Small Data GAN

Part 2

友達未來學院

陽明交大 AI 學院 魏澤人
tjw@nctu.edu.tw

About Me

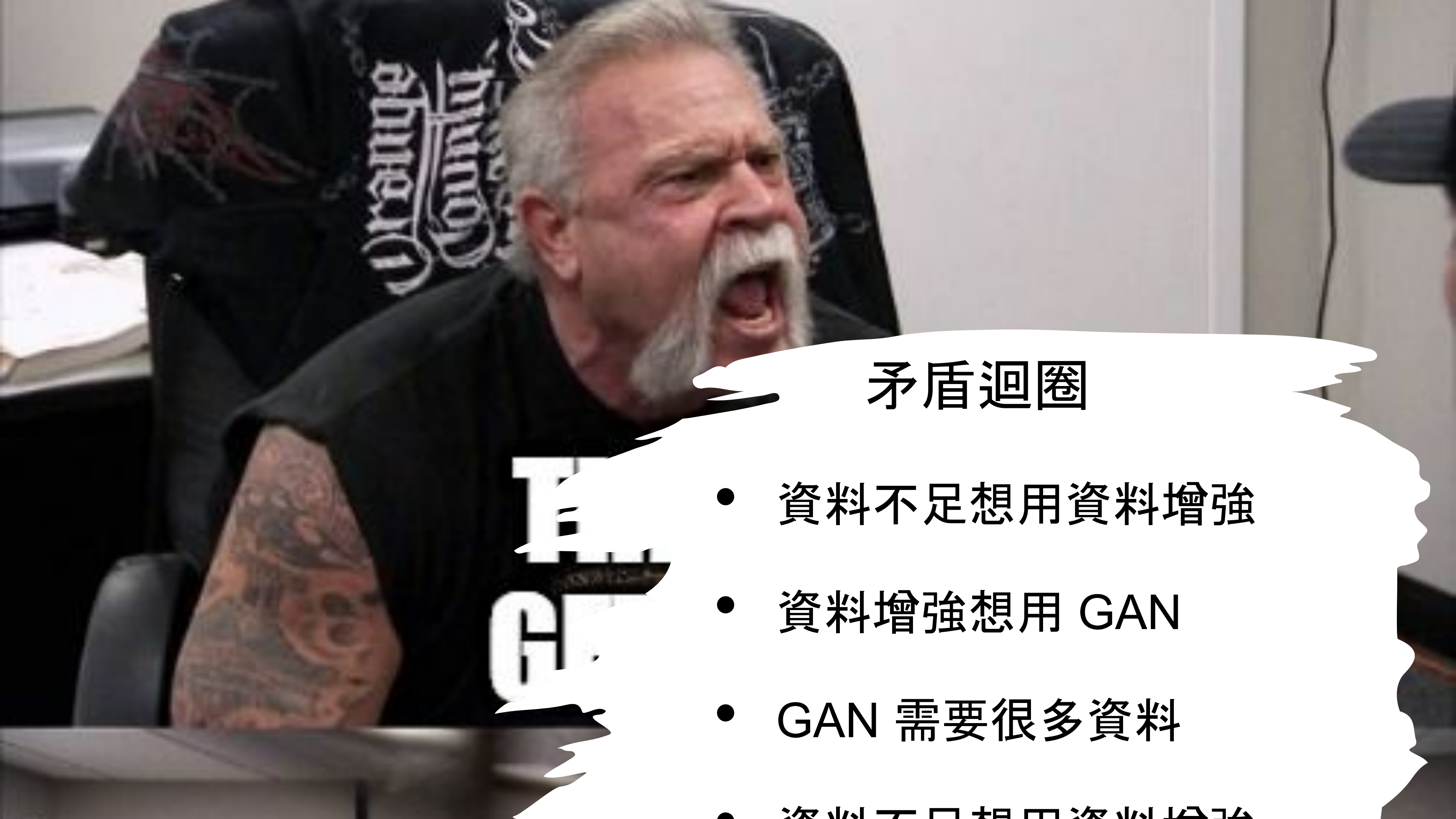
- 交大 AI 學院/緯創資通 AI 青年講座
- Google Developers Expert (Machine Learning)
- Organizer, GDG Hualien, Hualien.py, …
- Chief Mathematician, iiNumbers
- AI and Math Consultant, Ubitus, ITRI, …
- AWS Cloud Ambassador

Review(part 1)

Data Augmentation Using GAN

用 GAN 來做 Data Augmentation 如何呢？

情境: 資料不足、資料不平衡


矛盾迴圈
資料不足想用資料增強
資料增強想用 GAN
GAN 需要很多資料

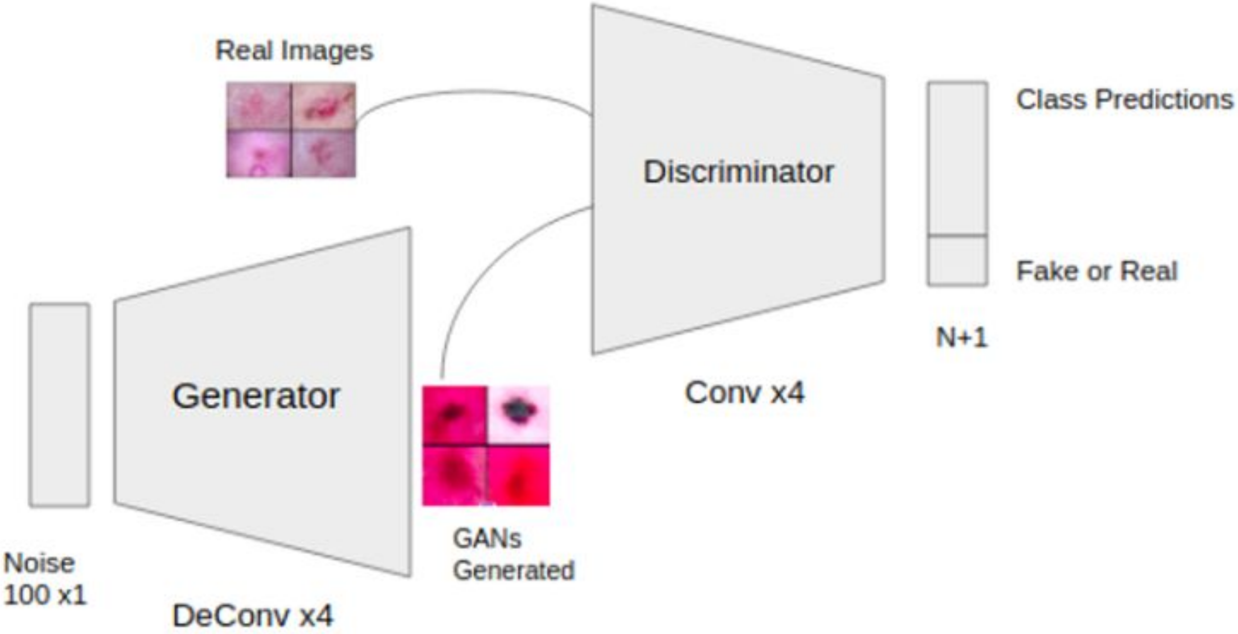
Summary

- Can use classic augmentation to generate training data for GAN.
  - But sometimes this does not work
- Combine classic augmentation with GAN synthesized samples may improve accuracy in some situations.

Summary

- GAN + classic augmentation usually improves the result a little bit
- Some times, this is exactly what you want, the last inch of the last mile.

Architecture: Semi-supervised GAN



Summary

- There are other ways of using GAN to improve the result of other tasks.
- GAN is a way of doing self-supervised learning
- Image to image translation requires fewer training data, compares to unconditional GAN.

AnoGAN

- Schlegl, T., Seeb¨ock, P., Waldstein, S. M., Schmidt-Erfurth, U., Langs, G., 2017a. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Niethammer, M., Styner,M., Aylward, S., Zhu, H., Oguz, I., Yap, P.-T., Shen, D. (Eds.), Information Processing in Medical Imaging. Springer International Publishing, Cham, pp. 146–157.
- Schlegl, Thomas, et al. "**f-AnoGAN**: Fast unsupervised anomaly detection with generative adversarial networks." Medical image analysis 54 (2019): 30-44.
- Shin, Dong-Hoon, Roy C. Park, and Kyungyong Chung. "Decision Boundary-Based Anomaly Detection Model using **Improved AnoGAN** from ECG Data." IEEE Access 8 (2020): 108664-108674.
- Zenati, H., Foo, C. S., Lecouat, B., Manek, G., and Chan-drasekhar, V. R. **Efficient GAN-Based Anomaly Detection**. abs/1802.06222, 2018. http://arxiv.org/abs/1802.06222.
- Di Mattia, Federico, et al. "A survey on gans for anomaly detection." arXiv preprint arXiv:1906.11632 (2019).
- Akcay, Samet, Amir Atapour-Abarghouei, and Toby P. Breckon. "**Ganomaly**: Semi-supervised anomaly detection via adversarial training." Asian conference on computer vision. Springer, Cham, 2018.

Limited Data

How about using Augmentation to train GAN?

- 當資料量少的時候,
- 最直接的做法就是 Data augmentation
- 訓練 GAN 的時候,
- 如果資料量不夠, 是不是也可以用 Data augmentation?

- GAN 是用來學 distribution 的。
- Augmentation 會改變 data distribution 的特性。
- 鏡射這些，可能相對安全，
- 但旋轉、平移、變色、扭曲，有可能會改變 distribution.

This is not a No

Improved Consistency Regularization for GANs

Zhengli Zhao[1,2] **Sameer Singh**[1] **Honglak Lee**[2] **Zizhao Zhang**[2] **Augustus Odena**[2] **Han Zhang**[2]

[1] University of California, Irvine

[2] Google Research

zhengliz@uci.edu, zhanghan@google.com

Consistency Regularization

- Consistency regularization techniques are popular in the semi-supervised learning literature.
- The basic idea is simple: an input image is perturbed in some semantics-preserving ways and the sensitivity of the classifier to that perturbation is penalized.

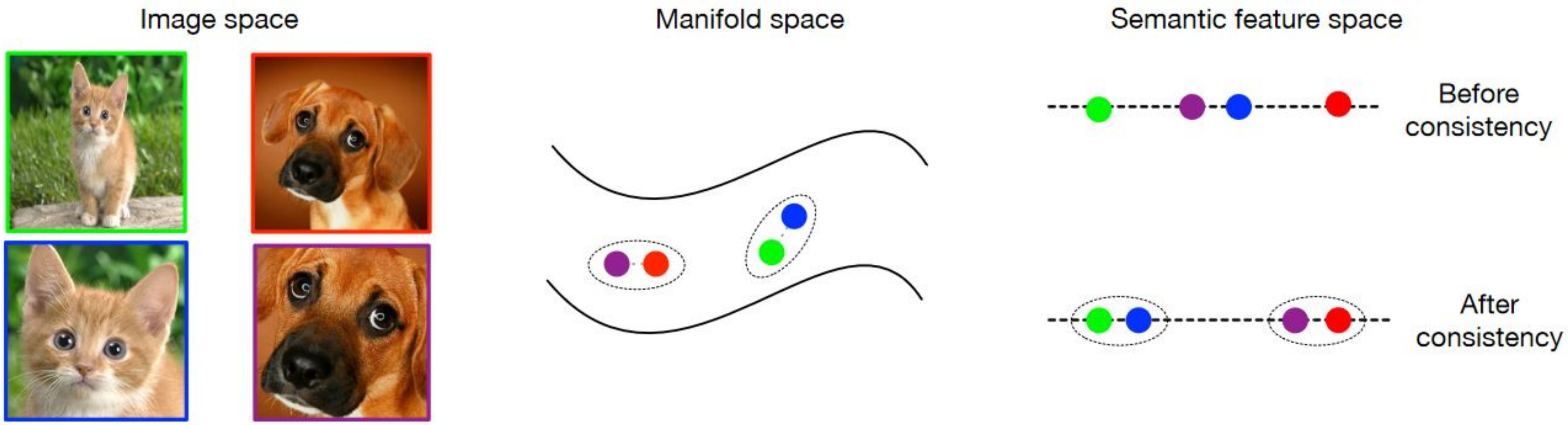
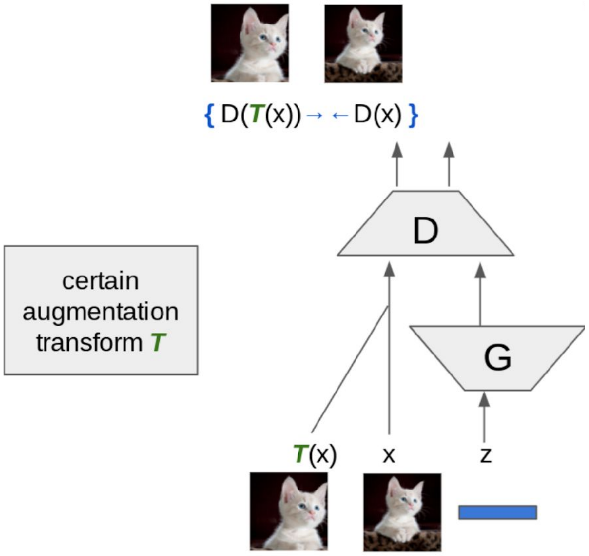


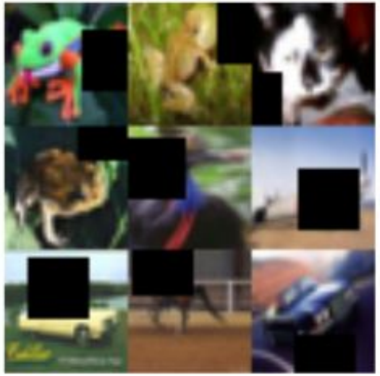
Figure 1: An illustration of consistency regularization for GANs. Before consistency regularization, the zoomed-in dog and the zoomed-in cat (bottom left) can be closer than they are to their original images in feature space induced by the GAN discriminator. This is illustrated in the upper right (the semantic feature space), where the purple dot is closer to the blue dot than to the red dot, and so forth. After we enforce consistency regularization based on the implicit assumption that image augmentation preserves the semantics we care about, the purple dot pulled closer to the red dot.



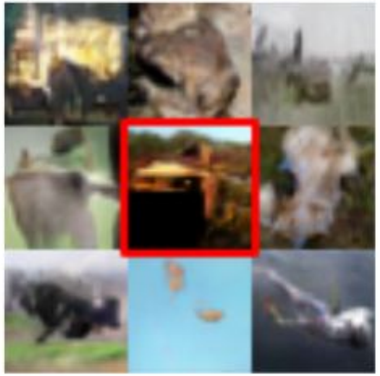
(1) CR-GAN

$$\min_D L_{cr} = \min_D \sum_{j=m}^{n} \lambda_j \left\| D_j(x) - D_j(T(x)) \right\|^2,$$

加上這項 regularization 就是 CR-GAN.



(d) 16 × 16 cutout.



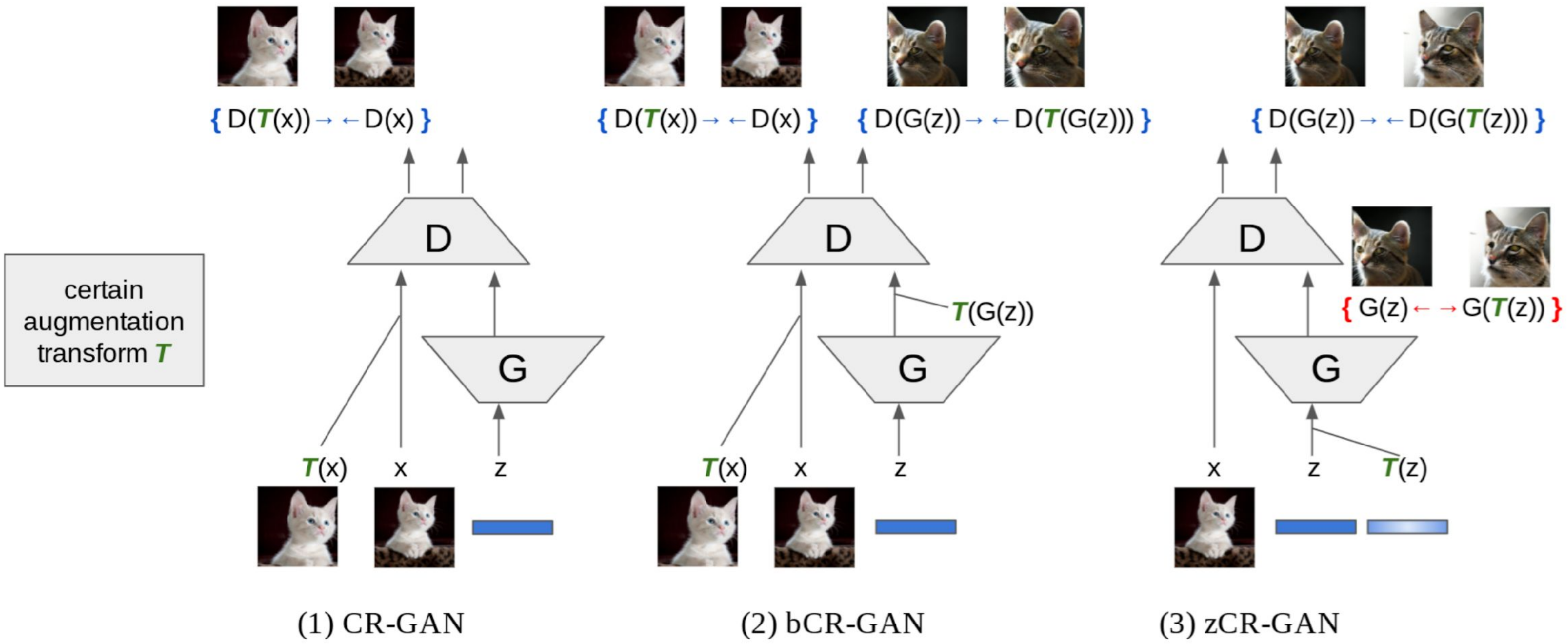
(e) CR samples.

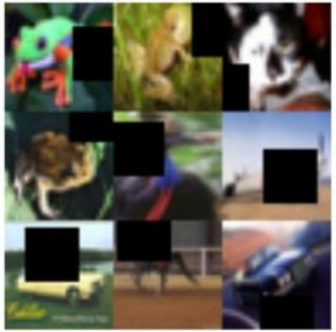
(g) 32 × 32 cutout.

(h) CR samples.

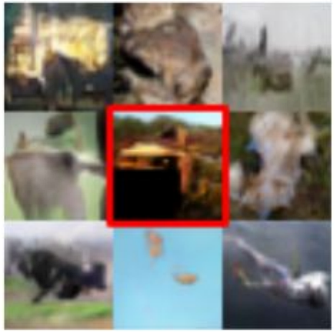
- 但 augmentation 會影響輸出。
- 比方做了 cutout augmentation
- 生出來的圖有時也會產生 cutout。

- CR 是 Consistency Regularization
- bCR 是 balanced 的版本
- zCR 是加上 latent space CR 的版本。


certain augmentation transform T
{ D(T(x)) → ← D(x) }
D
G
T(x)
x
z
{ D(T(x)) → ← D(x) }
{ D(G(z)) → ← D(T(G(z))) }
T(G(z))
{ D(G(z)) → ← D(G(T(z))) }
{ G(z) ← → G(T(z)) }
T(z)
(1) CR-GAN
(2) bCR-GAN
(3) zCR-GAN



(d) 16×16 cutout.



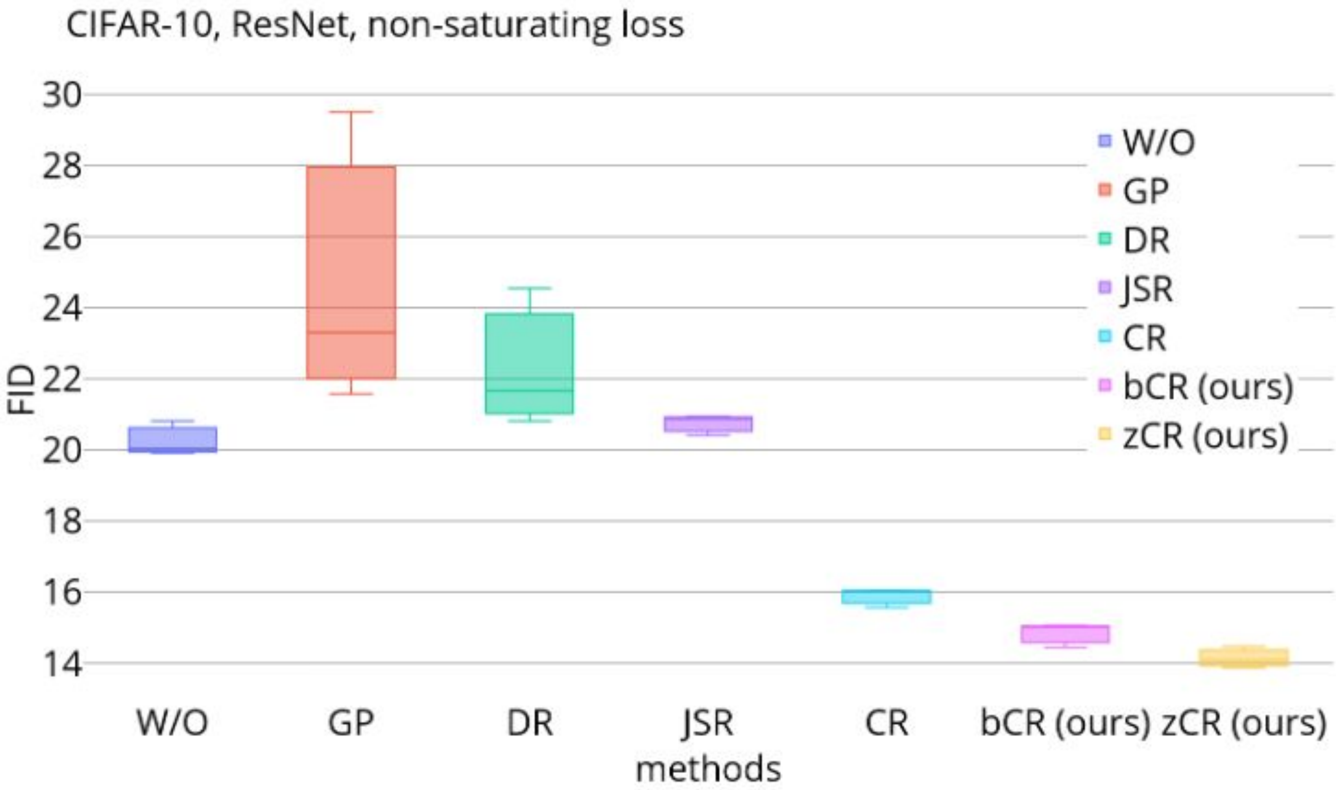
(e) CR samples.

(f) bCR samples.

(g) 32×32 cutout.

(h) CR samples.

(i) bCR samples.


CIFAR-10, ResNet, non-saturating loss
FID
30
28
26
24
22
20
18
16
14
W/O
GP
DR
JSR
CR
bCR (ours)
zCR (ours)
methods

Models	CIFAR-10	ImageNet
SNGAN	17.50	27.62
BigGAN	14.73	8.73
CR-BigGAN	11.48	6.66
bCR-BigGAN	10.54	6.24
zCR-BigGAN	10.19	5.87
ICR-BigGAN	**9.21**	**5.38**

Training Generative Adversarial Networks with Limited Data

Tero Karras
NVIDIA

Miika Aittala
NVIDIA

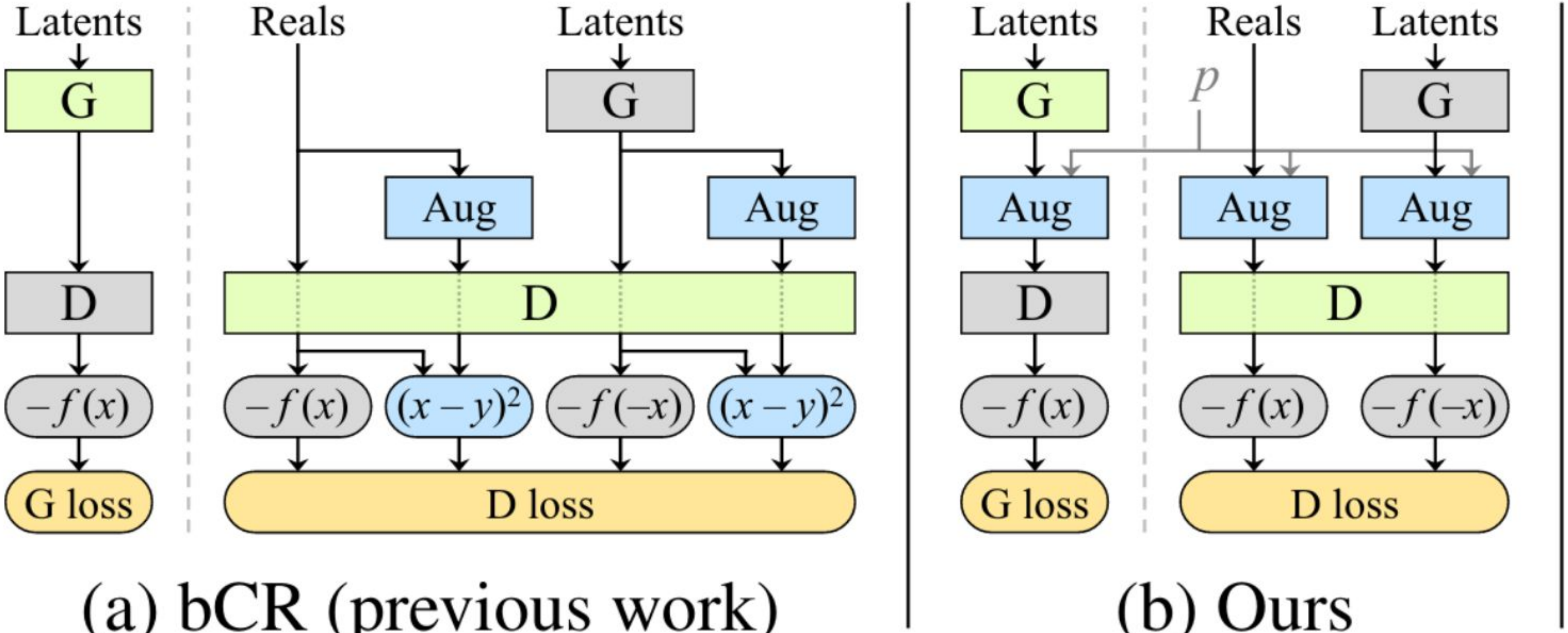
Janne Hellsten
NVIDIA

Samuli Laine
NVIDIA

Jaakko Lehtinen
NVIDIA and Aalto University

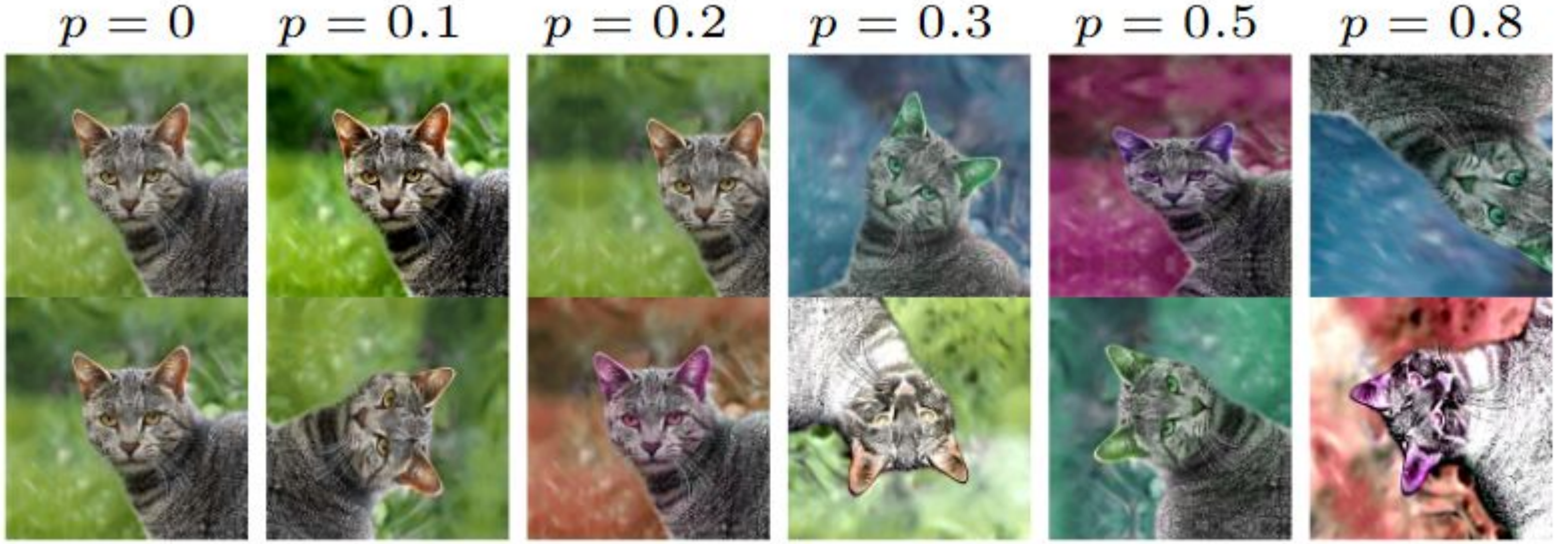
Timo Aila
NVIDIA

AKA StyleGAN2-ADA



(a) bCR (previous work)

(b) Ours



(c) Effect of augmentation probability p

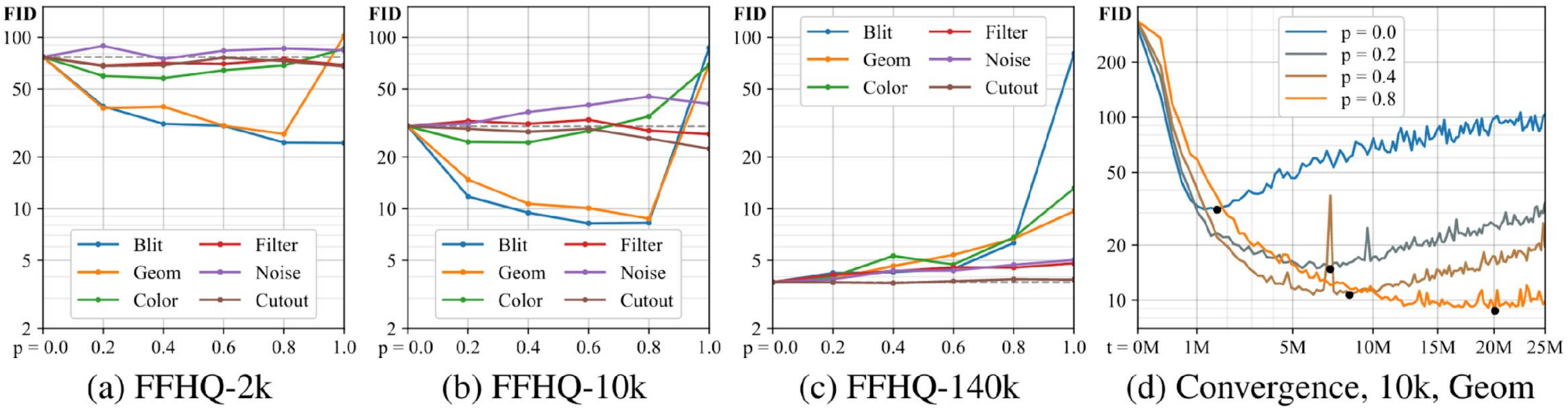


Figure 4: (a-c) Impact of p for different augmentation categories and dataset sizes. The dashed gray line indicates baseline FID without augmentations. (d) Convergence curves for selected values of p using geometric augmentations with 10k training images.

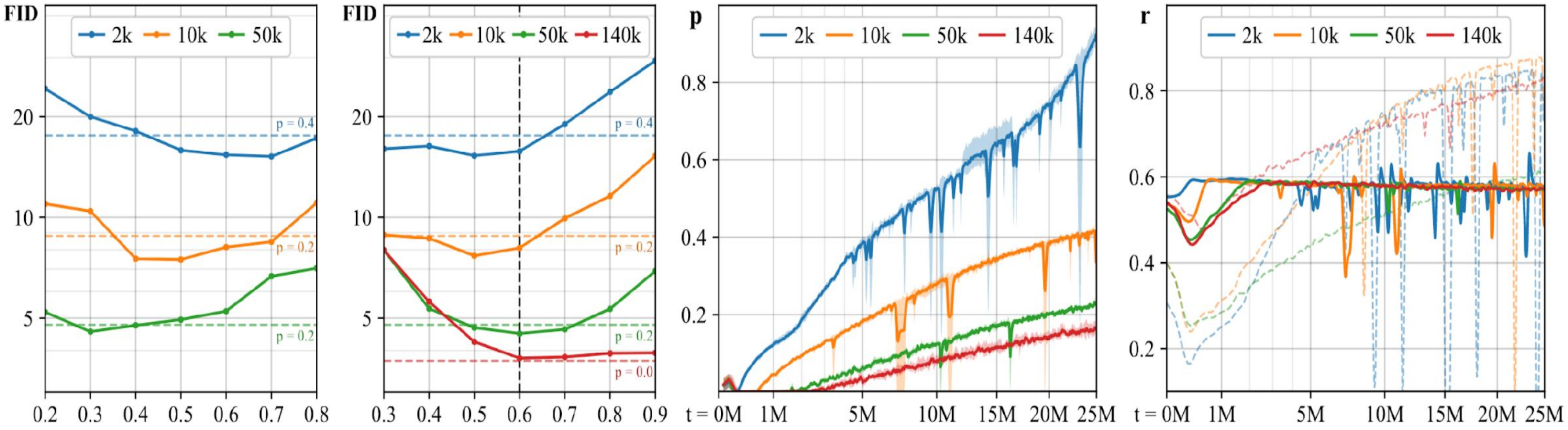
Adaptive discriminator augmentation

Used for preventing overfitting

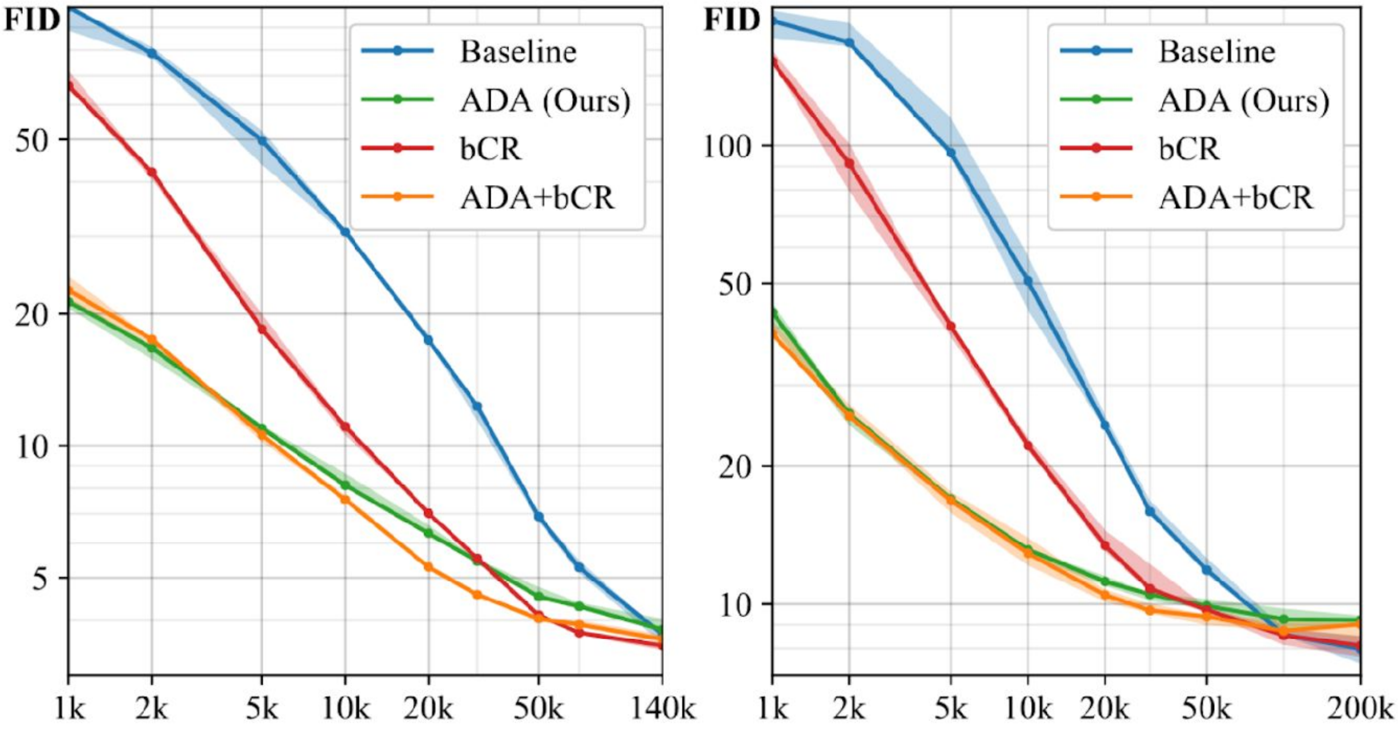
$$r_v = \frac{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{generated}}]}$$

$$r_t = \mathbb{E}[\text{sign}(D_{\text{train}})]$$

Overfitting Heuristics



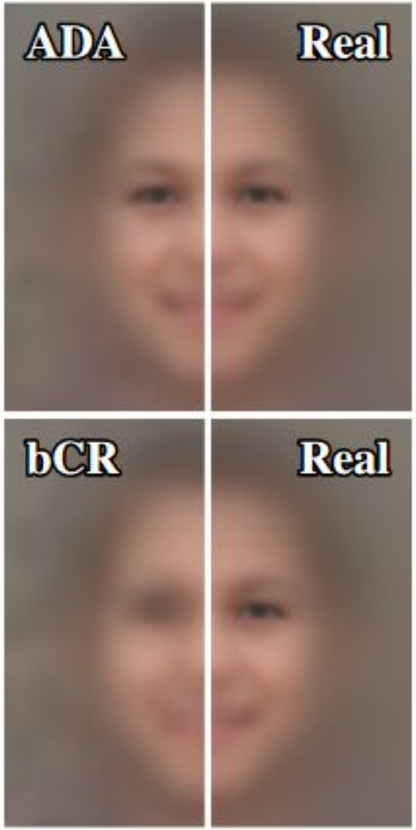
(a) r_v target sweep (b) r_t target sweep (c) Evolution of p over training (d) Evolution of r_t



(a) FFHQ (256×256) (b) LSUN CAT (256×256)

Dataset		Baseline	ADA	+ bCR
FFHQ	1k	100.16	**21.29**	22.61
	5k	49.68	10.96	**10.58**
	10k	30.74	8.13	**7.53**
	30k	12.31	5.46	**4.57**
	70k	5.28	4.30	**3.91**
	140k	3.71	3.81	**3.62**
LSUN CAT	1k	186.91	43.25	**38.82**
	5k	96.44	16.95	**16.80**
	10k	50.66	13.13	**12.90**
	30k	15.90	10.50	**9.68**
	100k	**8.56**	9.26	8.73
	200k	**7.98**	9.22	9.03

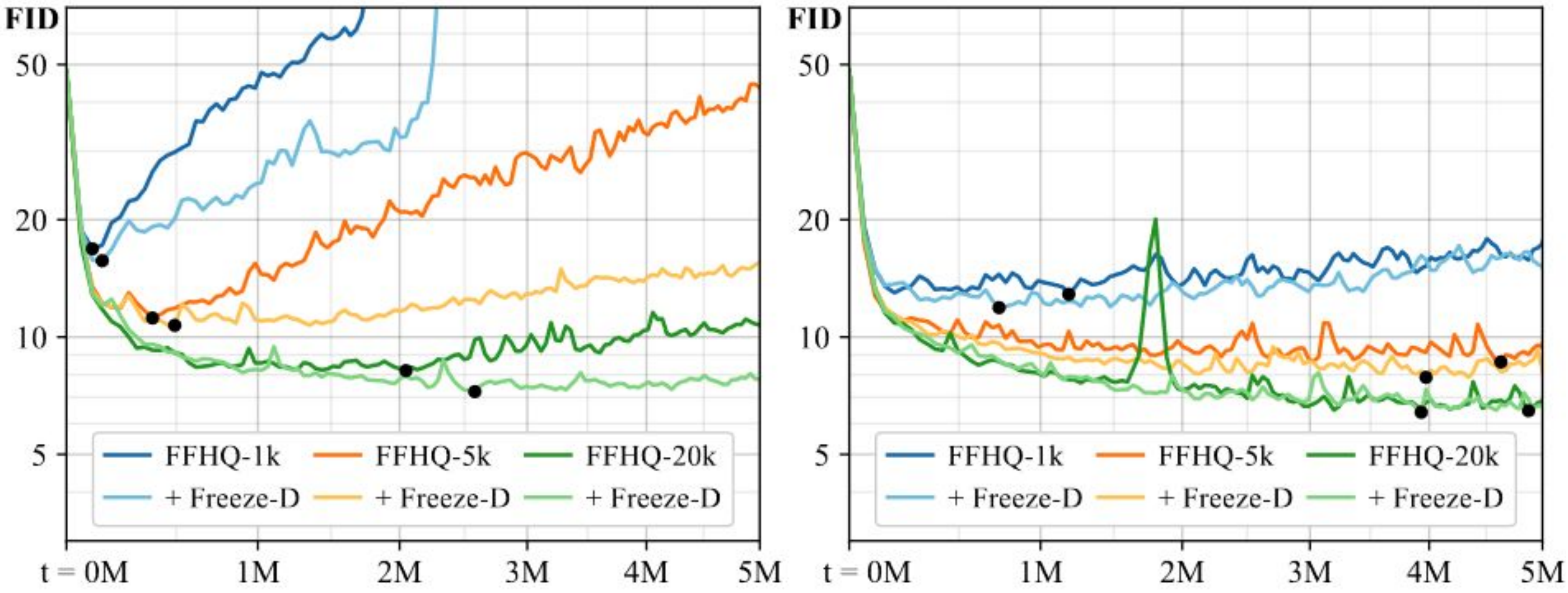
(c) Median FID



(d) Mean image

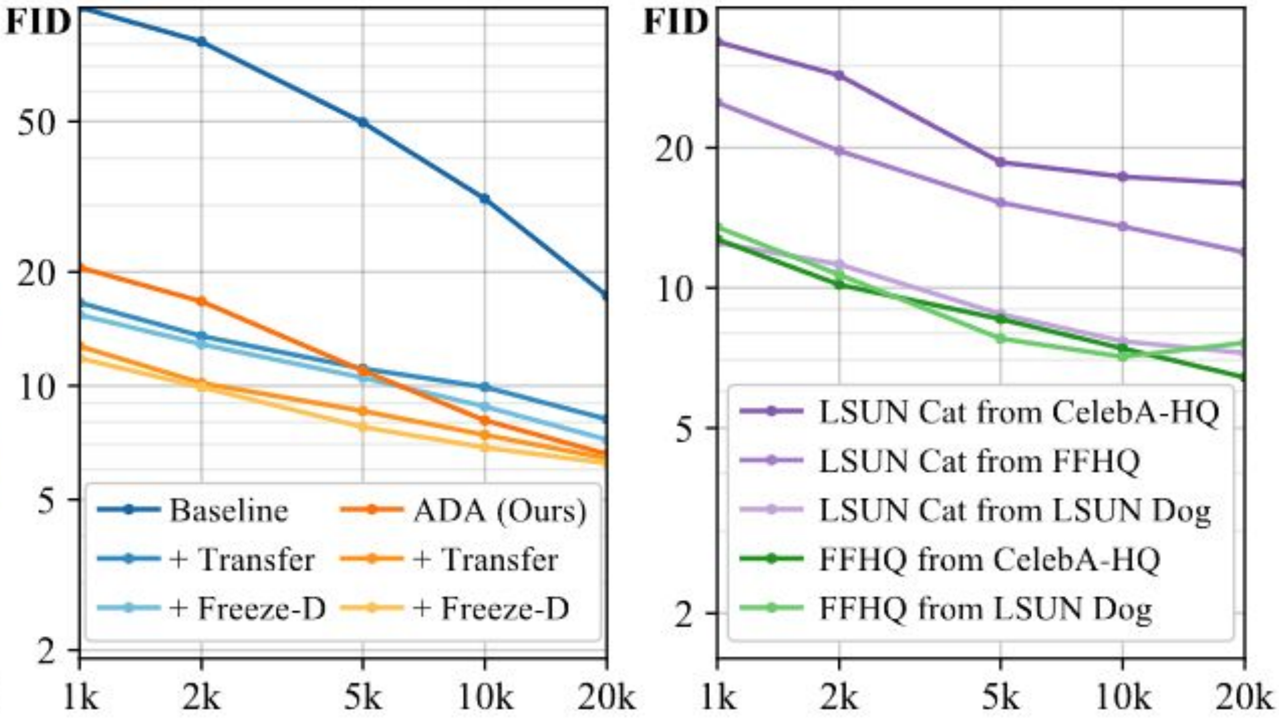
FFHQ (256×256)	**2k**	**10k**	**140k**
Baseline	78.80 ±2.31	30.73 ±0.48	3.66 ±0.10
PA-GAN [48]	56.49 ±7.28	27.71 ±2.77	3.78 ±0.06
WGAN-GP [15]	79.19 ±6.30	35.68 ±1.27	6.54 ±0.37
zCR [53]	71.61 ±9.64	23.02 ±2.09	**3.45** ±0.19
Auxiliary rotation [6]	66.64 ±3.64	25.37 ±1.45	4.16 ±0.05
Spectral norm [31]	88.71 ±3.18	38.58 ±3.14	4.60 ±0.19
Shallow mapping	71.35 ±7.20	27.71 ±1.96	3.59 ±0.22
Adaptive dropout	67.23 ±4.76	23.33 ±0.98	4.16 ±0.05
ADA (Ours)	**16.49** ±0.65	**8.29** ±0.31	3.88 ±0.13

(a) Comparison methods



(a) Without ADA

(b) With ADA



(c) Dataset sizes (d) Datasets

ADA (Ours), untruncated

FID **15.34** – KID **0.81**$\times 10^3$ – Recall 0.261

Original StyleGAN2 config F, untruncated

FID 19.47 – KID 3.16$\times 10^3$ – Recall **0.350**

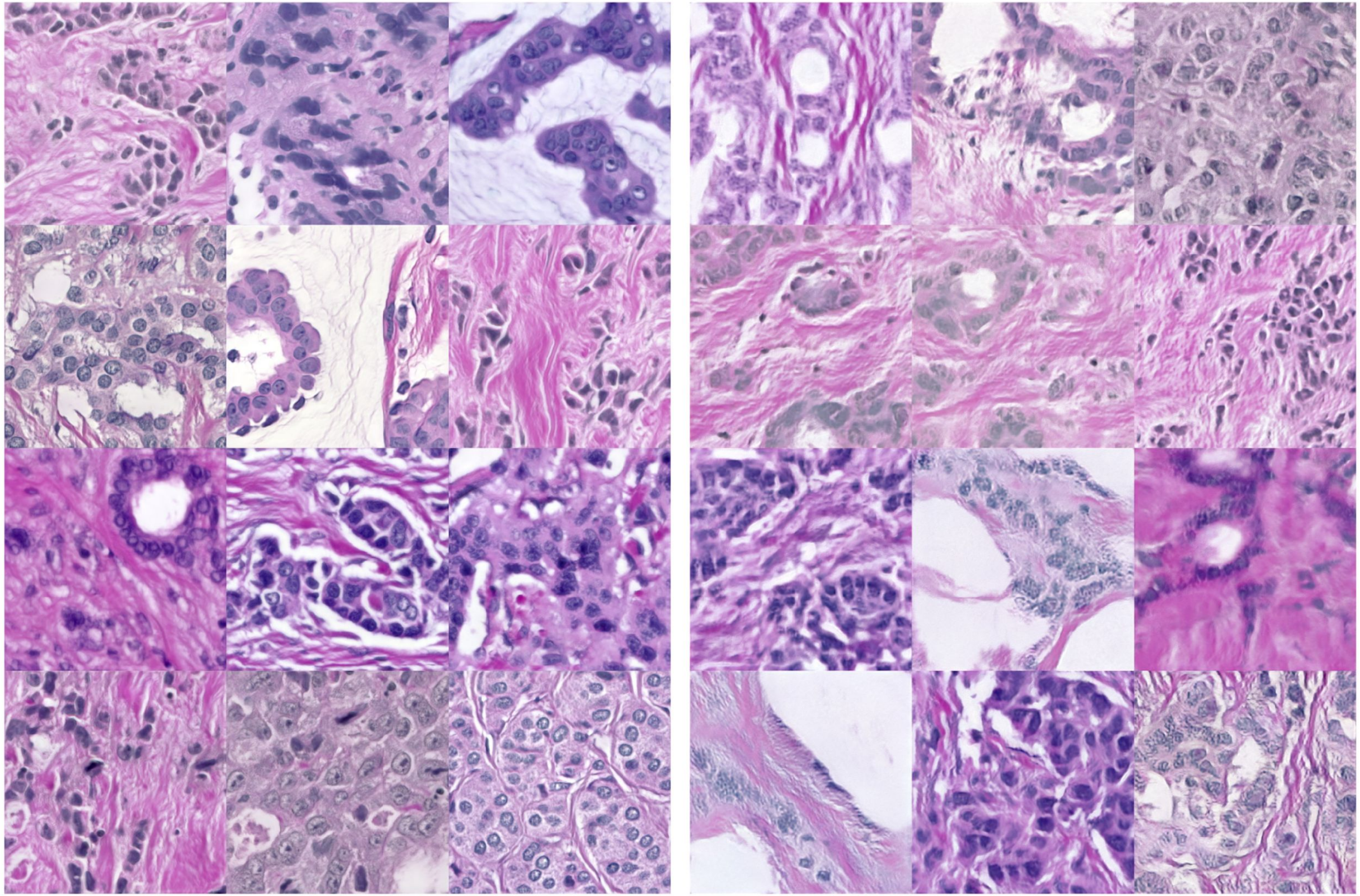
Figure 12: Uncurated 1024×1024 results generated for METFACES (1336 images) with and without ADA, along with real images from the training set. Both generators were trained using transfer learning, starting from the pre-trained StyleGAN2 for FFHQ. We recommend zooming in.

ADA (Ours), untruncated — Original StyleGAN2 config F, untruncated

FID **3.55** – KID **0.66**$\times 10^3$ – Recall **0.430** — FID 5.13 – KID 1.54$\times 10^3$ – Recall 0.215

Figure 14: Uncurated 512×512 results generated for AFHQ CAT [7] (5153 images) with and without

ADA (Ours), untruncated | Original StyleGAN2 config F, untruncated



FID **15.71** – KID **2.88**$\times10^3$ – Recall **0.340** | FID 97.72 – KID 89.76$\times10^3$ – Recall 0.027

Figure 13: Uncurated 512×512 results generated for BRECAHAD [1] (1944 images) with and

Summary

- 用 Augmentation 來訓練 GAN 可能會造成洩漏
- 可以控制 Discriminator 看到**平衡**的 distribution 來避免洩漏

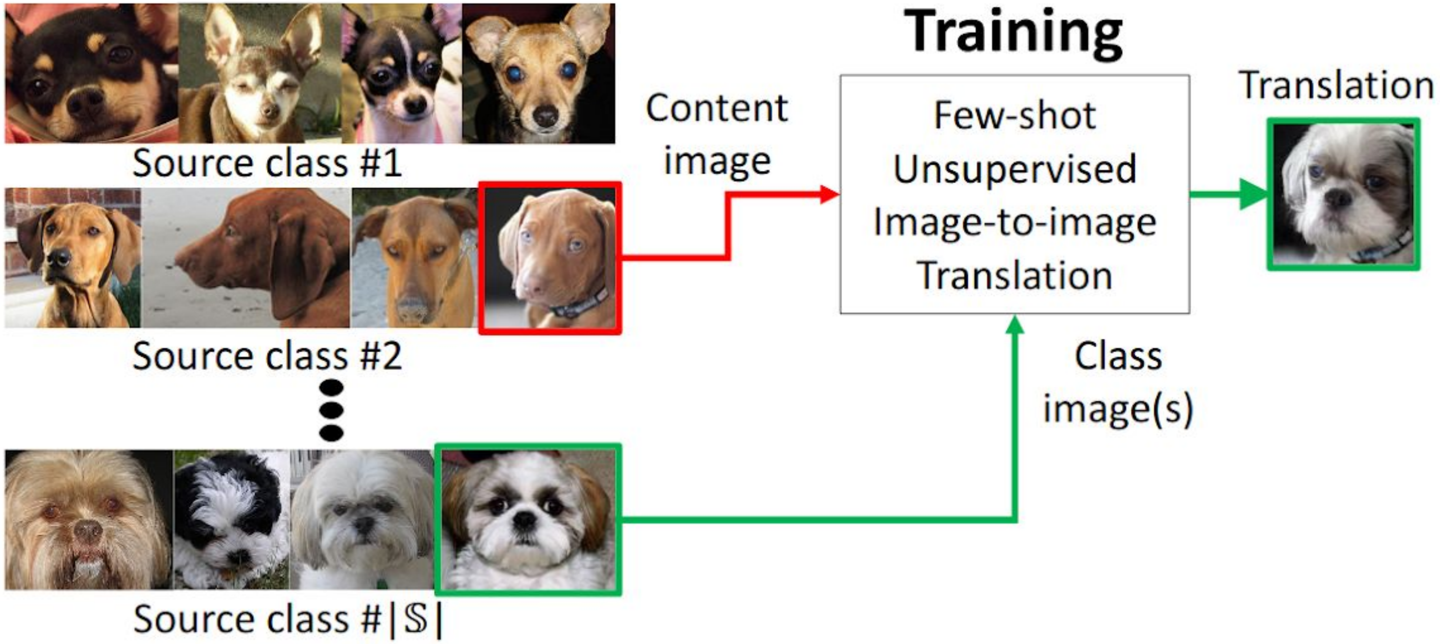
Few-shot GAN

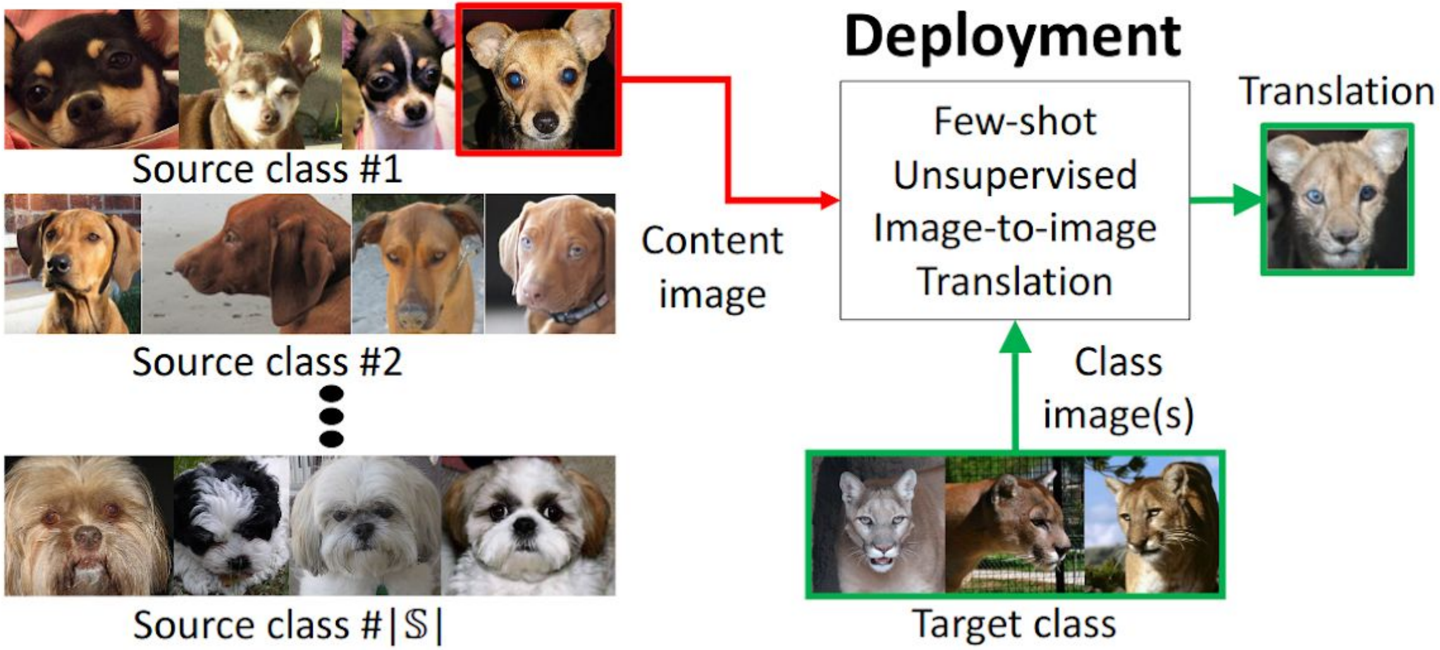
- http://nvidia-research-mingyuliu.com/ganimal
- https://nvlabs.github.io/FUNIT/
- Talking head https://arxiv.org/abs/1905.08233
- https://nvlabs.github.io/COCO-FUNIT/paper.pdf

1. Input
2. Cocker Spaniel
3. Appenzeller
4. Rottweiler
5. Komondor
6. Kerry Blue Terrier
7. Lakeland Terrier
8. Greater Swiss Mountain Dog
9. Lynx
10. Pug
11. Grey fox
12. Affenpinscher
13. Egyptian Cat
14. Cougar
15. Toy Poodle
16. Red Fox

Few-Shot Unsupervised Image-to-Image Translation

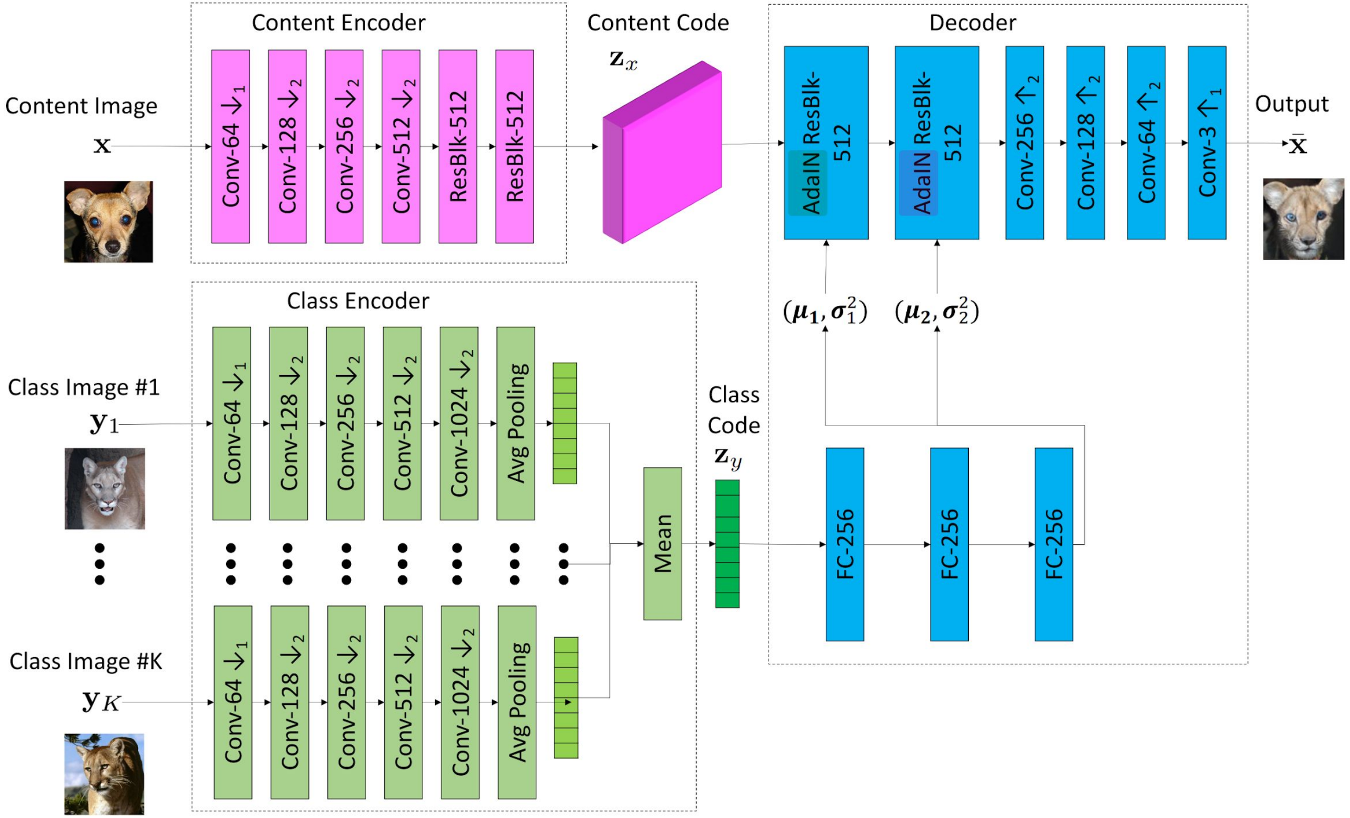
Ming-Yu Liu[1], Xun Huang[1,2], Arun Mallya[1], Tero Karras[1], Timo Aila[1], Jaakko Lehtinen[1,3], Jan Kautz[1]
[1]NVIDIA, [2]Cornell University, [3]Aalto University
{mingyul, xunh, amallya, tkarras, taila, jlehtinen, jkautz}@nvidia.com


Training
Source class #1
Content
image
Few-shot
Unsupervised
Image-to-image
Translation
Translation
Source class #2
Class
image(s)
Source class #|𝕊|


Source class #1
Source class #2
Source class #|𝕊|
Content image
Deployment
Few-shot Unsupervised Image-to-image Translation
Translation
Class image(s)
Target class


Input $\boldsymbol{x}$
Class image $\boldsymbol{y}_1$
Class image $\boldsymbol{y}_2$
StarGAN -Unfair-5
StarGAN -Fair-5
FUNIT-5 $\bar{\boldsymbol{x}}$


$\mathbf{y}_1$
$\mathbf{y}_2$
$\mathbf{x}$
$\bar{\mathbf{x}}$


Content Encoder
Content Code
Decoder
Content Image
x
Conv-64 ↓1
Conv-128 ↓2
Conv-256 ↓2
Conv-512 ↓2
ResBlk-512
ResBlk-512
z_x
AdaIN ResBlk-512
AdaIN ResBlk-512
Conv-256 ↑2
Conv-128 ↑2
Conv-64 ↑2
Conv-3 ↑1
Output
x̄
Class Encoder
(μ1, σ1²)
(μ2, σ2²)
Class Image #1
y1
Conv-1024 ↓2
Avg Pooling
Class
Code
z_y
Mean
FC-256
FC-256
FC-256
Class Image #K
yK

Loss of FUNIT

$$\min_D \max_G \mathcal{L}_{\mathrm{GAN}}(D, G) + \lambda_{\mathrm{R}}\mathcal{L}_{\mathrm{R}}(G) + \lambda_{\mathrm{F}}\mathcal{L}_{\mathrm{FM}}(G)$$

$$\begin{aligned}\mathcal{L}_{\mathrm{GAN}}(G, D) =& E_{\mathbf{x}}\left[-\log D^{c_x}(\mathbf{x})\right] + \\ & E_{\mathbf{x},\{\mathbf{y}_1,\ldots,\mathbf{y}_K\}}\left[\log\left(1 - D^{c_y}(\bar{\mathbf{x}})\right)\right]\end{aligned}$$

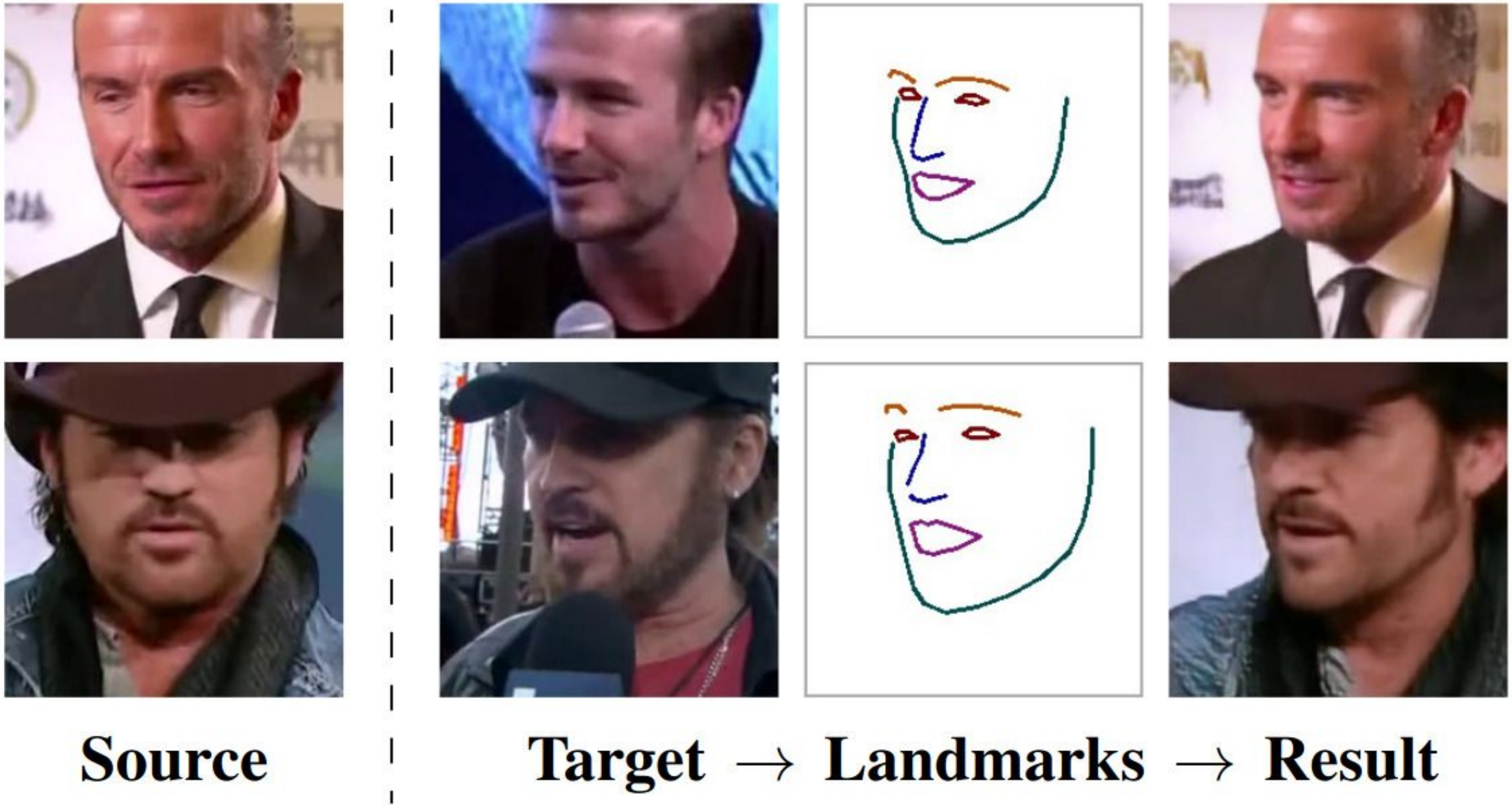
$$\mathcal{L}_{\mathrm{R}}(G) = E_{\mathbf{x}}\left[||\mathbf{x} - G(\mathbf{x}, \{\mathbf{x}\})||_1^1\right]$$

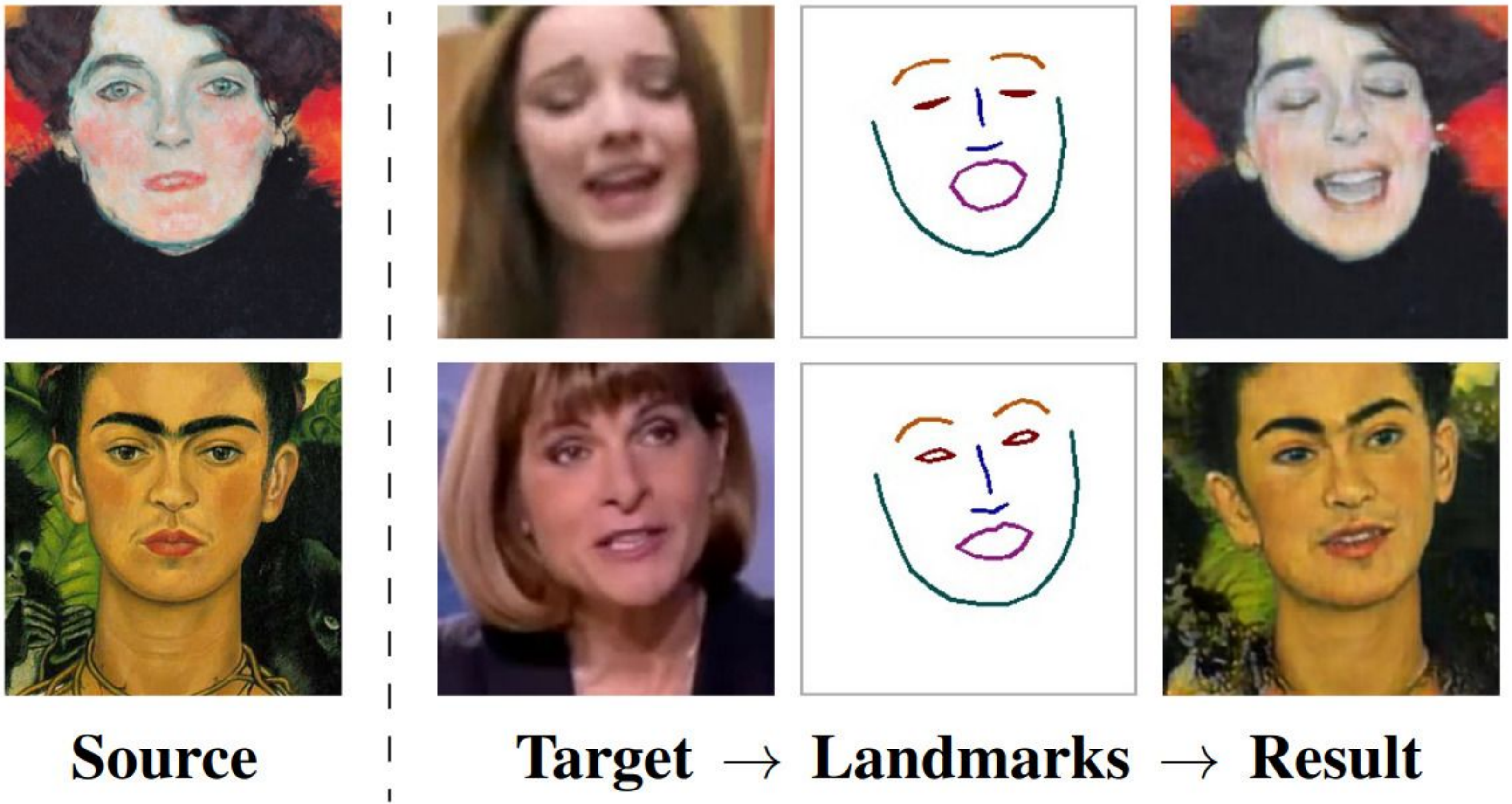
$$\mathcal{L}_{\mathrm{F}}(G) = E_{\mathbf{x},\{\mathbf{y}_1,\ldots,\mathbf{y}_K\}}\left[||D_f(\bar{\mathbf{x}})) - \sum_k \frac{D_f(\mathbf{y}_k)}{K}||_1^1\right].$$

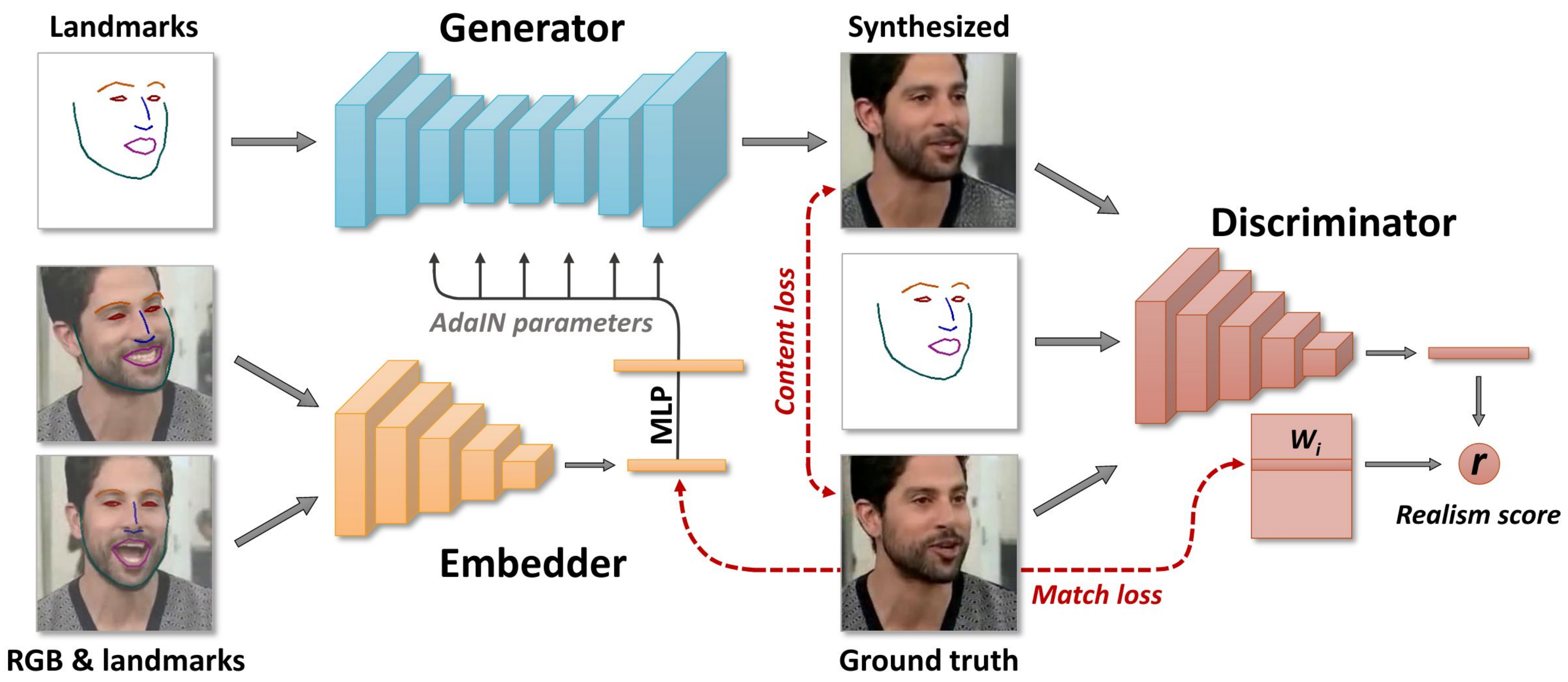
Few-Shot Adversarial Learning of Realistic Neural Talking Head Models

Egor Zakharov[1,2] Aliaksandra Shysheya[1,2] Egor Burkov[1,2] Victor Lempitsky[1,2]

[1]Samsung AI Center, Moscow [2]Skolkovo Institute of Science and Technology


Source
Target → Landmarks → Result


Source
Target → Landmarks → Result



Landmarks
Generator
Synthesized
Discriminator
AdaIN parameters
MLP
Content loss
RGB & landmarks
Embedder
W_i
r
Realism score
Ground truth
Match loss

Loss(G, E)

$$\mathcal{L}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = \mathcal{L}_{\text{CNT}}(\phi, \psi, \mathbf{P}) + \\ \mathcal{L}_{\text{ADV}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) + \mathcal{L}_{\text{MCH}}(\phi, \mathbf{W})$$

$$\mathcal{L}_{\text{ADV}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = \\ -D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b) + \mathcal{L}_{\text{FM}}$$

$$\mathcal{L}_{\text{FM}}(G, D_k) = \mathbb{E}_{(\mathbf{s},\mathbf{x})} \sum_{i=1}^{T} \frac{1}{N_i} [||D_k^{(i)}(\mathbf{s}, \mathbf{x}) - D_k^{(i)}(\mathbf{s}, G(\mathbf{s}))||_1$$

$$D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i;\ \theta, \mathbf{W}, \mathbf{w}_0, b) = \\ V\left(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t);\ \theta\right)^T (\mathbf{W}_i + \mathbf{w}_0) + b$$

Loss Discriminator

$$
\begin{aligned}
\mathcal{L}_{\text{DSC}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = \\
\max(0, 1 + D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i;\ \phi, \psi, \theta, \mathbf{W}, \mathbf{w}_0, b)) + \\
\max(0, 1 - D(\mathbf{x}_i(t), \mathbf{y}_i(t), i;\ \theta, \mathbf{W}, \mathbf{w}_0, b)) .
\end{aligned}
$$

Summary

- Few shot learning 的一種方式是
    - 將資料編碼
    - 利用少量資料的編碼來設定模型
    - GAN 可以用這種方式來達到 Few shot learning

Data Augmentation GAN

Data Augmentation GAN

Antreas Antoniou, Amos Storkey, Harrison Edwards, “Data Augmentation Generative Adversarial Networks”

2017

2020

Motamed, Saman, and Farzad Khalvati. “Inception Augmentation Generative Adversarial Network.” arXiv preprint arXiv:2006.03622

DAGAN

作用在 VGG-Face 的結果(Test Accuracy)如下

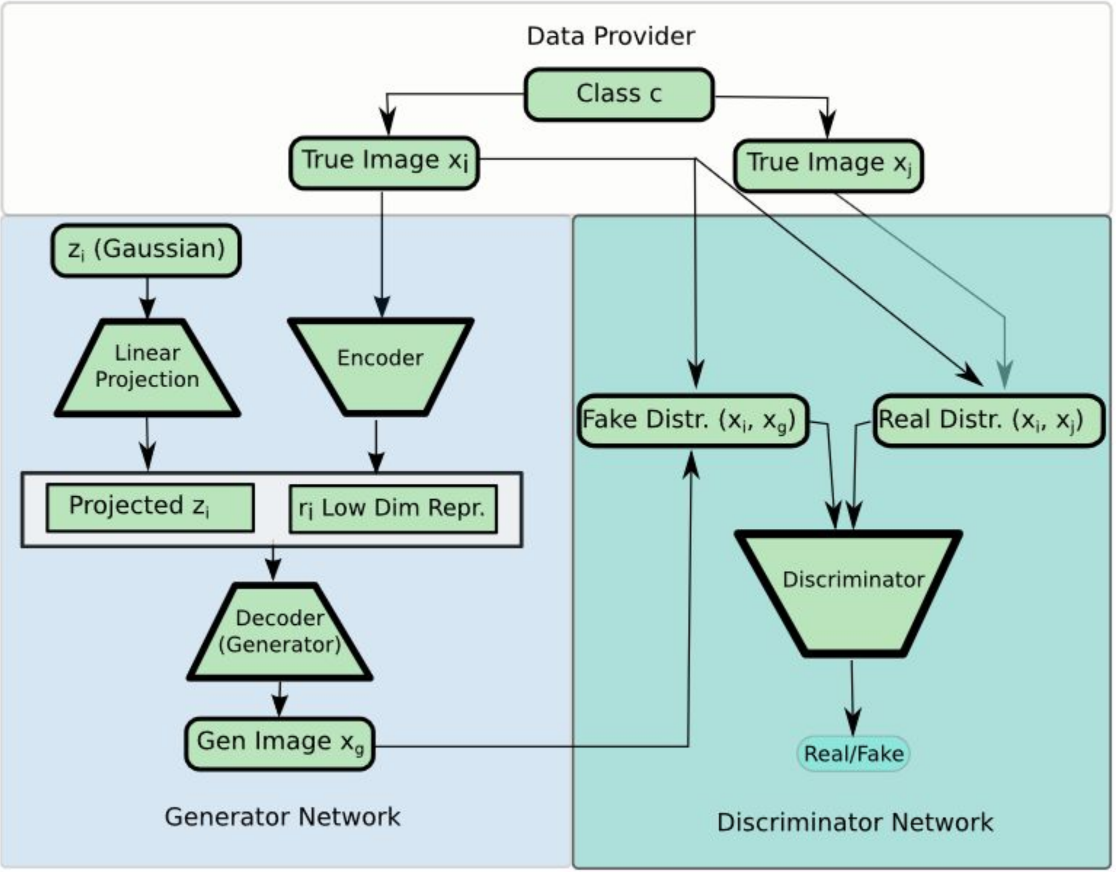
# Sample…	# Standard	# DAGAN
5	4.4%	12.6%
15	39.3%	42.9%
25	58%	58.5%

直接用 GAN 來 augmentation

利用 GAN 來學 augmentation.

- 給定輸入圖片
- 先取得 representation ,
- 然後利用生成模型　和 noise　來生出

這裡的生成模型可以視為 unsupervised meta-learning.

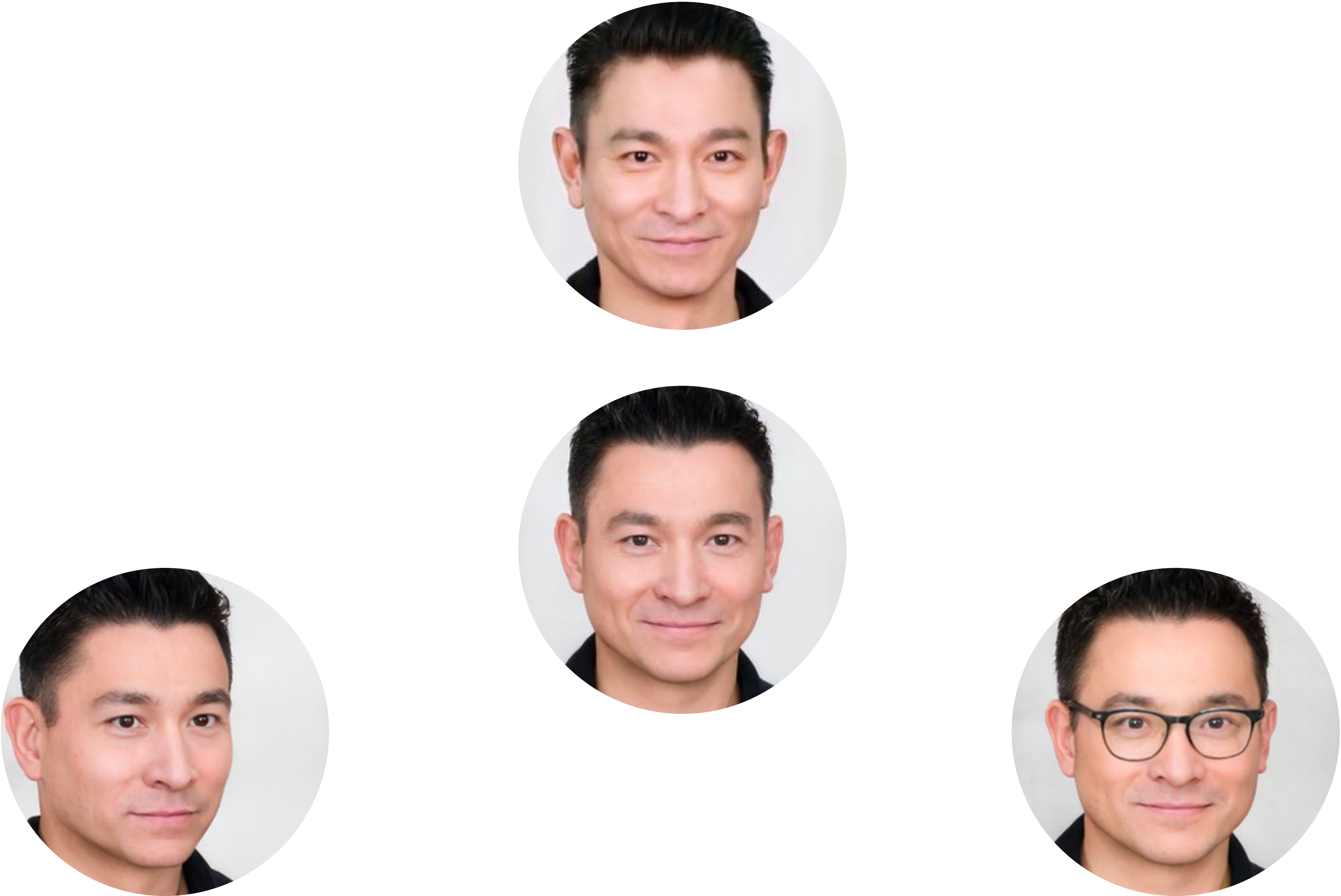

Data Provider
Class c
True Image x_i
True Image x_j
z_i (Gaussian)
Linear Projection
Encoder
Projected z_i
r_i Low Dim Repr.
Decoder (Generator)
Gen Image x_g
Generator Network
Fake Distr. (x_i, x_g)
Real Distr. (x_i, x_j)
Discriminator
Real/Fake
Discriminator Network









DAGAN

作用在 VGG-Face 的結果(Test Accuracy)如下

# Sample…	# Standard	# DAGAN
5	4.4%	12.6%
15	39.3%	42.9%
25	58%	58.5%

Inception-Augmentation GAN

架構類似 DAGAN，也是由輸入的圖片和 noise 產生新圖片

- 但包含較為先進的結構，像是attention, inception blocks

輸入的圖片先

- 經過 CNN 和 attention 降低維度
- 和 noise concate
- 然後再進入後面的網路。

訓練 GAN 的時候:

- 輸入 x, 然後希望生出來的圖片盡量和x 不同(以 SSIM 衡量)
- 但又符合原來的distribution (conditional GAN 的意義下)。

當 GAN 訓練完後

- 利用 AnoGAN 的方式來判斷異常。

原始資料有 3265 張的肺部圖片。

- 在沒有 augmentation 的情形下, AUC=0.83
- 使用 IAGAN, AUC=0.88.

這個方法主要的優點是，3265 張的數量不多

- 在其他 GAN 失效的情形下，利用圖片輔助生成，可以讓GAN 更容易生成。

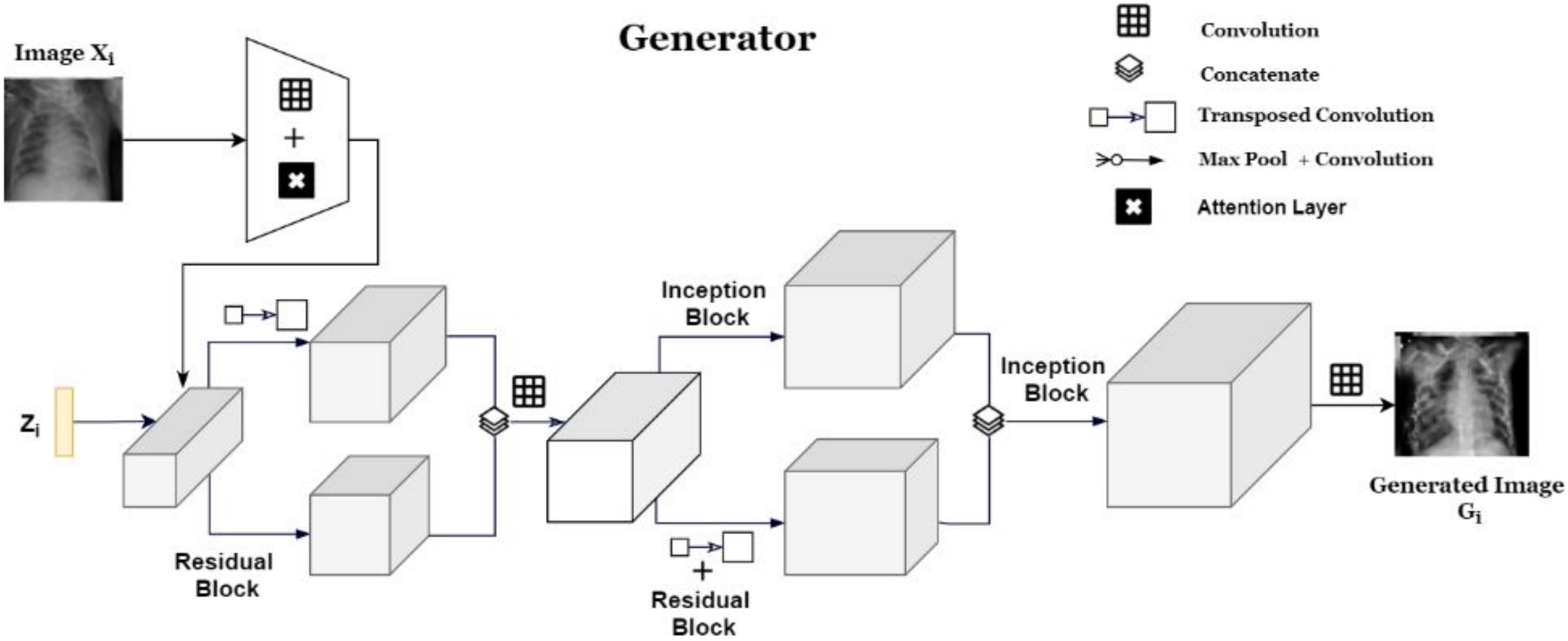


Figure 1: IAGAN's Generator Architecture

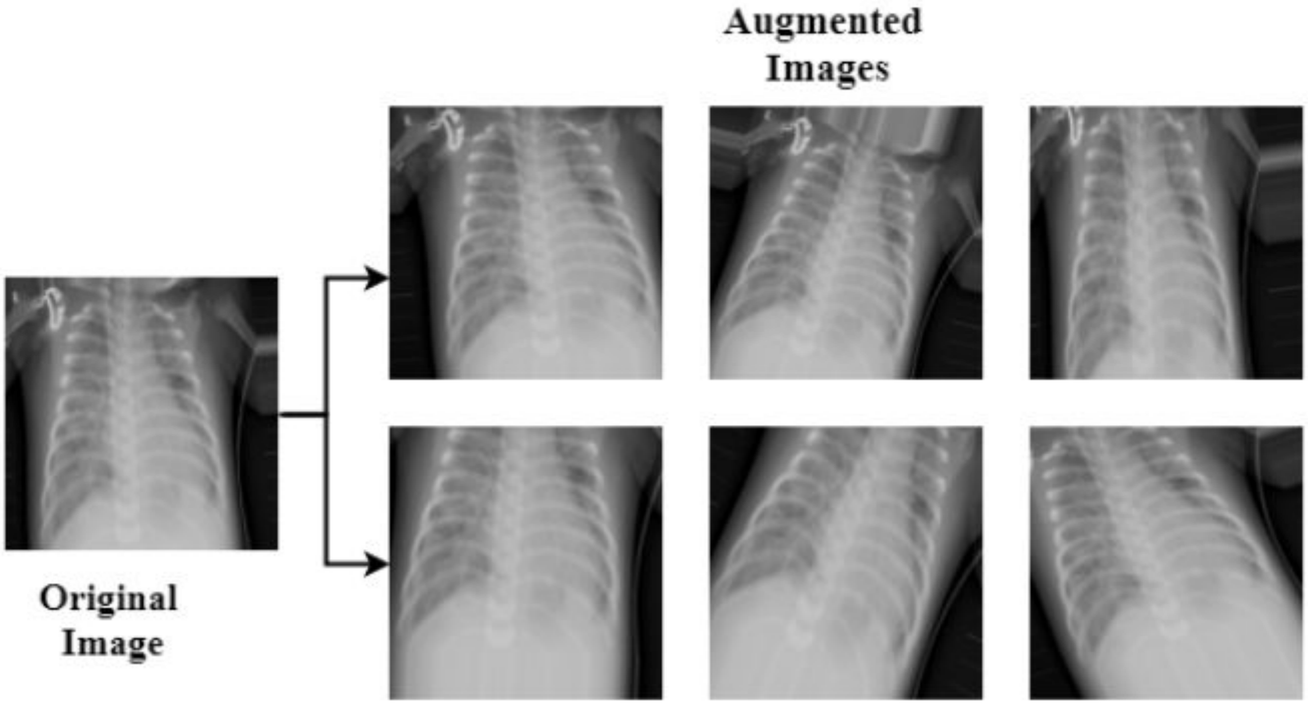


Figure 6: Traditional augmentation output sample

Summary

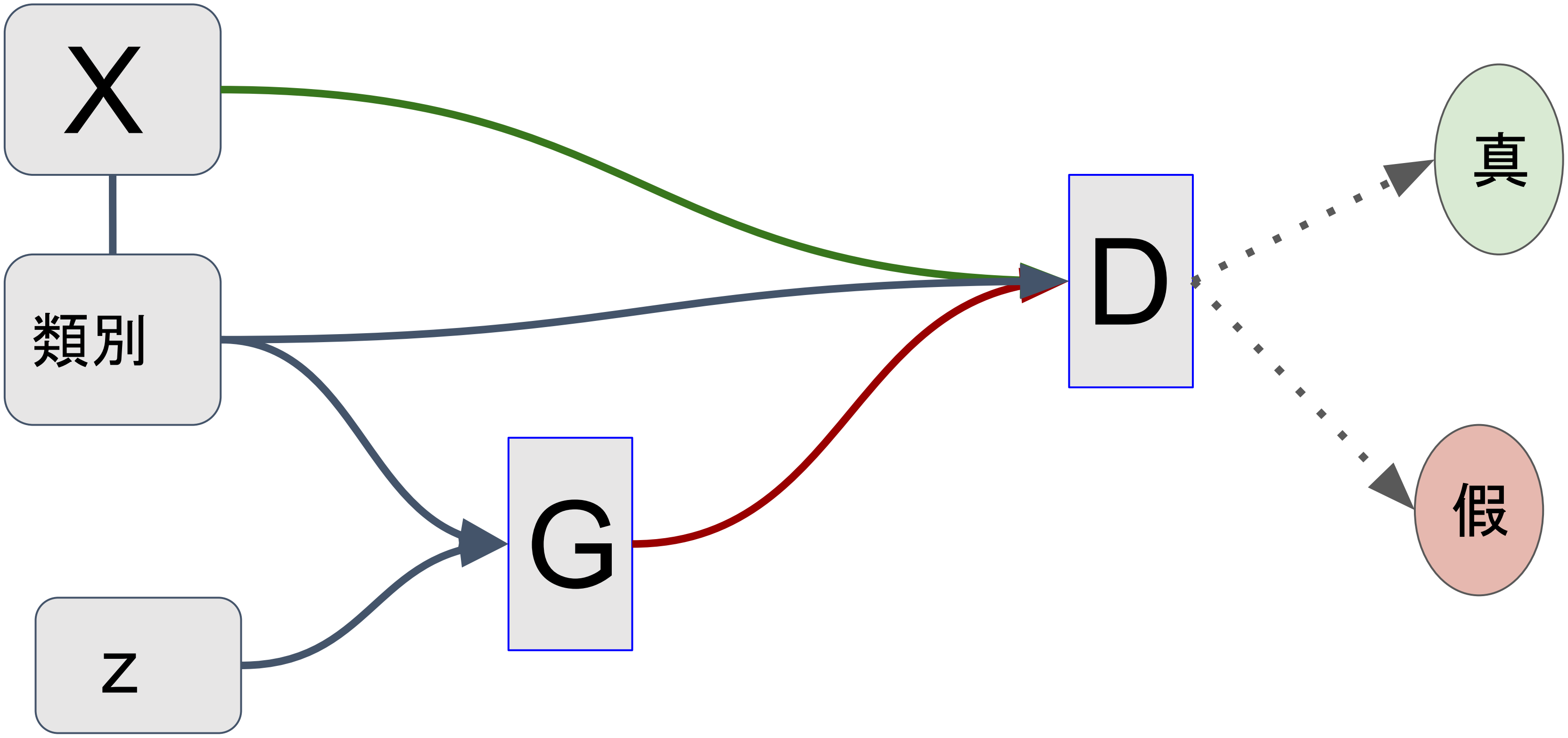
- 更多不同的 GAN 使用方式
- 利用圖到圖來
  - 降低訓練集數量的需求
  - 提升生成圖片的品質

StyleGAN2-ADA 特點是

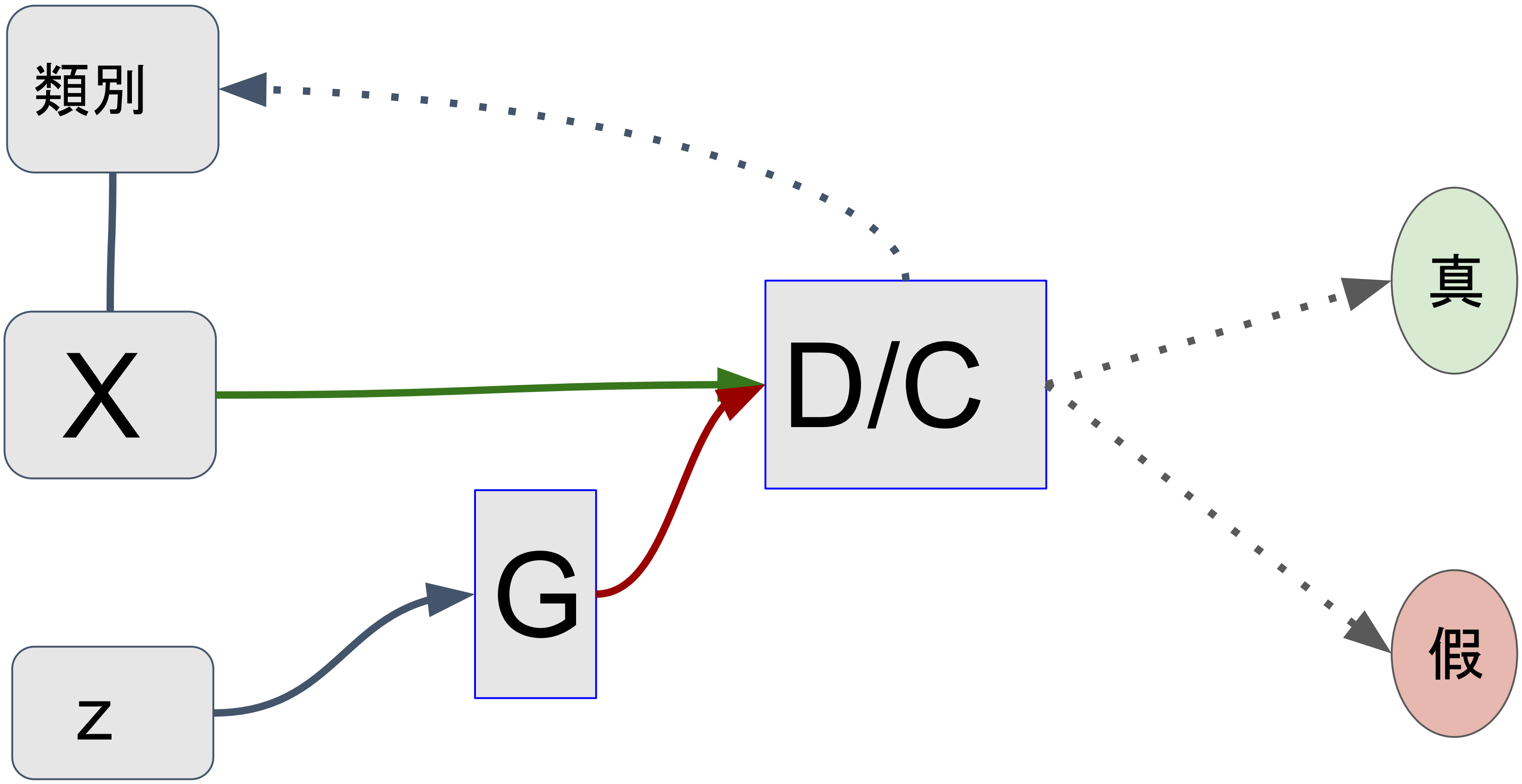
Review

(semi-supervised GAN)

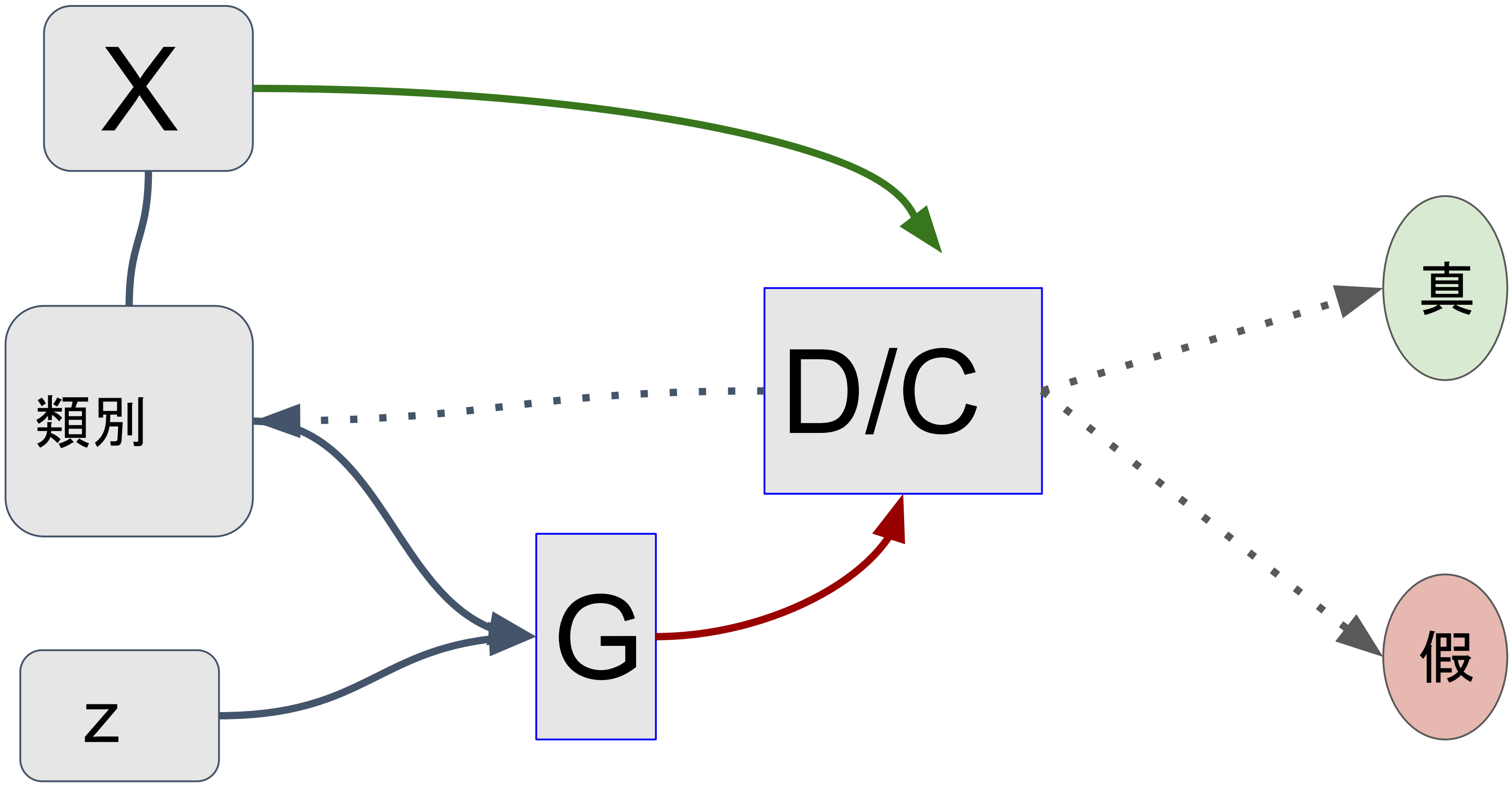
ACGAN	• https://arxiv.org/abs/1610.09585
InfoGAN	• https://arxiv.org/abs/1606.03657
Conditional GAN	• https://arxiv.org/abs/1411.1784
Semi-supervised GAN	• https://arxiv.org/abs/1606.01583



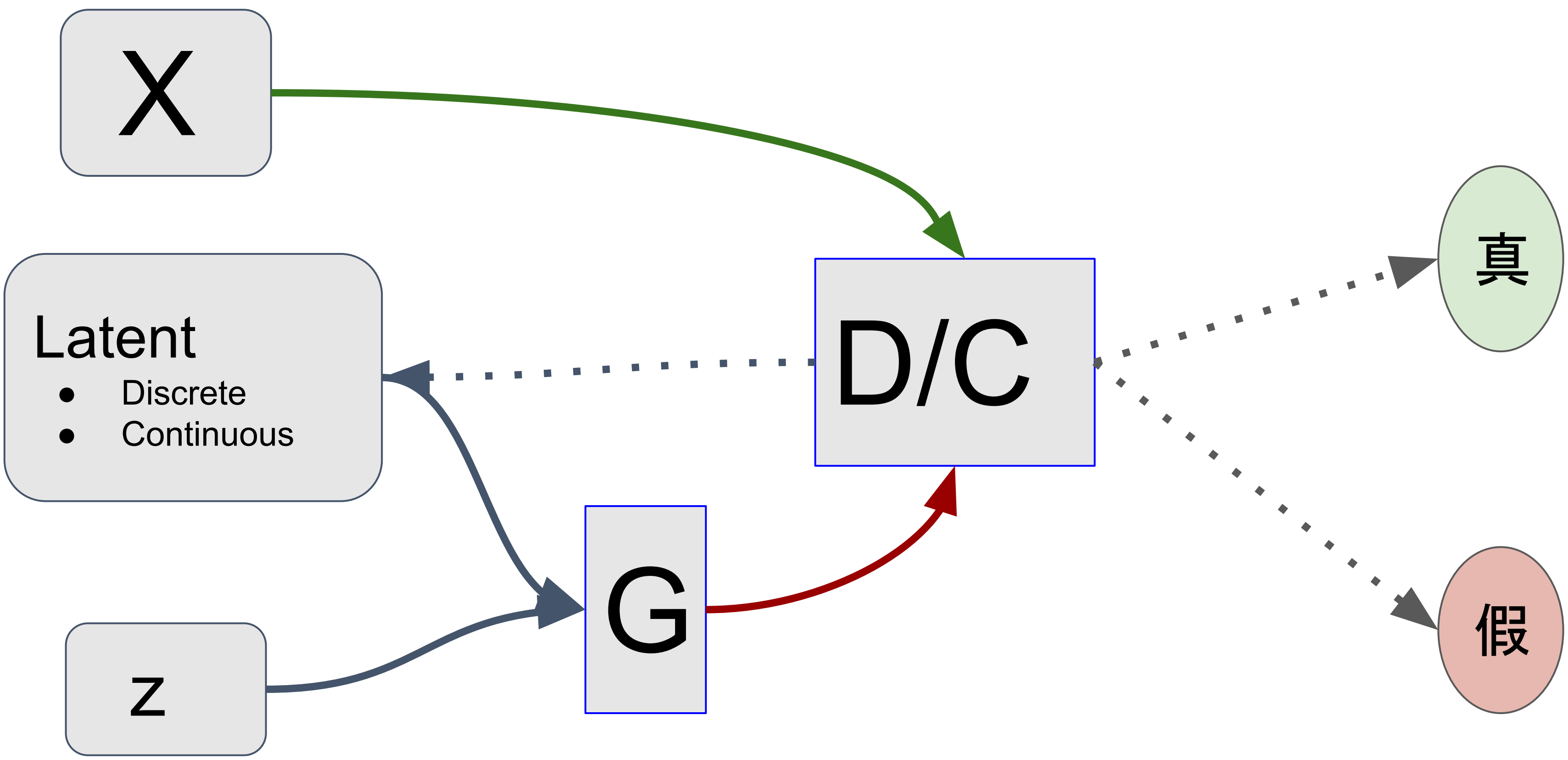
Conditional GAN



Semi-Supervised GAN



ACGAN

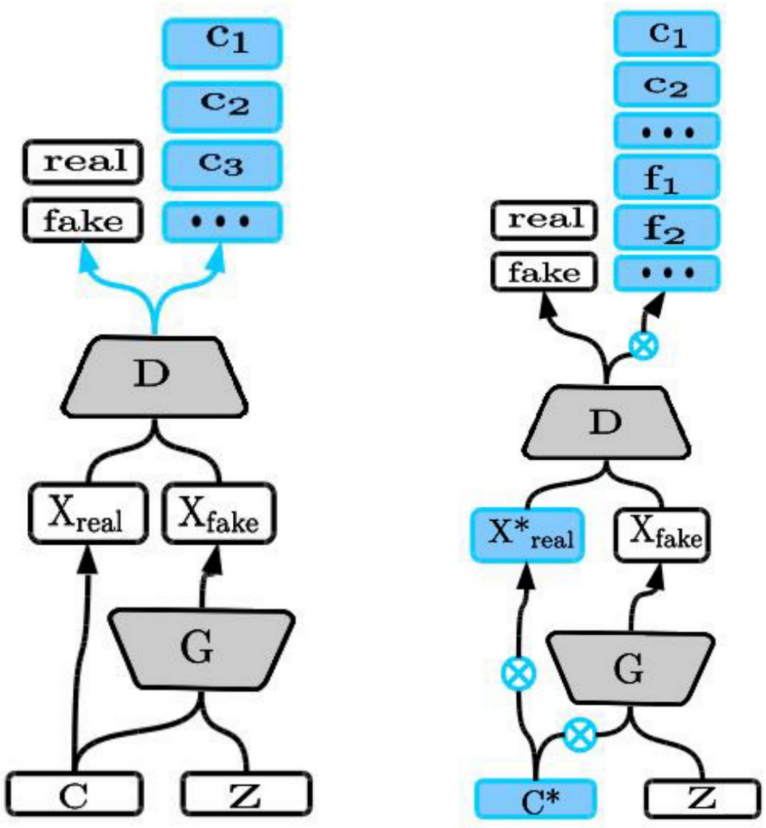


InfoGAN

GAN Based Few-shot Learning

Few-shot classifier GAN

- http://www.animlife.com/publications/ijcnn18.pdf

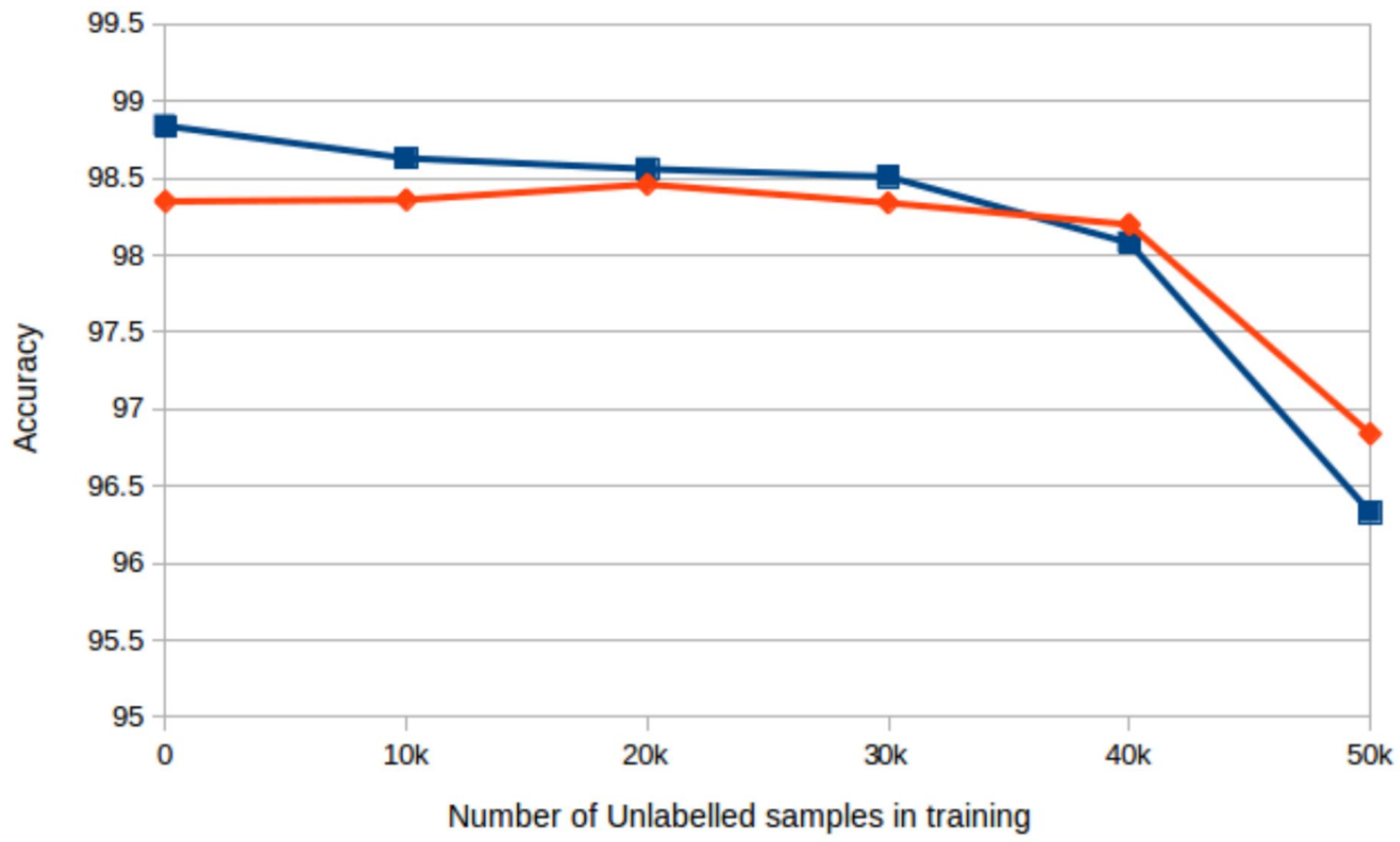


(d) ACGAN [27]　(e) Our FSCGAN

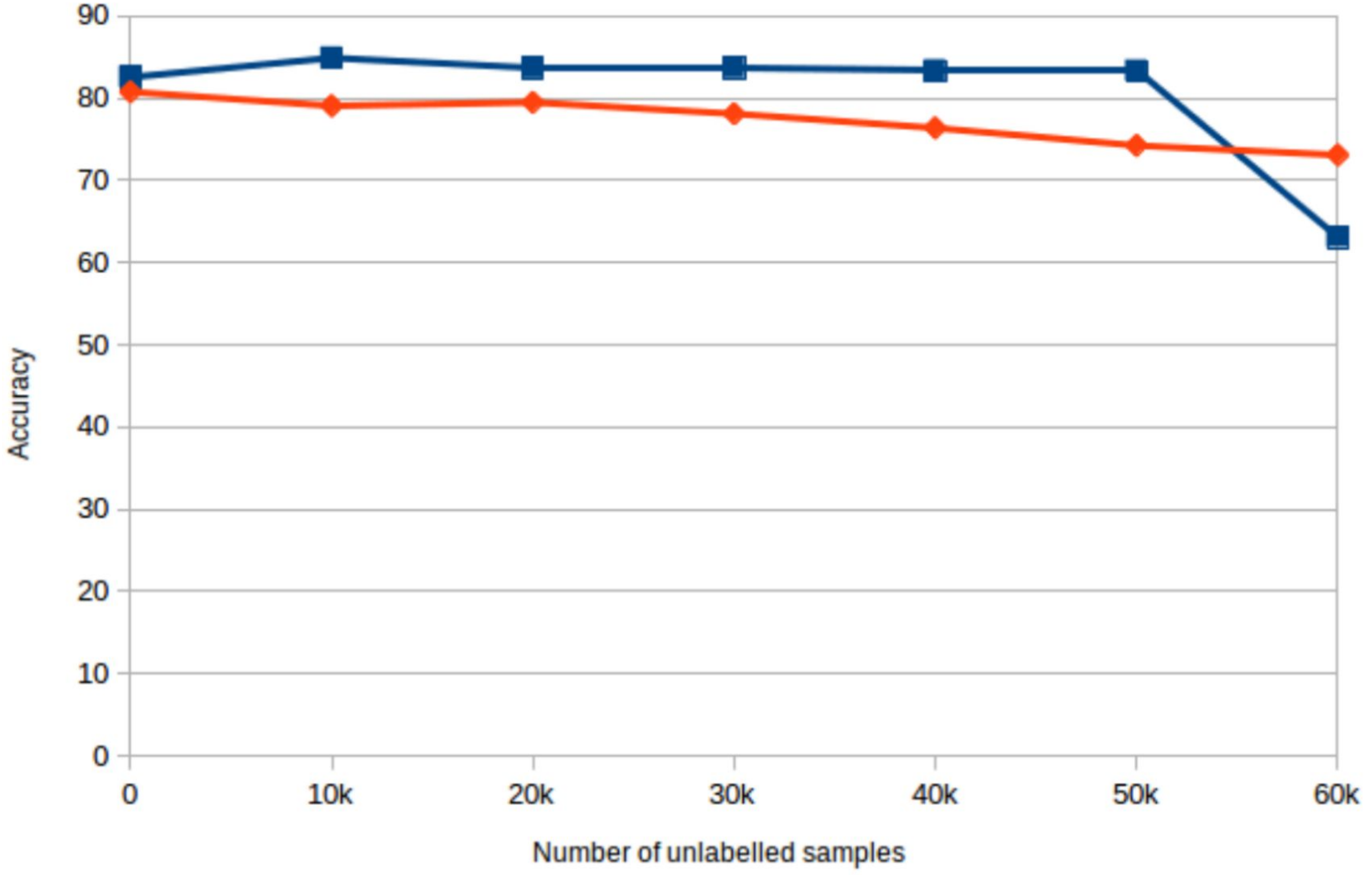
- 類似 ACGAN
  - 但有時沒有 condition
  - 利用網路內的 Switcher 選擇

Algorithm 1 Training Algorithm

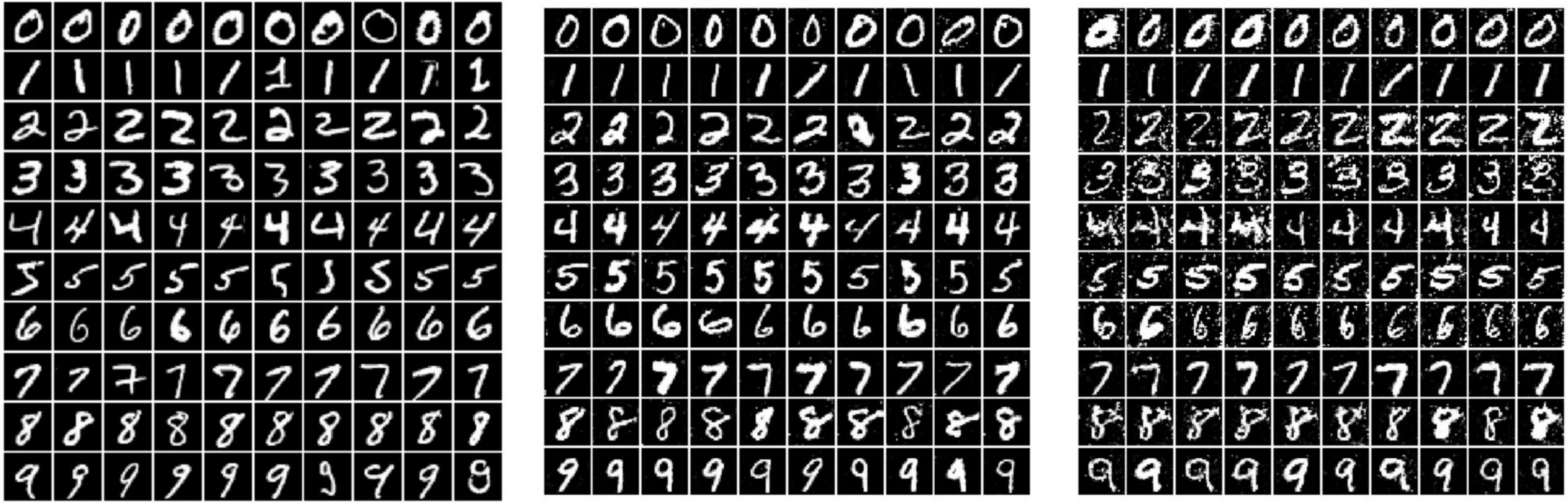
```
procedure TRAIN(data_batches, label_batches)
    for e epochs do
        for k steps do
            Fetch next labeled mini batches
            Perform Stochastic Gradient Descent on D
            Perform Stochastic Gradient Descent on G
            Evaluate(L_s, L_c)
            Update D and G losses
        end for
        for j steps do
            Fetch next unlabeled data mini batches
            Perform Stochastic Gradient Descent on D
            Perform Stochastic Gradient Descent on G
            Evaluate(V_s)
            Update D and G losses
        end for
    end for
end procedure
```



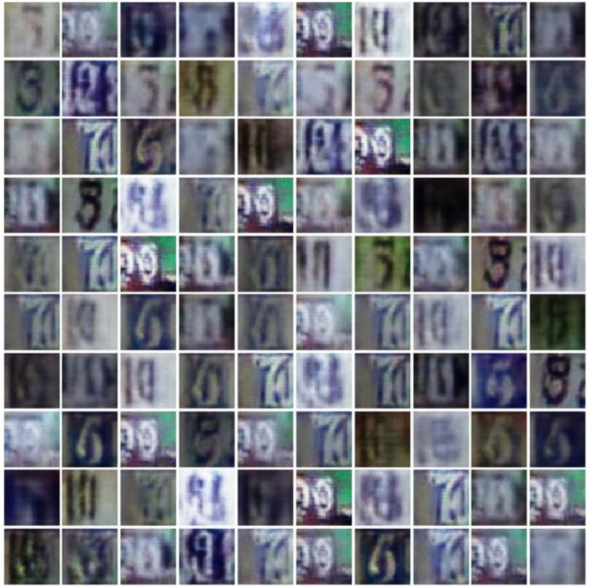
Accuracy for MNIST



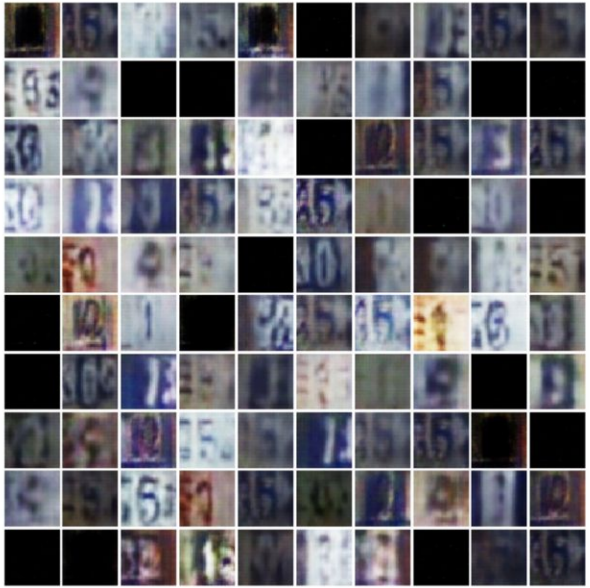
Accuracy for SVHN



Original sample



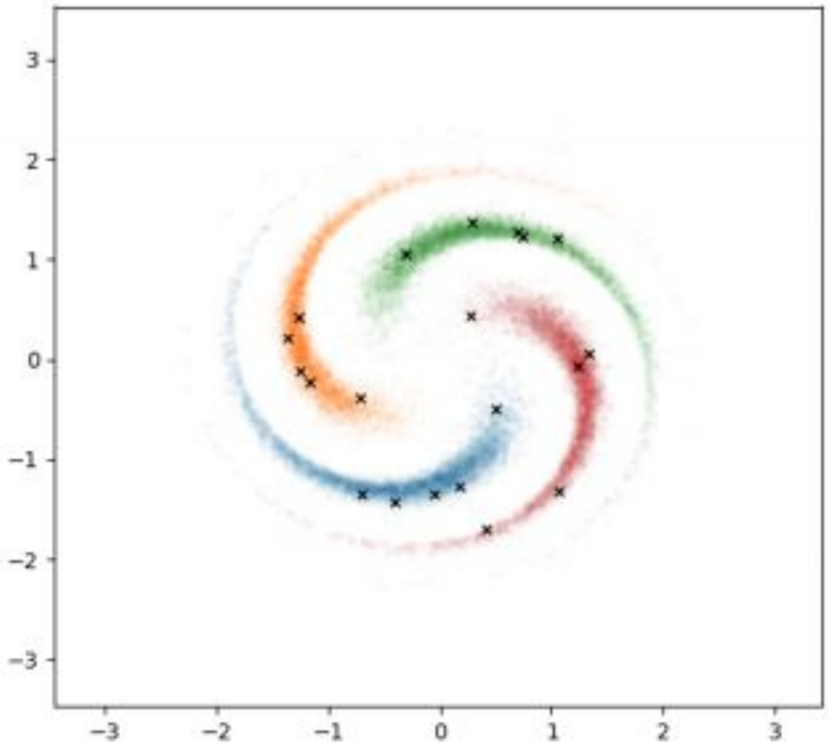
Generated sample (single fake class)

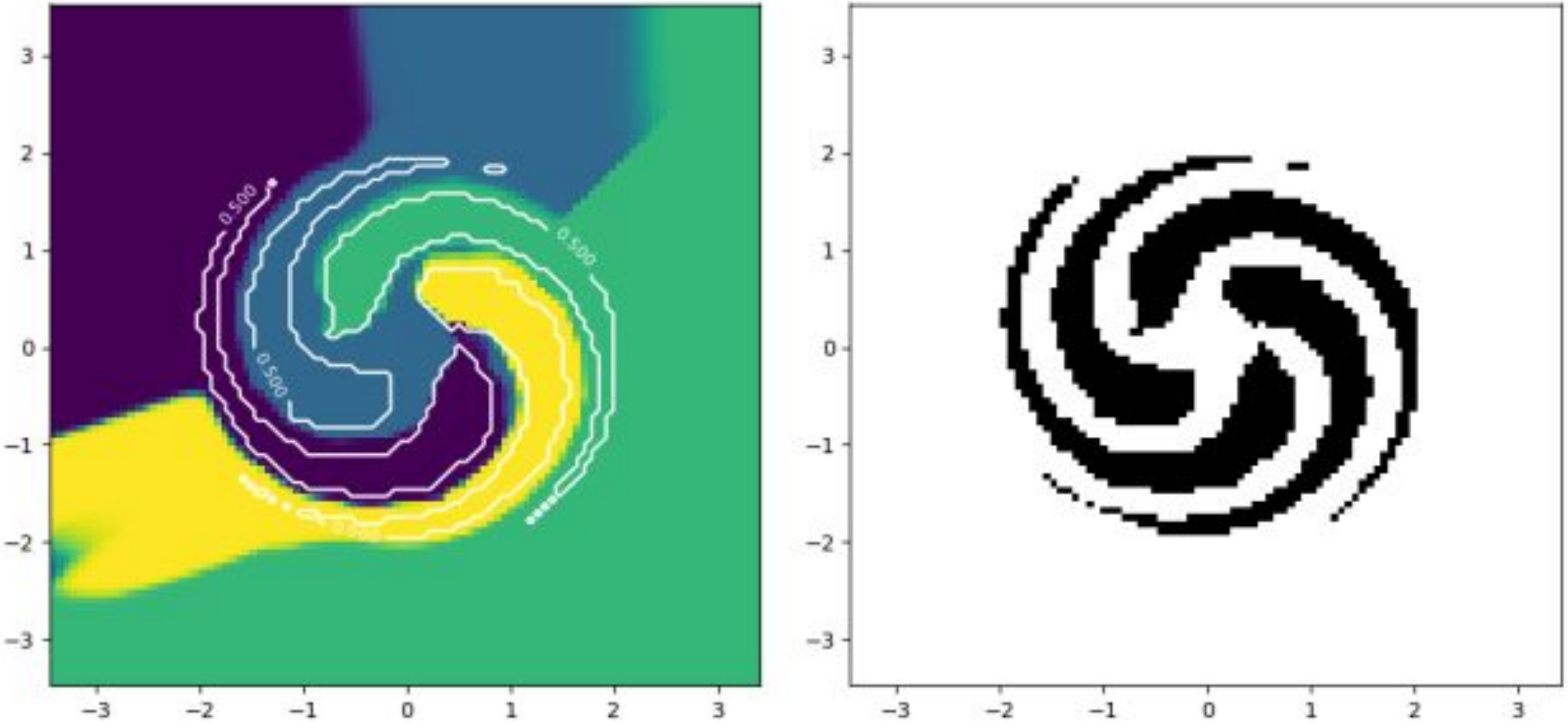


Generated sample (multiple fake classes)

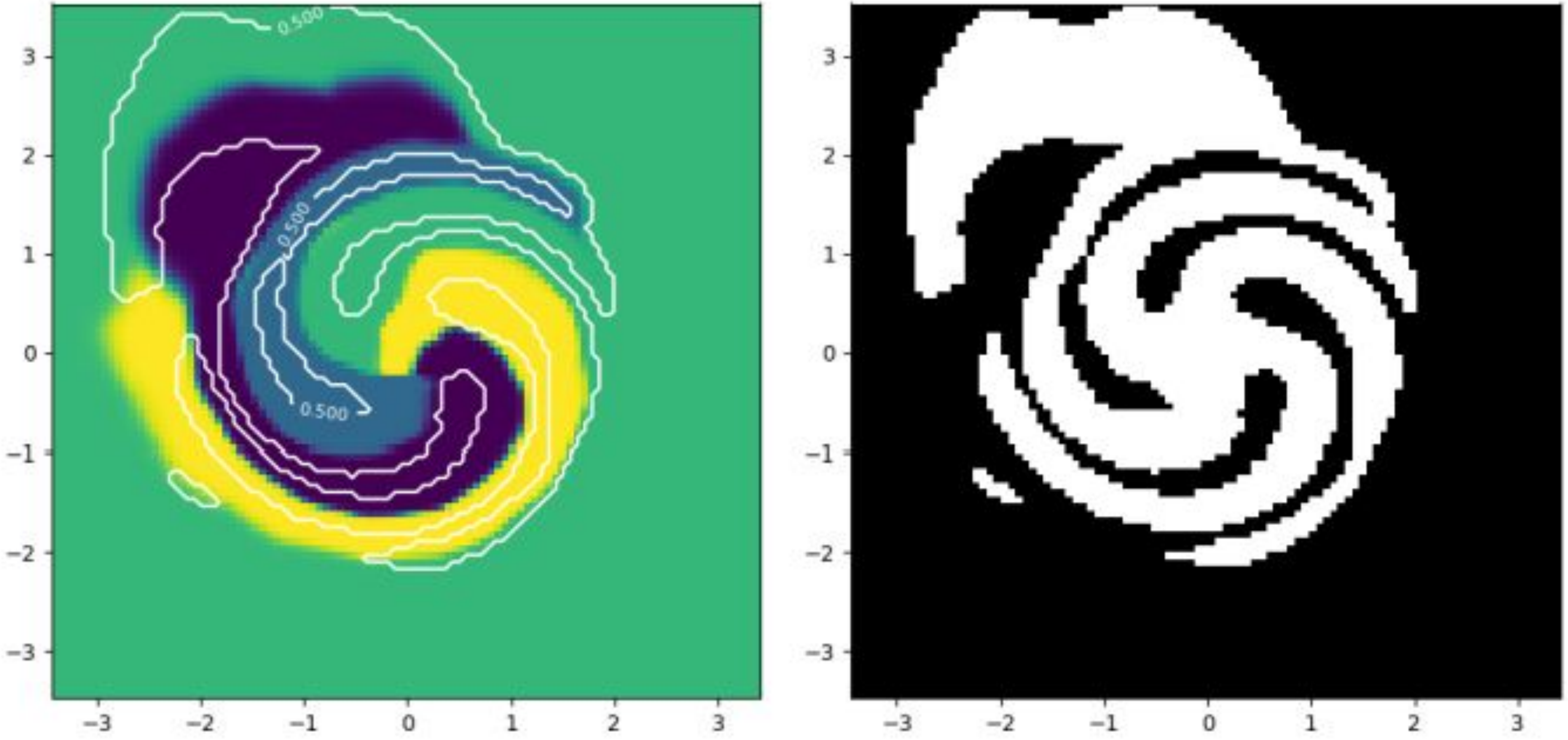
Good semi-supervised learning that requires a bad GAN.

- https://proceedings.neurips.cc/paper/2017/file/79514e888b8f2acacc68738d0cbb803e-Paper.pdf


3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3



0.500
0.500
0.500
0.500
−3
−2
−1
0
1
2
3



0.500
0.500
0.500
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Summary

- GAN 可以靈活應用未標記的資料
- 即使 GAN 生成的品質不好，也可能有幫助
- GAN 生出的東西當成是瑕疵可能可以在真實資料附近建立更緊密的邊界

GAN powerd Meta-Learning

MetaGAN

- https://papers.nips.cc/paper/2018/hash/4e4e53aa080247bc31d0eb4e7aeb07a0-Abstract.html

Settings of Few Shot Learning

- K-shot N-way supervised learning
- Sample-level Semi-supervised Learning
- Task-level Semi-supervised Learning

Task

- Predict y given x
- Support set $S_T = (S_T^s, S_T^u)$
  - $S_T^s = \{(x_i, y_i)\}$: labeled data
  - S_T^u: unlabeled data
- Query set $Q_T = (Q_T^s, Q_T^u)$

Dataset

- Training: contains tasks
  - (Support, Query)
- Testing: contains tasks
  - (Support, Query)

Sample-level Semi-supervised Learning

- Allow some training samples to be unlabeled within a task
- These training samples can either come from the same classes as the labeled samples
- or come from different "distractor" classes

Task-level Semi-supervised Learning

- also allow purely unsupervised task
  - in which both support and query samples are all unlabeled
- can be very natural in practice

Discriminator

- Use decent few-shot learners, e.g.
  - MAML
  - Relation Network
- Add an additional output, as in semi-supervised learning
  - So there are N+1 outputs

Loss

$$
\begin{aligned}
\mathcal{L}_D^{\mathcal{T}} &= \mathcal{L}_{\text{supervised}} + \mathcal{L}_{\text{unsupervised}}, \\
\mathcal{L}_{\text{supervised}} &= \mathbb{E}_{\mathbf{x},y\sim Q_{\mathcal{T}}^{s}} \log p_D(y|\mathbf{x}, y \le N) \\
\mathcal{L}_{\text{unsupervised}} &= \mathbb{E}_{\mathbf{x}\sim Q_{\mathcal{T}}^{u}} \log p_D(y \le N|\mathbf{x}) + \mathbb{E}_{\mathbf{x}\sim p_G^{\mathcal{T}}} \log p_D(N+1|\mathbf{x})
\end{aligned}
$$

Generator

- Minimize

$$L_G^{\mathcal{T}}(D) = -\mathbb{E}_{\mathbf{x} \sim p_G^{\mathcal{T}}}[\log(p_D(y \leq N|\mathbf{x}))].$$

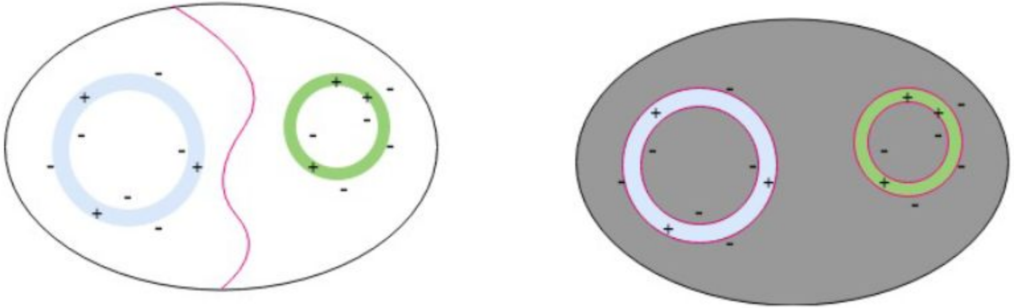
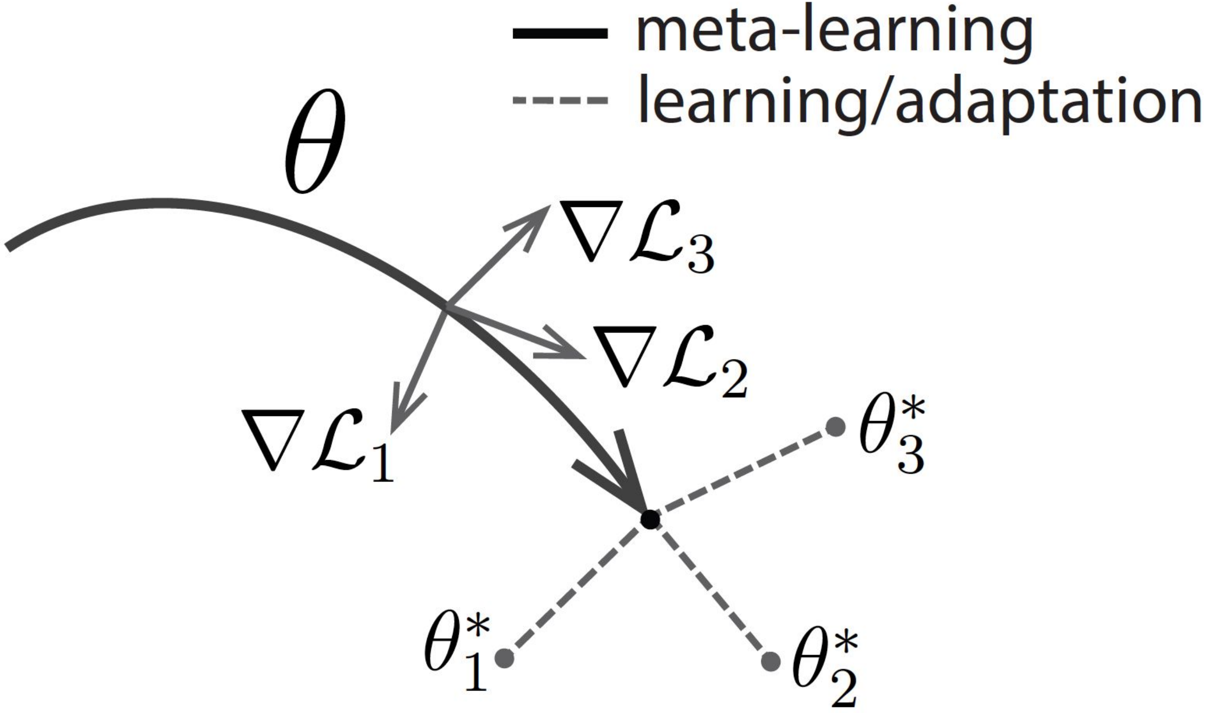


Figure 1: Left: decision boundary without metaGAN. Right: decision boundary with metaGAN. We use red curves to denote the decision boundary. Blue area in figure represents class A, green area represents class B, and gray area represents fake class. We use $+$ to denote real samples and $-$ to denote fake samples generated.

MAML



meta-learning
learning/adaptation
θ
$\nabla\mathcal{L}_3$
$\nabla\mathcal{L}_2$
$\nabla\mathcal{L}_1$
θ_3^*
θ_1^*
θ_2^*

GAN With MAML

- Use the MAML model as discriminator
- For a specific task, update the model
- Update the discriminator according to the loss

$$\ell_D^{\mathcal{T}} = -\mathbb{E}_{\mathbf{x},y\sim S_{\mathcal{T}}^s} \log p_D(y|\mathbf{x}, y \leq N) - \mathbb{E}_{\mathbf{x}\sim S_{\mathcal{T}}^u} \log p_D(y \leq N|\mathbf{x}) - \mathbb{E}_{\mathbf{x}\sim p_G^{\mathcal{T}}} \log p_D(N+1|\mathbf{x})$$

- Train the generator using the adapted discriminator

Algorithm 1 MetaGAN with MAML

$G(\mathrm{z}, \mathcal{T})$: **Generator network parameterized by** θ_g.
$D(x)$: **Discriminator network. parameterized by** θ_d.

Initialize θ_g, θ_d randomly.
while not done **do**
    Sample a batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$. ▷ Discriminator Update
    for all $\mathcal{T}_i$ **do**
        Get K real samples $\mathcal{D}_r = \{\mathbf{x}^{(i)}, y^{(i)}\}$ from $\mathcal{T}_i$.
        Sample K generated samples $\mathcal{D}_f = \{\mathbf{x}^{(j)}\} = G(\mathbf{z}^{(j)}, \mathcal{T}_i)$ from $G(\mathbf{z}, \mathcal{T}_i)$.
        Evaluate discriminator loss $\ell_D^{\mathcal{T}_i}$ with D_r and D_f.
        Compute adapted discriminator parameters $\theta'_{d_i} = \theta_d - \alpha \nabla_{\theta_d} \ell_D^{\mathcal{T}_i}$.
    end for
    Update θ_d using loss $\mathcal{L}_D$
    Sample a batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$. ▷ Generator Update
    for all $\mathcal{T}_i$ **do**
        Sample K generated samples $\mathcal{D}_f = \{\mathbf{x}^{(j)} = G(\mathbf{z}^{(j)}, \mathcal{T}_i)\}$ from $G(\mathbf{z}, \mathcal{T}_i)$.
        Compute adapted discriminator parameters $\theta'_{d_i} = \theta_d - \alpha \nabla_{\theta_d} L_D$.
        Compute generator loss gradient $\nabla_{\theta_g} L_G^{\mathcal{T}_i}$ with the adapted discriminator.
    end for
    Update generator parameters θ_g with accumulated generator loss gradients.
end while

Relation Network

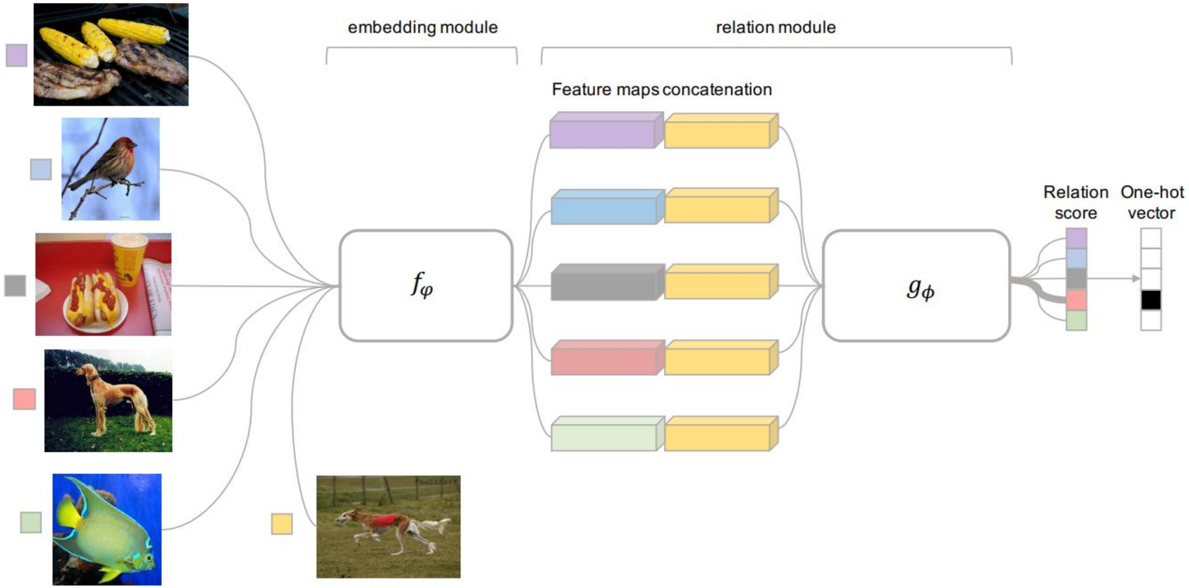


Figure 1: Relation Network architecture for a 5-way 1-shot problem with one query example.

GAN With Relation Network

- Use softmax classification on relevance score between query set and support set

$$p_D(y = k|\mathbf{x}_j) = \frac{\exp(r_{k,j})}{1 + \sum_{i=1}^{N} \exp(r_{i,j})}$$

Model	5-way Acc. 1-shot	5-way Acc. 5-shot	20-way Acc. 1-shot	20-way Acc. 5-shot
Neural Statistician	98.1	99.5	93.2	98.1
Prototypical Nets	98.8	99.7	96.0	98.9
MAML	98.7 ± 0.4	$\mathbf{99.9 \pm 0.1}$	95.8 ± 0.3	98.9 ± 0.2
Ours: MetaGAN + MAML	99.1 ± 0.3	99.7 ± 0.21	96.4 ± 0.27	98.9 ± 0.18
Relation Net	99.6 ± 0.2	99.8 ± 0.1	97.6 ± 0.2	99.1 ± 0.1
Ours: MetaGAN + RN	$\mathbf{99.67 \pm 0.18}$	99.86 ± 0.11	$\mathbf{97.64 \pm 0.17}$	$\mathbf{99.21 \pm 0.1}$

Table 1: Few-shot classification results on Omniglot.

Model	**5-way Acc.** 1-shot	5-shot
Prototypical Nets	49.42 ± 0.78	68.20 ± 0.66
MAML(5 gradient steps)	48.70 ± 1.84	63.11 ± 0.92
MAML(5 gradient steps, first order)	48.07 ± 1.75	63.15 ± 0.91
MAML(1 gradient step, first order)	43.64 ± 1.91	58.72 ± 1.20
Ours: MetaGAN + MAML(1 step, first order)	46.13 ± 1.78	60.71 ± 0.89
Relation Net	50.44 ± 0.82	65.32 ± 0.7
Ours: MetaGAN + RN	$\mathbf{52.71} \pm \mathbf{0.64}$	$\mathbf{68.63} \pm \mathbf{0.67}$

Table 2: Few-shot classification results on Mini-Imagenet.

Model	Omniglot 1-shot 5-way	Mini-Imagenet 1-shot 5-way	Mini-Imagenet 5-shot 5-way
Prototypical Nets(Supervised)	94.62 ± 0.09	43.61 ± 0.27	59.08 ± 0.22
Semi-Supervised Inference(PN)	97.45 ± 0.05	48.98 ± 0.34	63.77 ± 0.20
Soft k-Means	97.25 ± 0.10	50.09 ± 0.45	$\mathbf{64.59 \pm 0.28}$
Soft k-Means+Cluster	$\mathbf{97.68 \pm 0.07}$	49.03 ± 0.24	63.08 ± 0.18
Masked Soft k-Means	97.52 ± 0.07	$\mathbf{50.41 \pm 0.31}$	64.39 ± 0.24
Ours: Relation Nets(Supervised)	94.81 ± 0.08	44.24 ± 0.24	58.72 ± 0.31
Ours: MetaGAN + RN	97.58 ± 0.07	50.35 ± 0.23	64.43 ± 0.27

Table 3: Sample-level Semi-Supervised Few-shot classification results on Omniglot and Mini-Imagenet.

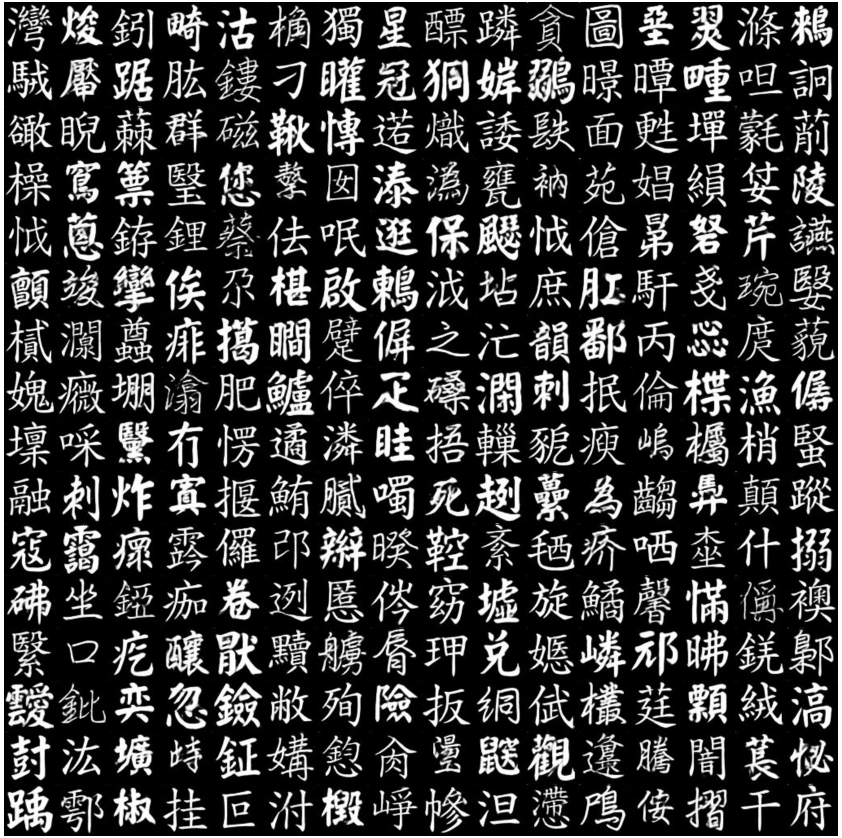
Model	**Omniglot** 1-shot 5-way	**Mini-Imagenet** 1-shot 5-way
Prototypical Net(Supervised)	93.66 ± 0.09	42.28 ± 0.32
Relation Net(Supervised)	93.82 ± 0.07	43.87 ± 0.20
Ours: MetaGAN + RN	$\mathbf{97.12 \pm 0.08}$	$\mathbf{47.43 \pm 0.27}$

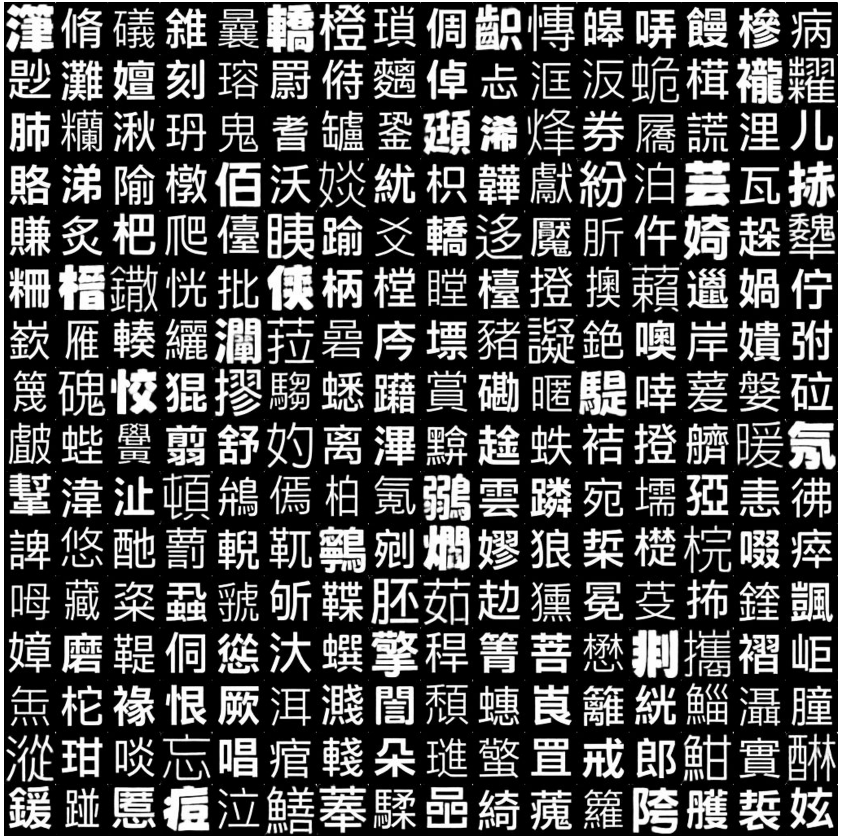
Table 4: Task-level Semi-Supervised 1-shot classification results on Omniglot and Mini-Imagenet.

Summary

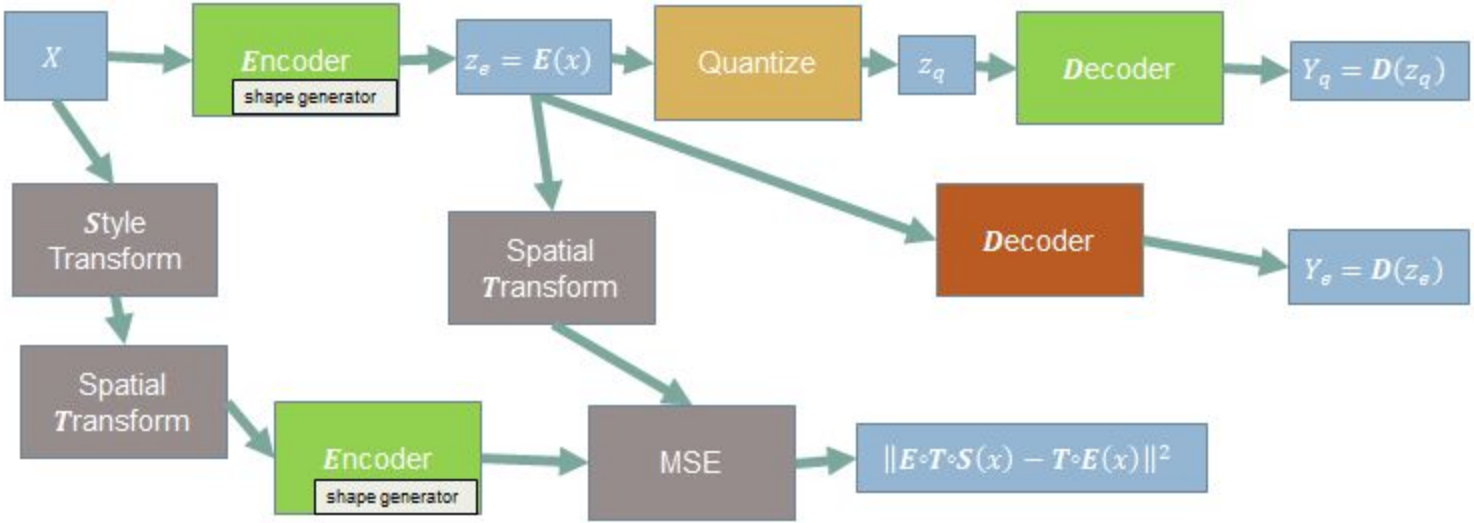
- GAN 可以搭配很多不同的架構
- 包含 Meta-learning
- GAN 在 Few-shot 中可以模仿未知的其他類別

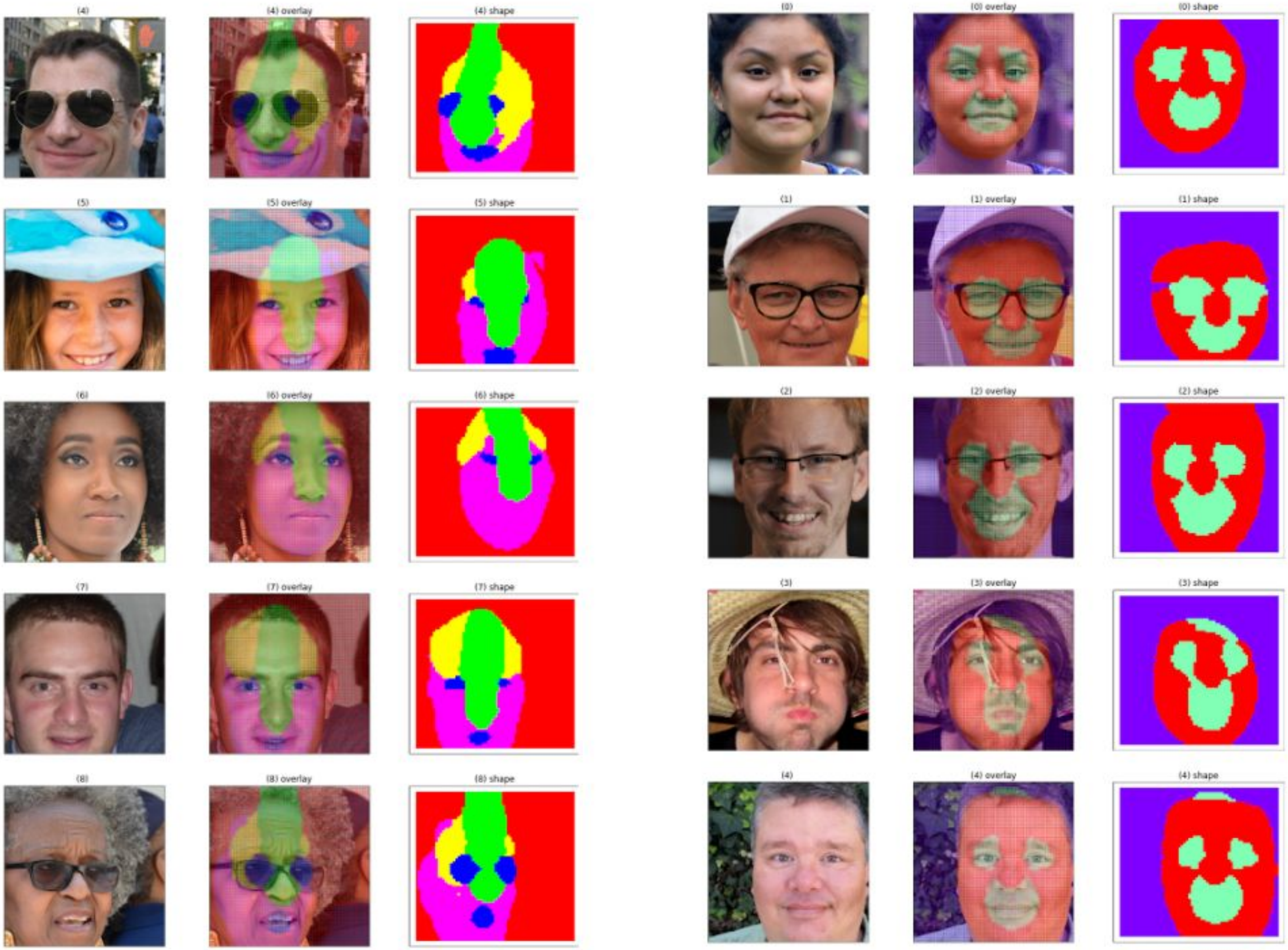
Applications





Architecture





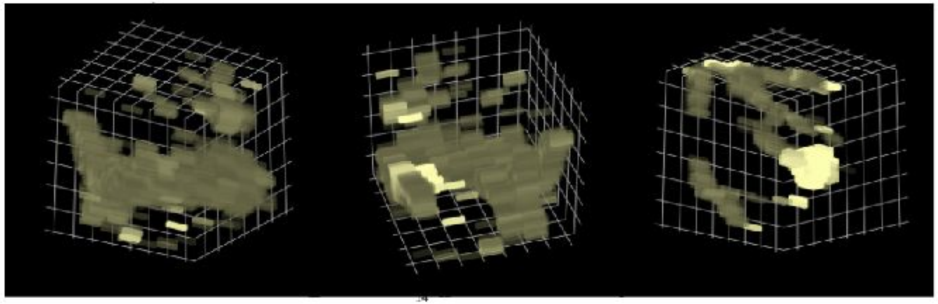
(4)
(4) overlay
(4) shape
(5)
(5) overlay
(5) shape
(6)
(6) overlay
(6) shape
(7)
(7) overlay
(7) shape
(8)
(8) overlay
(8) shape
(0)
(0) overlay
(0) shape
(1)
(1) overlay
(1) shape
(2)
(2) overlay
(2) shape
(3)
(3) overlay
(3) shape
(4)
(4) overlay
(4) shape

GAN based AOI

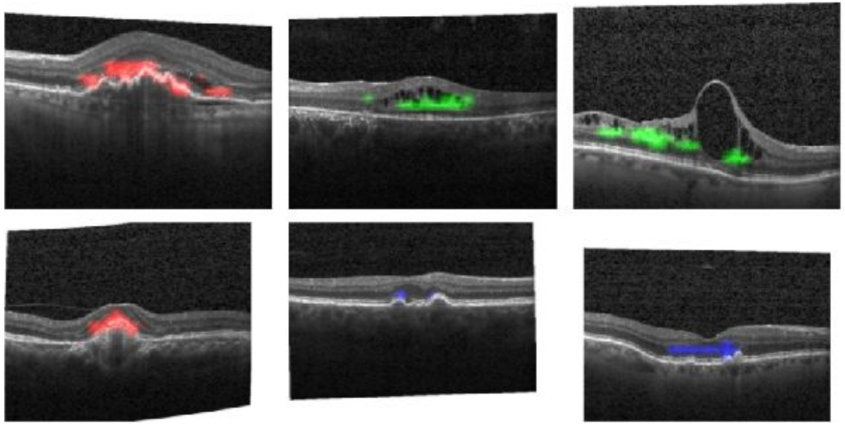
- Data
  - 200K, NG and OK are about 1:2.5
  - We use only 50% of data
- StyleGAN2
  - Generate 100K NG fake
  - FID is about 50
- Result(NG as positive)
  - Specificity>=70%, FNR 0.018%
  - =train with 100% data(without using GAN)
  - More stable

電腦視覺應用

醫療 3D 結核偵測

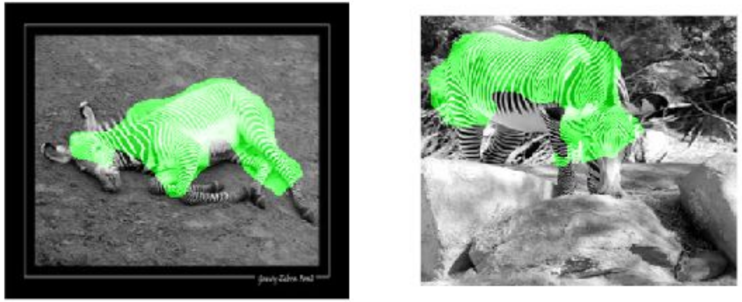


視網膜病變部位偵測



公安問題偵測

區域偵測範例：斑馬



(複選) MetaGAN 設計用來搭配