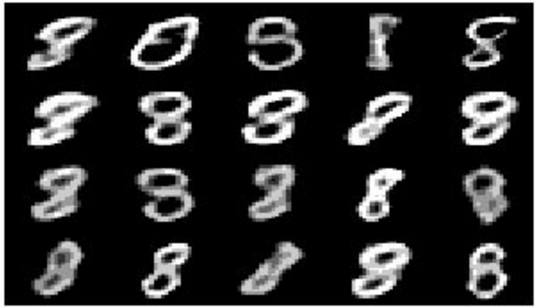
- The **higher** the better
- Indicates the variety in the generated images (since high variety is challenging for GANs)

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

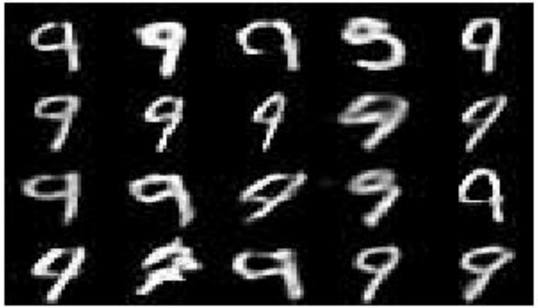
Experiments: 1. MNIST, w/o FT, off-manifold



5



8



9

G is pre-trained to synthesize all *MNIST* digits except for the target one

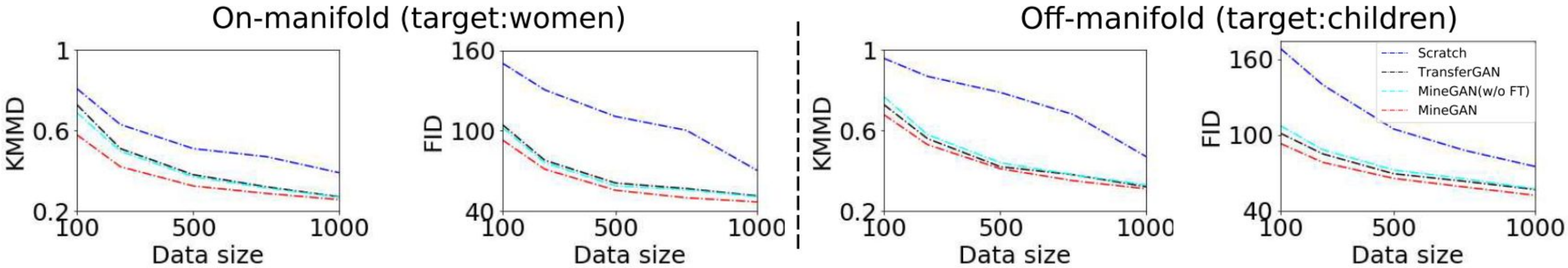
target set: 1000 images 28x28

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 2. On- and off-manifold faces, 1 GAN

1024 x 1024 images, varying target set size

Progressive GAN pre-trained on *CelebA* -> *FFHQ women*, *FFHQ children*

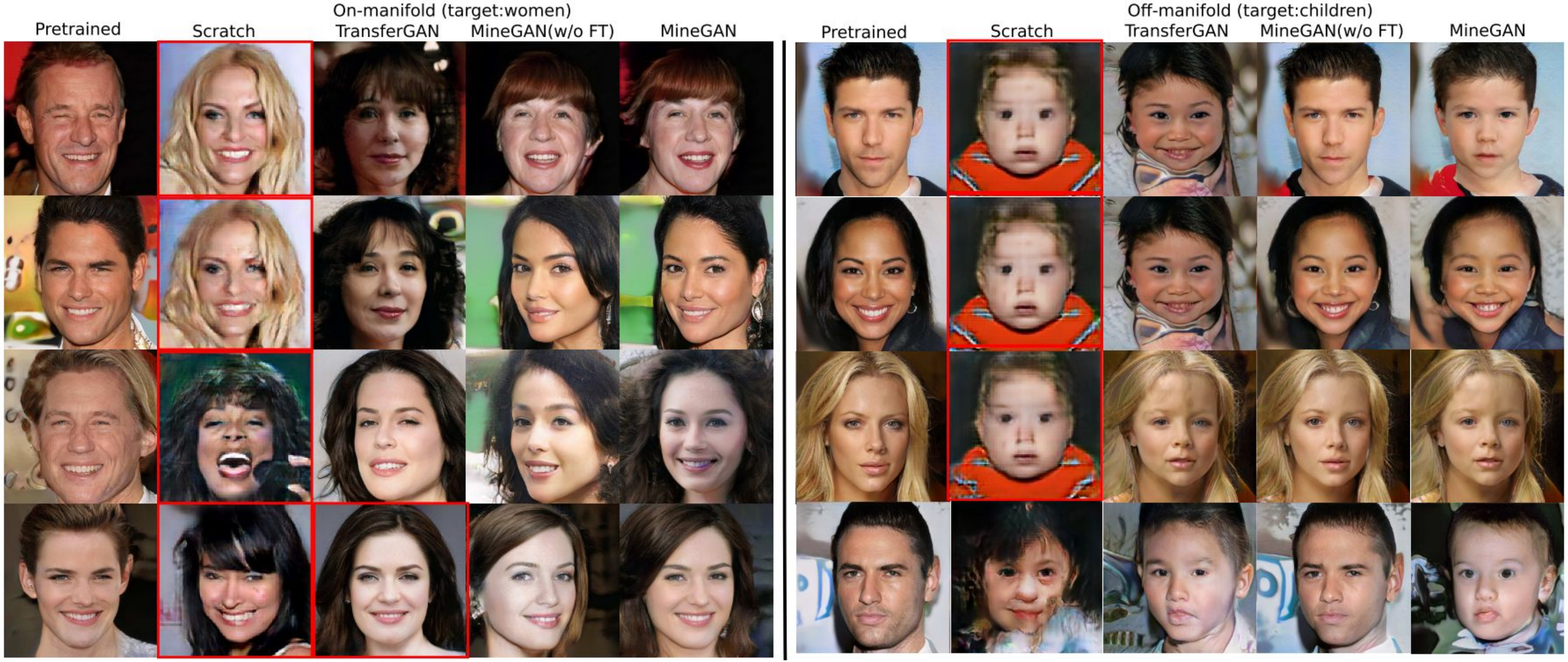


even w/o FT,
slightly **better** than TransferGAN

even w/o FT,
slightly **worse** than TransferGAN

Experiments: 2. on- and off-manifold faces, 1 GAN

Target set size - 100 images



Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 3. Far off-manifold faces, small dataset, 1 GAN

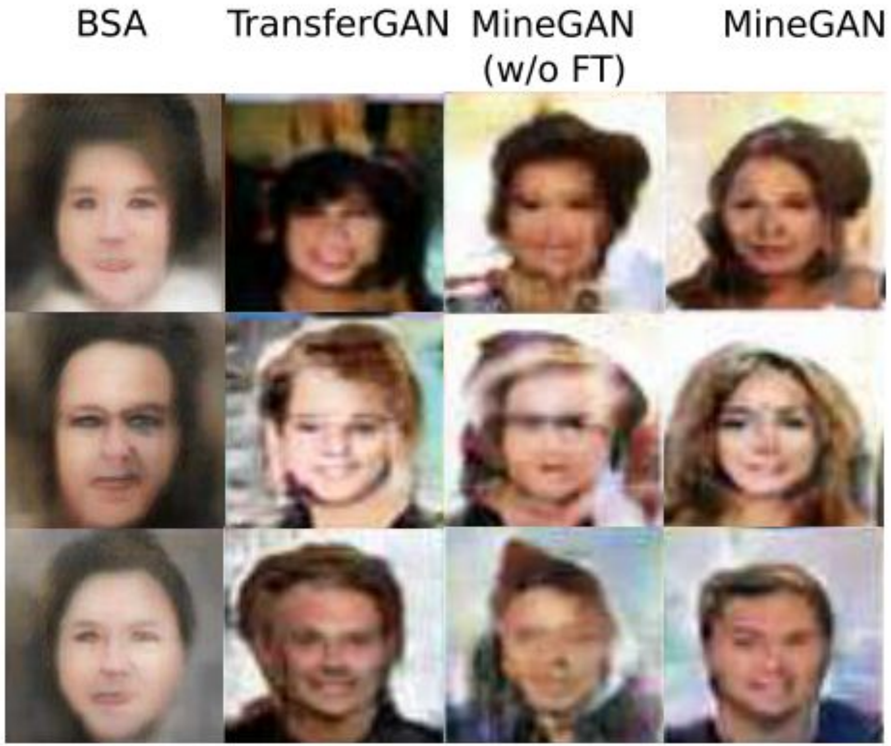
128 x 128 images, target sets sizes - 25 images

SNGAN pre-trained on *ImageNet* -> *FFHQ*, *Anime Face*

	FFHQ KMMD	FFHQ MV	Anime Face KMMD	Anime Face MV
Scratch	0.890	-	0.753	-
TransferGAN	0.346	0.506	0.347	0.785
BSA	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**

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 3. Far off-manifold faces, small dataset, 1 GAN



Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 4. 2 GANs, target is a mix of subsets from both

256 x 256 images, target set size - 200 images

2 Progressive GANs pre-trained on *LSUN*: 1 on *cars*, 1 on *buses*
-> *Red Vehicles* (red buses + red cars). 3 sets with cars/buses: 3/7, 1, 7/3

	FID
Scratch	190 / 185 / 196
TransferGAN (car)	76.9 / 72.4 / 75.6
TransferGAN (bus)	72.8 / 71.3 / 73.5
MineGAN (w/o FT)	67.3 / 65.9 / 65.8
MineGAN	**61.2** / **59.4** / **61.5**

Estimated Pi:
car: 0.34 / 0.48 / 0.64
bus: 0.66 / 0.52 / 0.36

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 4. 2 GANs, target is a mix of subsets from both

256 x 256
200 target images

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 5. 4 GANs, 2 off-manifold targets

256 x 256 images, target sets sizes - 200 images

4 Progressive GANs pre-trained on *LSUN*: *Livingroom, Kitchen, Church, Bridge*
-> 2 separate *LSUN* targets: *Bedroom, Tower*

	Tower FID	Bedroom FID
Scratch	176	181
TransferGAN (livingroom)	78.9	65.4
TransferGAN (church)	73.8	71.5
MineGAN (w/o FT)	69.2	58.9
MineGAN	**62.4**	**54.7**

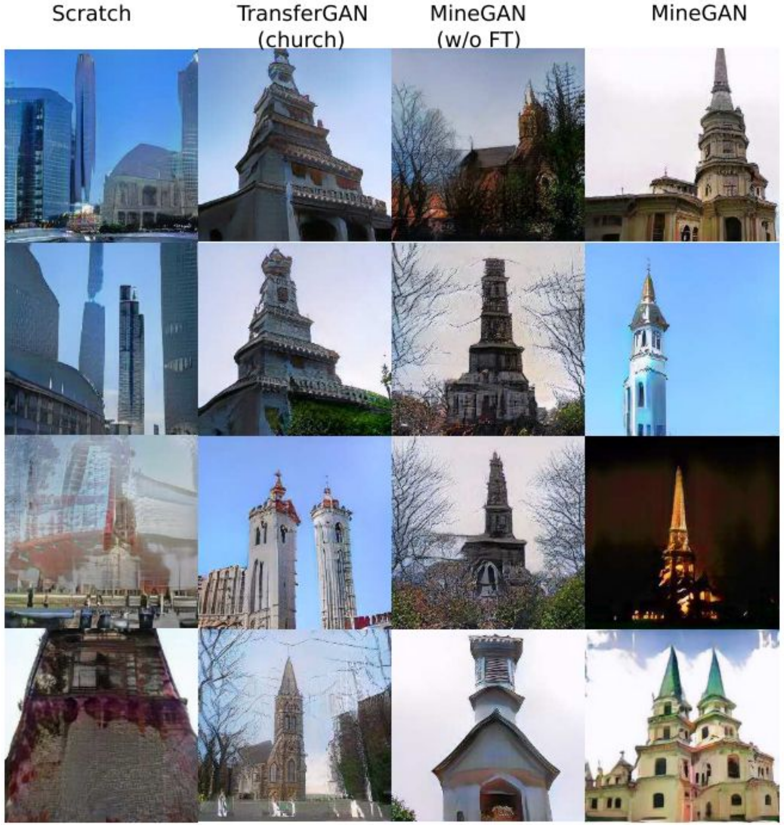
Estimated Pi

	Tower	Bedroom
livingroom	0.07	0.45
kitchen	0.06	0.40
bridge	0.42	0.08
church	0.45	0.07

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 5. 4 GANs, 2 off-manifold targets

256 x 256
target - 200 images



Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 6. Conditional GANs, off- and on-manifold

Target sets sizes - 500 images per category

BigGAN pre-trained on *ImageNet*
-> (on-manifold) *ImageNet: cock, tape player, broccoli, fire engine, harvester*
-> (off-manifold) *Places365: alley, arch, art gallery, auditorium, ballroom*

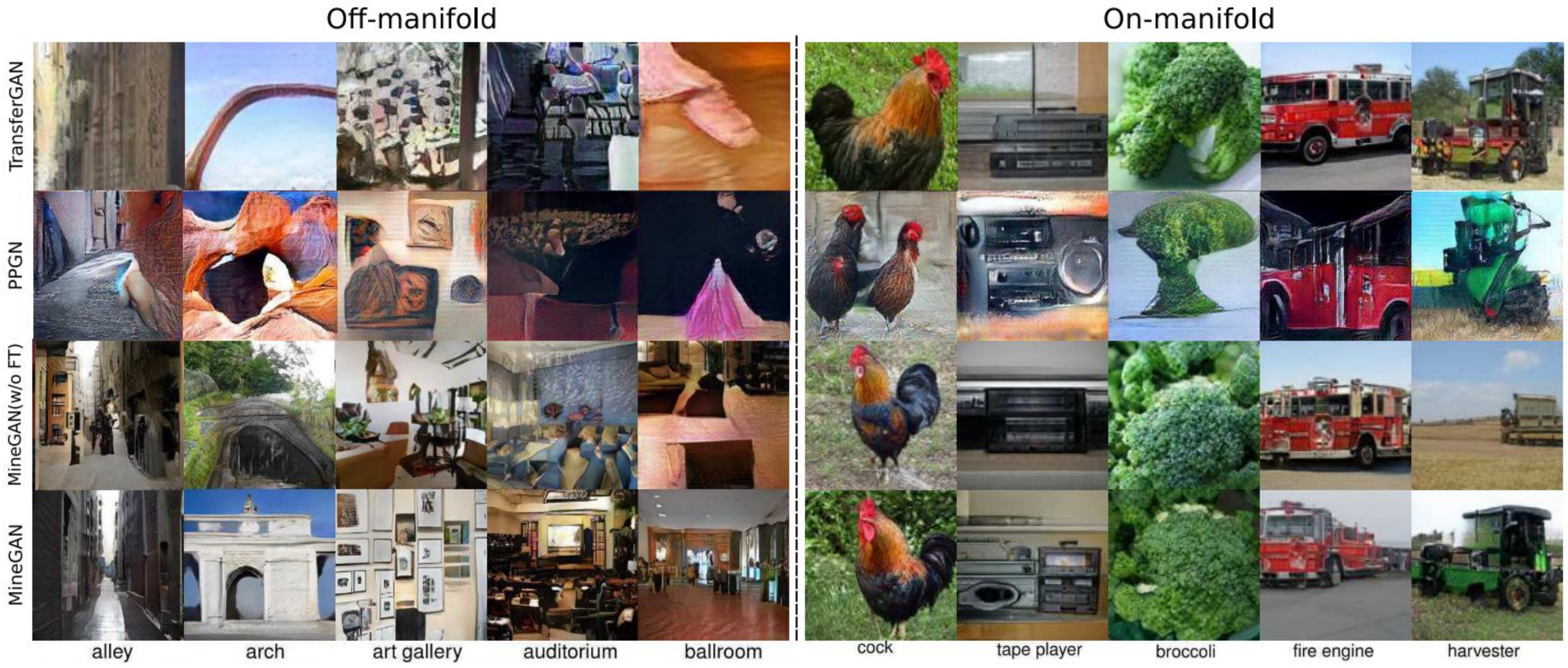
Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 6. Conditional GANs, off- and on-manifold

	Label	Off-manifold FID / KMMD	On-manifold FID / KMMD	Time
Scratch	N/N	190 / 0.96	187 / 0.93	5.1
TransferGAN	N/Y	89.2 / 0.53	58.4 / 0.39	5.1
DGN-AM	Y/Y	214 / 0.98	180 / 0.95	3020
PPGN	Y/Y	139 / 0.56	127 / 0.47	3830
MineGAN (w/o FT)	N/N	**82.3 / 0.47**	61.8 / **0.32**	5.2
MineGAN	N/N	**78.4 / 0.41**	**52.3 / 0.25**	5.2

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

Experiments: 6. Conditional GANs, off- and on-manifold



Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

MineGAN - conclusions

- New method for GAN knowledge transfer, first to use multiple GANs
- Less overfitting, more efficient, better results
- Works well with small target sets
- Efficiently uses multiple GANs, correctly estimates relevance of each
- Flexible transfer for conditional GANs
- Does not optimize **z,** but finds more relevant regions
- MineGAN (w/o FT) performs comparable w/ SOTA, preserves knowledge
- Can be applied to different GANs

Wang et al. (2020). MineGAN: effective knowledge transfer from GANs to target domains with few images

MineGAN - future research ideas

MineGAN + BSA

Add shift and scale to all channels of all conv layers.

Mine **z** -> learn how much to use which filters -> (FT all parameters).

Nested mining

If the target set is far off-manifold (and/or very small),

but there is another one between source and target -> transfer to it first.

Visualizing latent space with MineGAN

Have low-dim **u** (2-4) and map it to higher-dimensional **z**.

Then show outputs along different **u** axes.