Analysing Failure Modes in Unsupervised Image-to-Image Translation

Doctoral Qualifying Oral Exam

Boston University 2021
Ben Usman

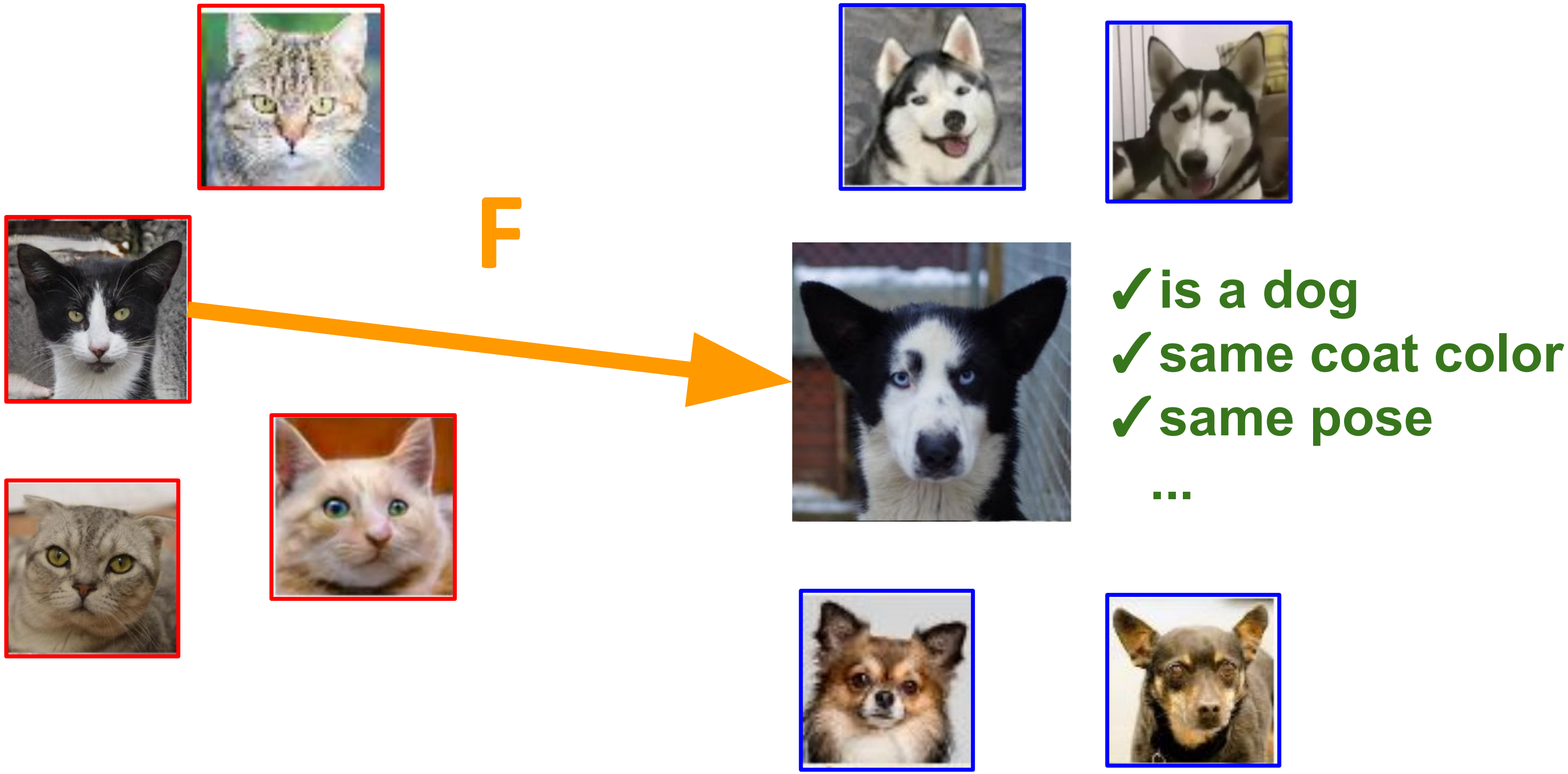
Presentation Plan

1. Problem and motivation
2. Existing solutions and possible failure modes
3. Tools for analysing these failure modes in prior work
   a. **“Generalization and Equilibrium in Generative Adversarial Nets”** by Arora et al., PMLR 2017.
   b. **“Training Generative Adversarial Networks with Limited Data”** by Karras et al., NeurIPS 2020.
   c. **“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”** by Galanti et al., JMLR 2021. // 2017-2021

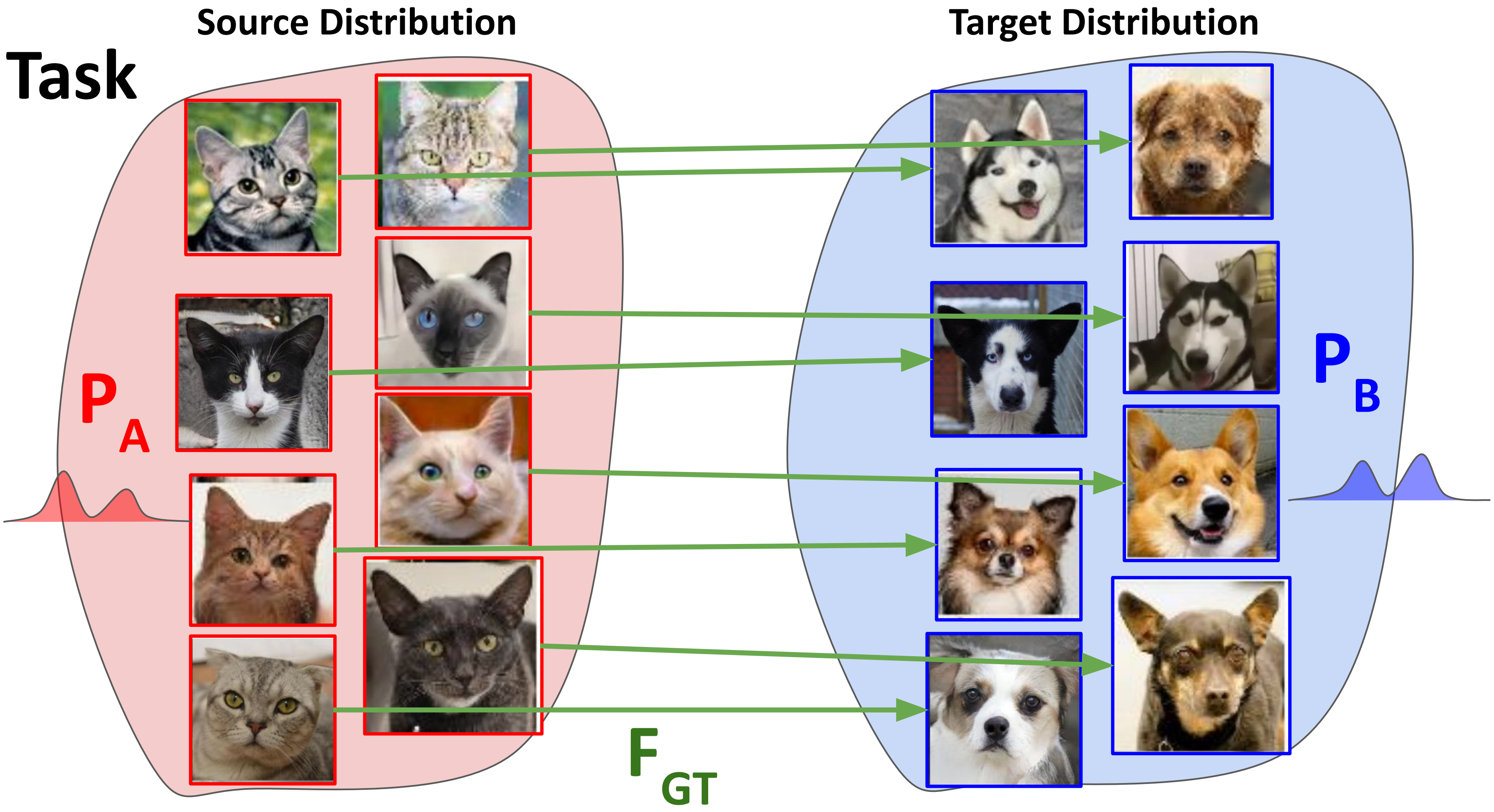
Task

Source Samples (Cats)

Target Samples (Dogs)



Task



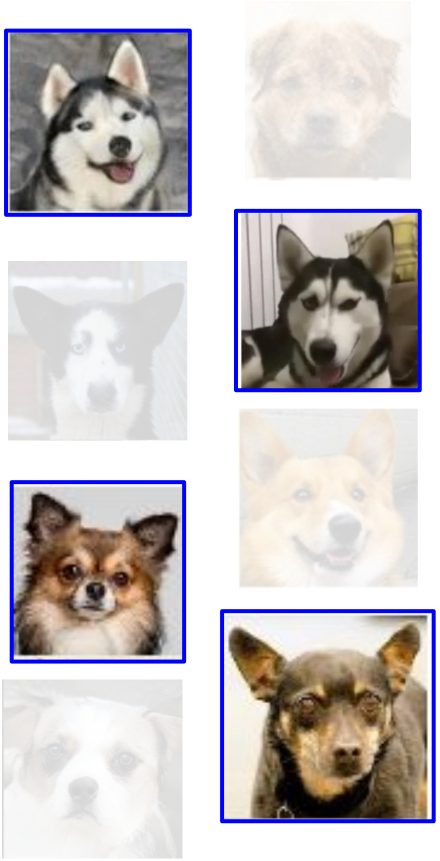
Ground truth 1-to-1 cross-domain mapping.

Task

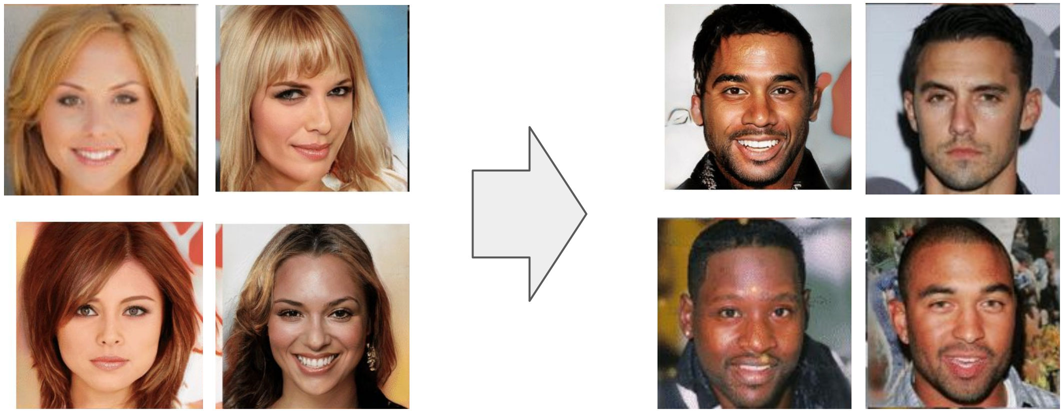
Source Samples

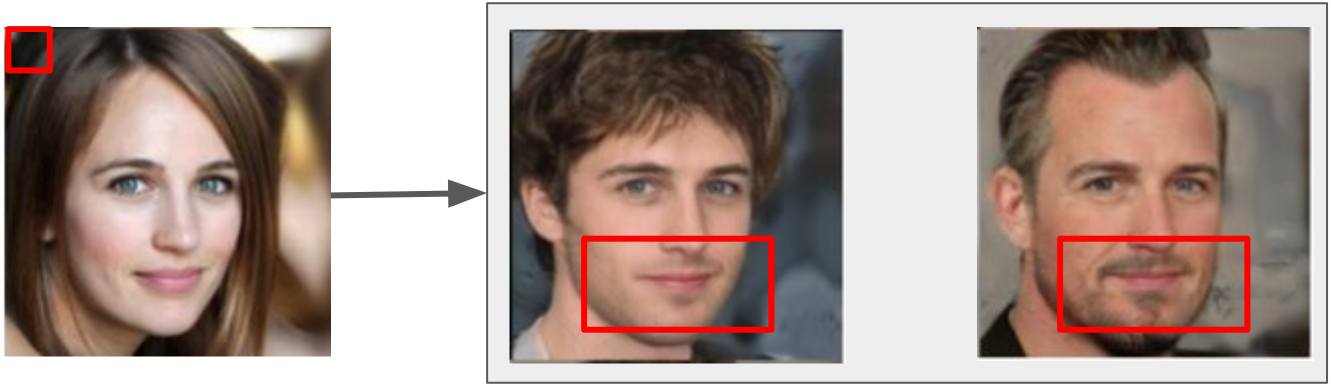
Goal:
reconstruct F from unpaired samples

Target Samples

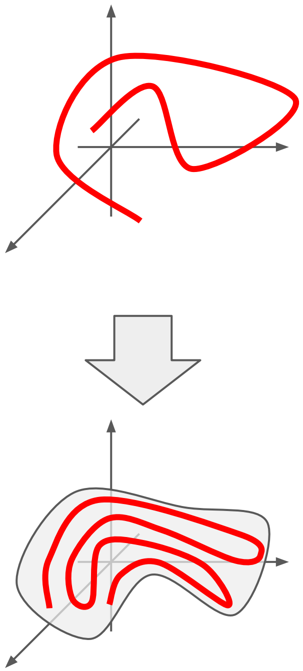


Task





1D manifold



2D manifold

How to find a good F?

Source Samples

$A = \{a_i\}$

F

Translated Source Samples

$F(A) = \{F(a_i) : a_i \in A\}$

minimize statistical distance d(F(A), B)

Target Samples

$B = \{b_j\}$

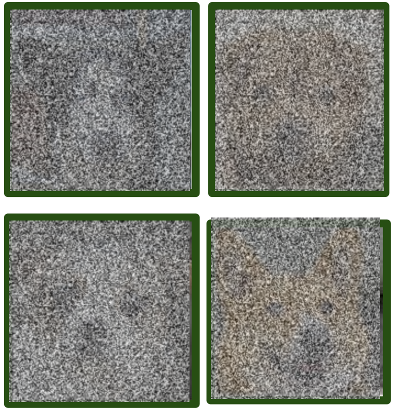
$$\min_{F \in \mathcal{F}} d(F(A), B) + R(F)$$

How to find a good F? - what we expect

Source Samples

Translated Source Samples

$F_{t=0}$



HIGH statistical distance d(F(A), B)

Target Samples

… optimizing F …

$$\min_{F \in \mathcal{F}} d(F(A), B) + R(F)$$

$F_{t=T}$

LOW statistical distance d(F(A), B)

Why care about this problem?

This is an **unsupervised generative** problem **that has GT outputs**!

As a result, we are learning a **neural data model,** but can reason about its **correctness** and the **prediction error vs GT outputs** (e.g. L2).

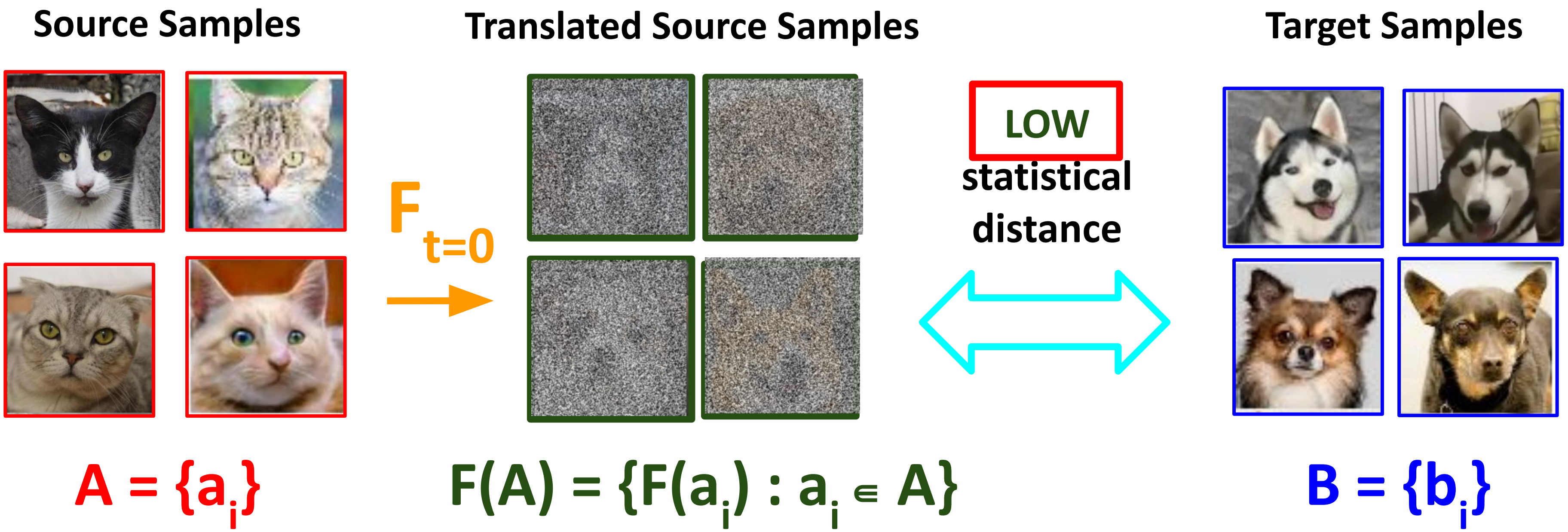
In contrast,
- in GANs - there are **no expected outputs**
- in classification/regression - often no need to **model data**.

Presentation Plan

1. Problem and motivation
2. Existing solutions and possible failure modes
3. Tools for analysing these failure modes in prior work
   a. **“Generalization and Equilibrium in Generative Adversarial Nets”** by Arora et al., PMLR 2017.
   b. **“Training Generative Adversarial Networks with Limited Data”** by Karras et al., NeurIPS 2020.
   c. **“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”** by Galanti et al., JMLR 2021. // 2017-2021

What could go wrong?

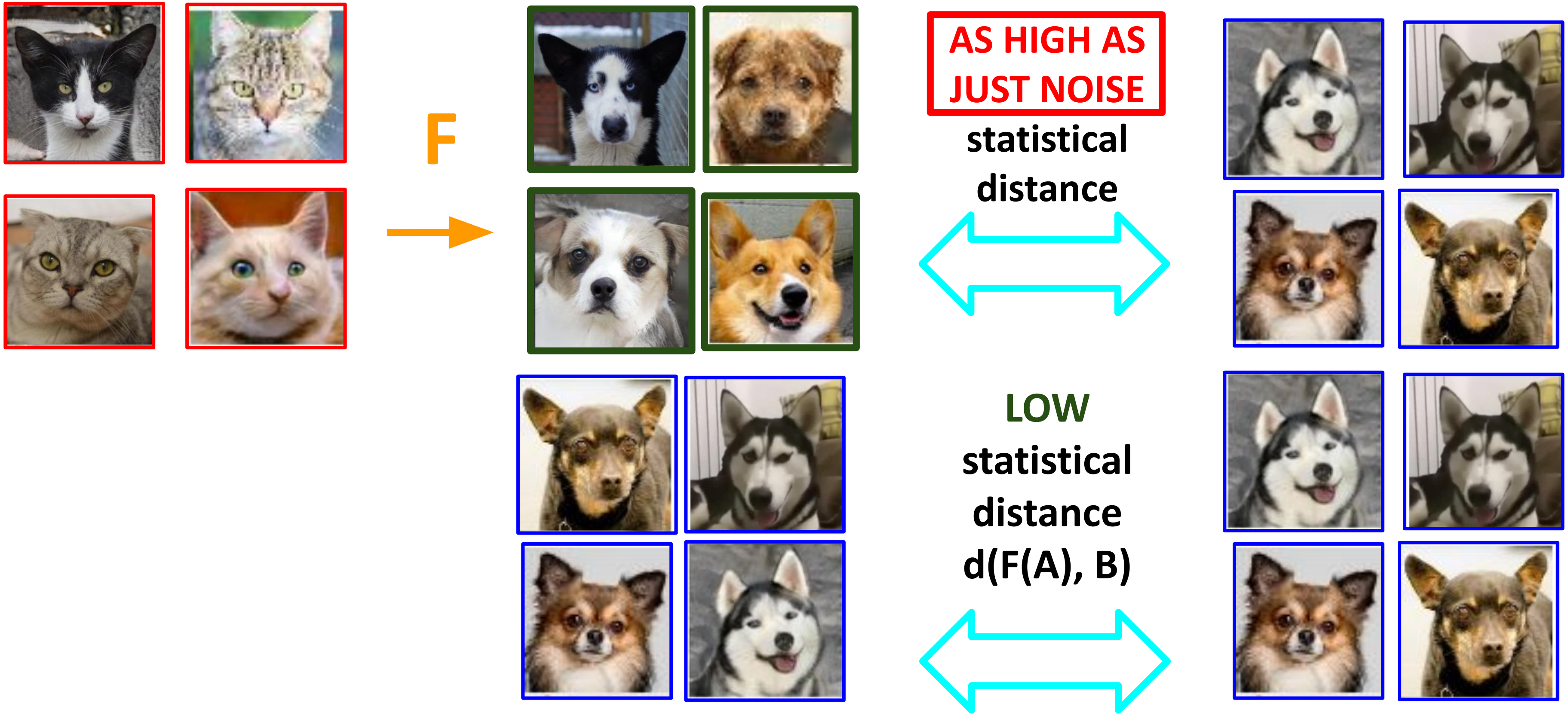
I: The statistical distance is <u>too weak</u>



$$\min_{F \in \mathcal{F}} d(F(A), B) + R(F)$$

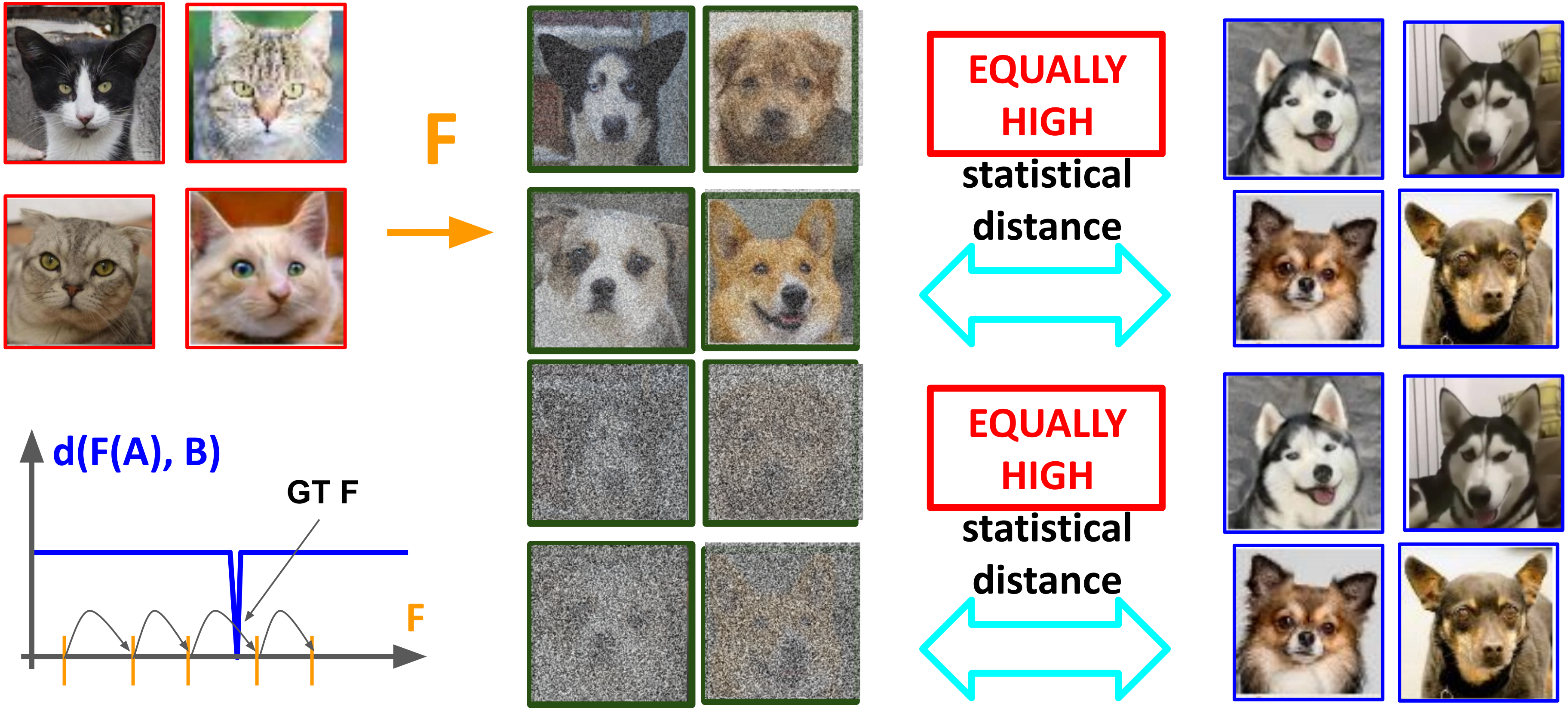
What could go wrong?

I: The statistical distance is too strong



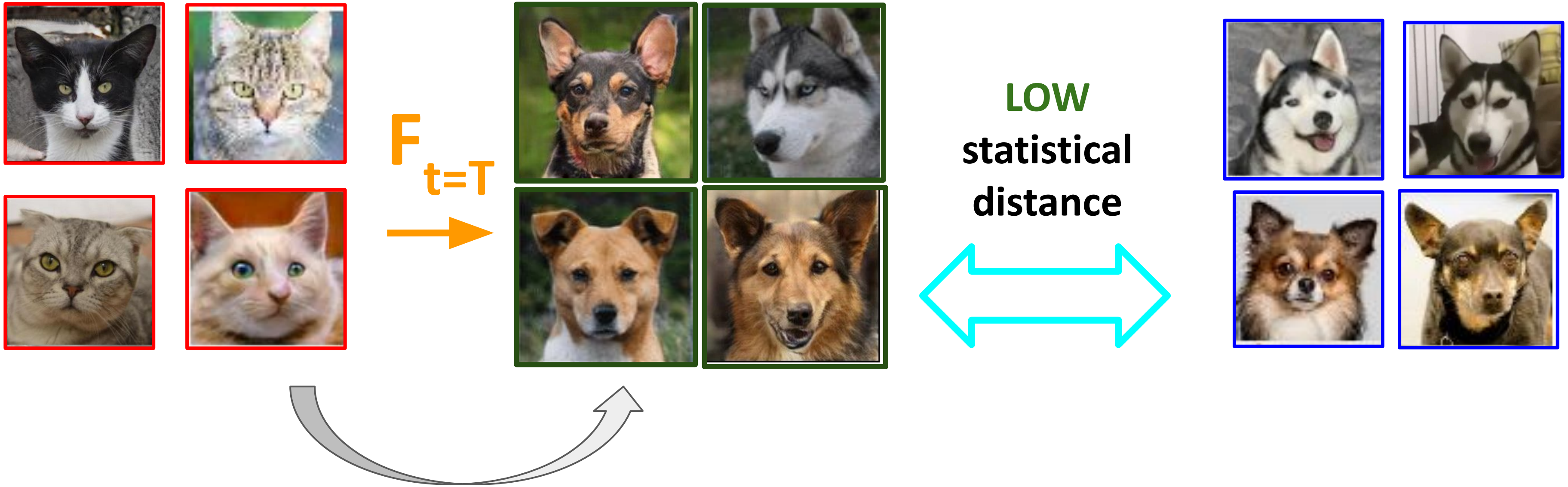
What could go wrong?

II: The stat distance is <u>too sharp (hard to optimize)</u>



What could go wrong?

III: The final mapping is <u>nonsensical</u>



NO SEMANTIC CORRESPONDENCE

$$\min_{F \in \mathcal{F}} d(F(A), B) + R(F)$$

Selected prior work

1. **“Generalization and Equilibrium in Generative Adversarial Nets”** by Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, Yi Zhang, Proceedings International Conference on Machine Learning (PMLR) 2017.

2. **“Training Generative Adversarial Networks with Limited Data”** by Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, Timo Aila; Advances in Neural Information Processing Systems (NeurIPS) 2020.

3. **“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”** by Tomer Galanti, Sagie Benaim, Lior Wolf; JMLR 2021. *// paper series (2017-2021)*

Why these papers?

Introduce important theoretical tools to understand related problems:

- **Chernoff bound and ε-cover method** to estimate sample complexity of adversarial statistical distances
- ***ε*-approximate Nash equilibrium** to analyse the existence of the solution to the adversarial alignment problem
- **Markov operators and group structure of augmentations** to estimate statistical distances between distributions under data augmentations
- **unsupervised bias-variance tradeoff and Rademacher complexity** to relate the prediction error with the alignment error and the complexity of the function class

Other background papers

I also use there papers / books to provide context / refer for proofs

- “Simple Strategies for Large Zero-Sum Games with Applications to Complexity Theory” by Lipton & Young, STOC’94
- “Foundations of Machine Learning” Mohri, Rostamizadeh, Talwalkar, 2nd Edition, 2018

- “Towards Principled Methods for Training Generative Adversarial Networks”, Arjovsky & Bottou, ICLR’17
- “Stabilizing Training of Generative Adversarial Networks through Regularization”, Roth et al, NeurIPS’17
- “Which Training Methods for GANs do actually Converge?”, Mescheder et al., ICML’18
- “Kernel of CycleGAN as a Principle homogeneous space”, Moriakov et al., ICLR’20
- “Guiding the One-to-One Mapping in CycleGAN via Optimal Transport’, Lu et al., AAAI’19

Other background papers

I have “backup slides” covering these papers as well in the end:

- "Table for estimating the goodness of fit of empirical distributions", Smirnov, Annals of Mathematical Statistics ’48 - introduces KS-test
- “A Kernel Two-Sample Test”, Gretton et al, JMLR’12 - introduces MMD test
- “Permutation tests for equality of distributions in high-dimensional settings”, Hall & Tajvidi, Biometrika’02; “Multivariate Generalizations of the Wald-Wolfowitz and Smirnov Two-Sample Tests”, Friedman & Rafsky, Ann Stat 79 - multivariate extensions of non-parametric tests
- “Wasserstein GAN” Arjovsky et al, ICML’17 - introduces WGAN objective
- “Are GANs Created Equal? A Large-Scale Study”, Lucic et al., NeurIPS’18; “Pros and Cons of GAN Evaluation Measures”, Ali Borji, arxiv’18; “Improved Precision and Recall Metric for Assessing Generative Models”, Kynkäänniemi et al., NeurIPS’19 - introduces FID, KID, IS, GAN-F1 score and compares them

Other background papers

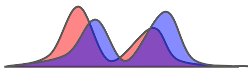
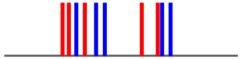
I have “backup slides” covering these papers as well in the end:

- “On the Decreasing Power of Kernel and Distance based Nonparametric Hypothesis Tests in High Dimensions”, Ramdas et al., AAAI'15 - shows that with a “fair alternative” MMD test has exponentially low power in higher dims
- “Revisiting Classifier Two-Sample Tests”, Lopez-Paz et al., ICLR’17 - compares the test power of the GAN-like objective to MMD/KS/other test
- “Reducing Noise in GAN Training with Variance Reduced Extragradient”, Chavdarova et al., NeuIPS’19

How to choose the statistical distance?

Def 1: the statistical distance *d(A, B)* "generalizes"

$$\left| d(\mathcal{D}_{real}, \mathcal{D}_G) - d(\hat{\mathcal{D}}_{real}, \hat{\mathcal{D}}_G) \right| \leq \varepsilon$$

(with exponentially high probability over the choice of m samples as the number of samples increases)

Lem 1: JSD and Wasserstein distances "do not generalize"!

$$\mathcal{N}(0, \tfrac{1}{d} I) = \mu \quad \Rightarrow \quad d_{JS}(\mu, \hat{\mu}) = \log 2, \qquad d_W(\mu, \hat{\mu}) \geq 1.1$$

$$\mathrm{JS}(p;q) = \frac{1}{2} \int \left(p \log \frac{2p}{p+q} + q \log \frac{2q}{p+q} \right) \mathrm{d}\mu$$

(for q = 0 almost everywhere)

$$\Pr[\forall i \in [m] \|y - x_i\| \geq 1.2] \geq 1 - m \exp(-\Omega(d)) \geq 1 - o(1)$$

$$d_W(\mu, \hat{\mu}) \geq 1.2 \Pr[\forall i \in [m] \|y - x_i\| \geq 1.2] \geq 1.1$$

(prob of all pairs of points in A and B being at least 1.2 away from each other does not decay fast enough with m)

[**"Generalization and Equilibrium in Generative Adversarial Nets"** by Arora et al., PMLR 2017]

How to choose the statistical distance?

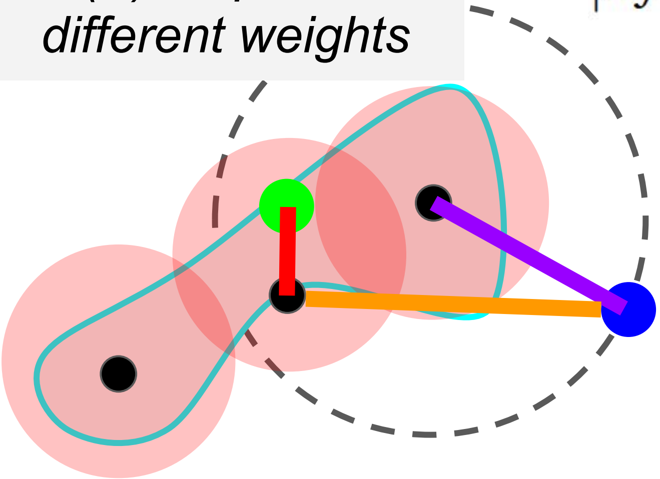
Def 2: *F*-divergence wrt ϕ

$$d_{\mathcal{F},\phi}(\mu,\nu) = \sup_{D\in\mathcal{F}} \mathop{\mathbb{E}}_{x\sim\mu}[\phi(D(x))] + \mathop{\mathbb{E}}_{x\sim\nu}[\phi(1-D(x))] - 2\phi(1/2)$$

Lem 2: $m \geq \frac{cp\Delta^2 \log(LL_\phi p/\epsilon)}{\epsilon^2}$, *we have with probability at least* $1 - \exp(-p)$

$$|d_{\mathcal{F},\phi}(\hat{\mu},\hat{\nu}) - d_{\mathcal{F},\phi}(\mu,\nu)| \leq \epsilon$$

D(X) outputs for different weights



Proof:

a) all **D weights** can be approx. with err. < **e** using a "covert set" of size ~ K = [log(1/e) / e]

=> worst cast D approx. error < **e * L**

b) for any "single fixed" D the est. error < **A**

[**"Generalization and Equilibrium in Generative Adversarial Nets"** by Arora et al., PMLR 2017]

How to choose the statistical distance?

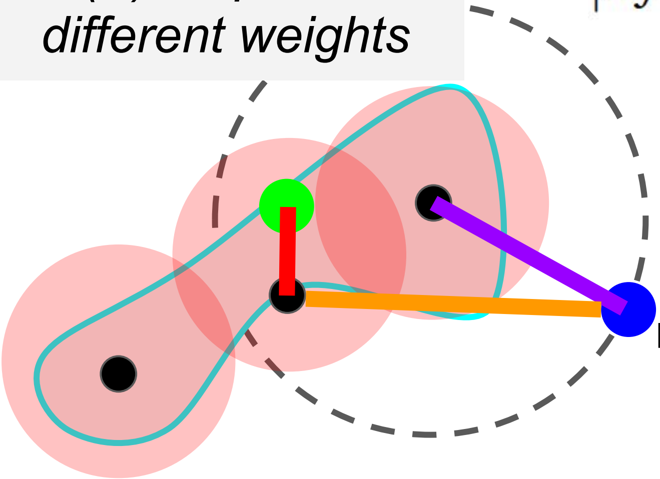
Def 2: *F*-divergence wrt ϕ

$$d_{\mathcal{F},\phi}(\mu,\nu) = \sup_{D\in\mathcal{F}} \mathop{\mathbb{E}}_{x\sim\mu}[\phi(D(x))] + \mathop{\mathbb{E}}_{x\sim\nu}[\phi(1-D(x))] - 2\phi(1/2)$$

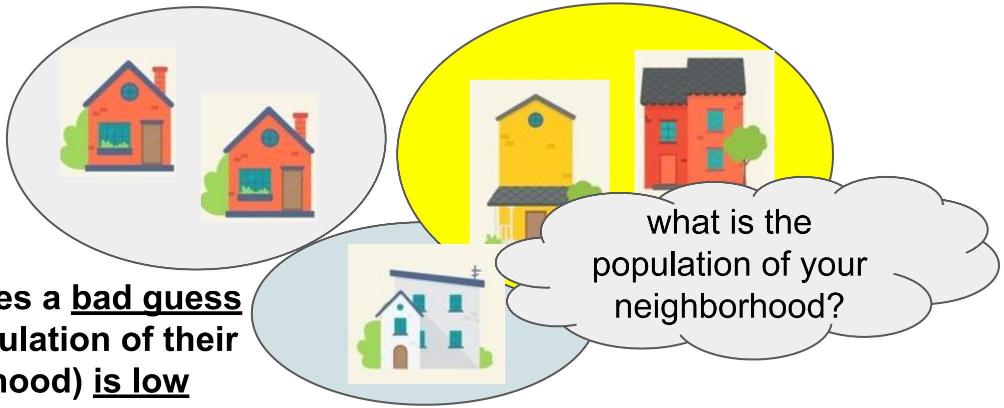
Lem 2: $m \geq \frac{cp\Delta^2 \log(LL_\phi p/\epsilon)}{\epsilon^2}$, *we have with probability at least* $1-\exp(-p)$

$$|d_{\mathcal{F},\phi}(\hat{\mu},\hat{\nu}) - d_{\mathcal{F},\phi}(\mu,\nu)| \leq \epsilon$$

D(X) outputs for different weights



P(person makes a <u>bad guess</u> about the population of their neighbourhood) <u>is low</u>



[“Generalization and Equilibrium in Generative Adversarial Nets” by Arora et al., PMLR 2017]

How to choose the statistical distance?

Def 2: *F*-divergence wrt ϕ

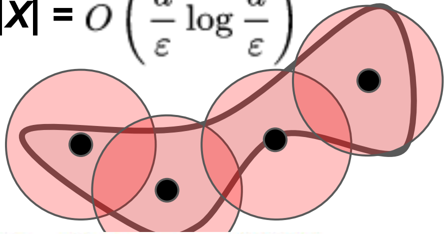
$$d_{\mathcal{F},\phi}(\mu,\nu) = \sup_{D\in\mathcal{F}} \mathop{\mathbb{E}}_{x\sim\mu}[\phi(D(x))] + \mathop{\mathbb{E}}_{x\sim\nu}[\phi(1-D(x))] - 2\phi(1/2)$$

Lem 2: $m \geq \frac{cp\Delta^2\log(LL_\phi p/\epsilon)}{\epsilon^2}$, *we have with probability at least* $1 - \exp(-p)$

$$|d_{\mathcal{F},\phi}(\hat{\mu},\hat{\nu}) - d_{\mathcal{F},\phi}(\mu,\nu)| \leq \epsilon$$

ε-net method on discr weights

$|X| = O\left(\frac{d}{\varepsilon}\log\frac{d}{\varepsilon}\right)$



within distance $\epsilon/8LL_\phi$

$\log|\mathcal{X}| \leq O(p\log(LL_\phi p/\epsilon))$.

$\Pr[x \geq E[x] + t] \leq e^{-2t^2/n}$ - Chernoff bound for $x_i \sim [0,1]$

applied to a single discriminator from the ε-net (bounded outputs!)

$$\Pr[|\mathop{\mathbb{E}}_{x\sim\mu}[\phi(D_v(x))] - \mathop{\mathbb{E}}_{x\sim\hat{\mu}}[\phi(D_v(x))]| \geq \frac{\epsilon}{4}] \leq 2\exp(-\frac{\epsilon^2 m}{2\Delta^2})$$

* |X|

$\Rightarrow$ (+ union bound) fo $m \geq \frac{Cp\Delta^2\log(LL_\phi p/\epsilon)}{\epsilon^2}$ the error is <ε/4 and

$$\|v - v'\| \leq \epsilon/8LL_\phi \Rightarrow |\mathop{\mathbb{E}}_{x\sim\hat{\mu}}[\phi(D_{v'}(x))] - \mathop{\mathbb{E}}_{x\sim\hat{\mu}}[\phi(D_v(x))]| \leq \epsilon/8$$

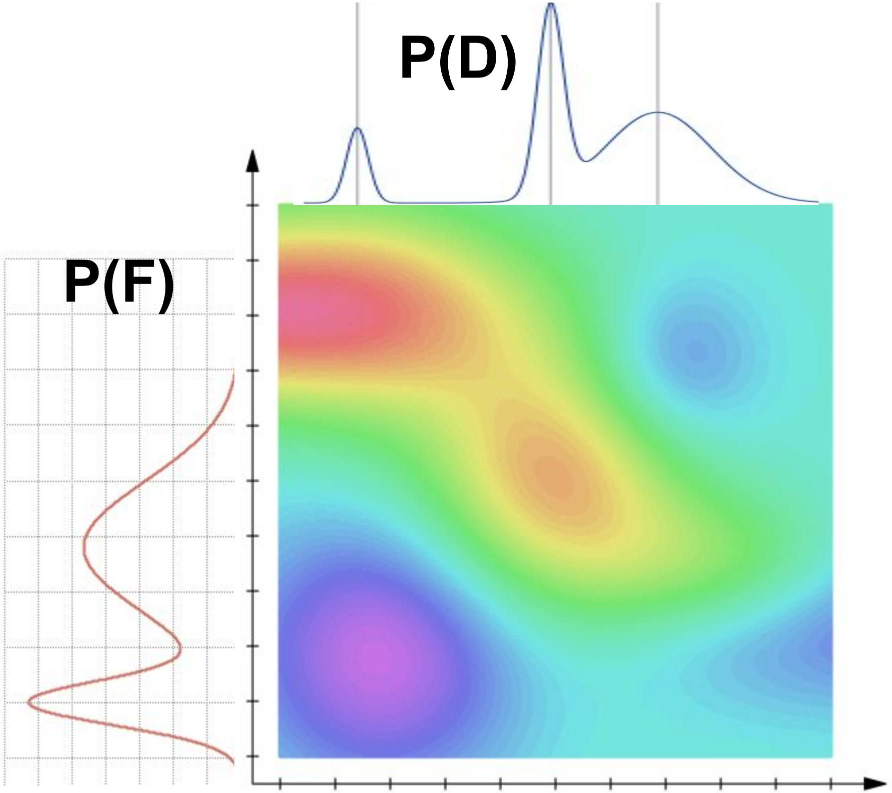
[**“Generalization and Equilibrium in Generative Adversarial Nets”** by Arora et al., PMLR 2017]

Does the minimum exist?

$$\min_F \underbrace{\max_D D(X) - D(F(Y))}_{\min h(F)}$$

$$\begin{array}{ccc} & D=D_A \quad D=D_B & \\ \begin{array}{c} F=F_A \\ F=F_B \end{array} & \begin{bmatrix} 2 & -2 \\ -1 & 1 \end{bmatrix} & \begin{bmatrix} .3 \\ .7 \end{bmatrix} \end{array}$$

The "pure" equilibrium might not always exist, but a **mixed strategy** that yields an equilibrium **always exist**! [Nash '50; Glicksberg '52]



Def 3: ϵ-approximate equilibrium

$$\forall v \in \mathcal{V}, \quad \mathop{\mathbb{E}}_{u \sim \mathcal{S}_u}[F(u,v)] \leq V + \epsilon;$$

$$\forall u \in \mathcal{U}, \quad \mathop{\mathbb{E}}_{v \sim \mathcal{S}_v}[F(u,v)] \geq V - \epsilon.$$

Th: if p-parameter (k-1)-layer network can generate/discriminate each sample => ∃ k-layer G and D with A parameters that are in ϵ-eq

Proof:

1) an infinite mixture of $G_i(z) = x_i$, x_i ~ P(X) is mixed Nash eq.

2) K-sized epsilon-net over samples x_i and params of D gives small error => “subsampled” G’ and D’ are in ϵ/2-eq

3) can approximate “subsampled” G’(x) with a neural G’’(x) that “mixes” outputs of G_i with weights produced by a neural ϵ/2-approx. “1-vs-K indicator” h(z), i.e. $G''(x) = \sum_i G_i(z) * h_i(z)$

[**“Generalization and Equilibrium in Generative Adversarial Nets”** by Arora et al PMLR 2017]
[**“Simple Strategies for Large Zero-Sum Games with Applications to Complexity Theory”** by Lipton & Young, STOC’94]

Proof:

1) an infinite mixture of $G_i(z) = x_i$ is Nash equilibrium
2) ε/$4LL'L_\phi$-net (of size T) over params of G and D gives (with high prob) error <ε/2 => "subsampled" G' and D' are in ϵ-eq
3) a 2-layer network h(z) can δ-approx. a "multi-way step fn"
4) we build new G that "mixes" outputs of G'_i with weights produced by h(z), it is ε/2-away from "true mixture of G_i's"

$$\begin{aligned} F^\star(G, D') &\geq \mathop{\mathbb{E}}_{i\in[T], v\in D'} F(u_i, v) \\ &\quad - \left|F^\star(G, D') - \mathop{\mathbb{E}}_{i\in[T], v\in D'} F(u_i, v)\right| \\ &\geq V - \epsilon/2 - 2\Delta\frac{\epsilon}{4\Delta} \\ &\geq V - \epsilon. \end{aligned}$$

$$\begin{aligned} F^\star(G', D) &\leq \mathop{\mathbb{E}}_{i\in[T], u\in G'} F(u, v_i) \\ &\quad + \left|F^\star(G', D) - \mathop{\mathbb{E}}_{i\in[T], u\in G'} F(u, v_i)\right| \\ &\leq V + \epsilon/2 + 2\Delta\frac{\epsilon}{4\Delta} \\ &\leq V + \epsilon. \end{aligned}$$

second half of the proof is based on
[**"Simple Strategies for Large Zero-Sum Games with Applications to Complexity Theory"** by Lipton & Young, STOC'94]

Takeaway

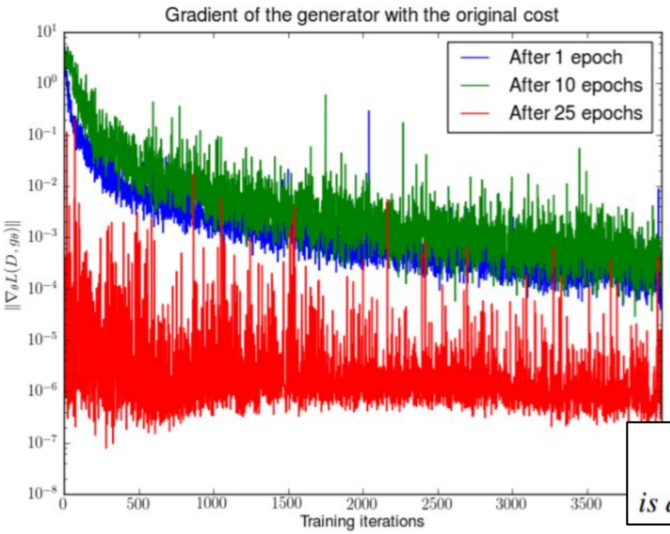
1. The “sample GAN loss” reasonably quickly converges to its “true” value.
2. Jensen-Shannon and Wasserstein distances **do not**.
3. For large networks ϵ-approximate equilibriums **exists**.

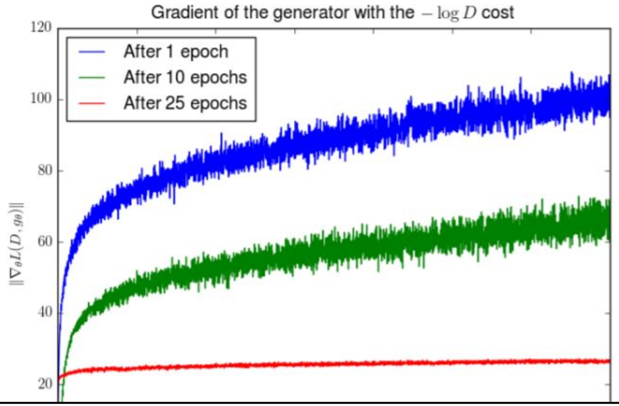
Comments

1. No point in approximating JSD, Wasserstein (and MMD) precisely because their sample estimates are too far from actual values!
2. No point in using them for evaluation either (in higher dimensions)!
3. We are still optimizing for “exact” not “ϵ-approximate” equilibriums.
4. Those equilibriums might also be very hard to get into!

[**“Generalization and Equilibrium in Generative Adversarial Nets”** by Arora et al., PMLR 2017]

GAN-loss is pretty bad optimization-wise





$$\mathbb{E}_{z\sim p(z)}\left[-\nabla_\theta \log D(g_\theta(z))\right]$$

is a centered Cauchy distribution with ***infinite expectation and variance.***[4]

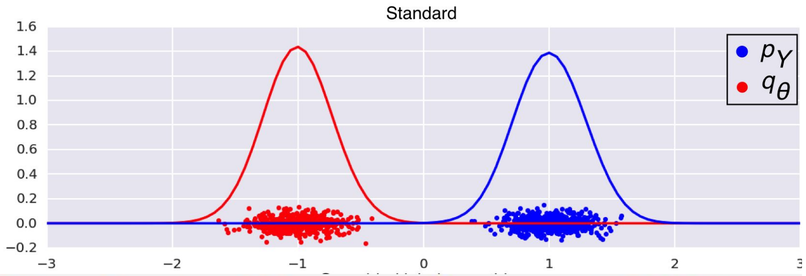
(with generator fixed after X epochs)

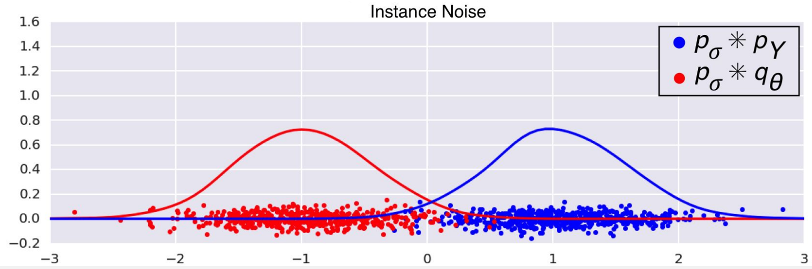
But noise might help:

$$W(\mathbb{P}_r, \mathbb{P}_g) \leq 2V^{\frac{1}{2}} + 2C\sqrt{JSD(\mathbb{P}_{r+\epsilon}\|\mathbb{P}_{g+\epsilon})}$$

[**“Towards Principled Methods for Training Generative Adversarial Networks”**, Arjovsky & Bottou, ICLR’17]

Instance noise in the discriminator might help.
Closed-form regularizer exist.





Regularized Jensen-Shannon GAN

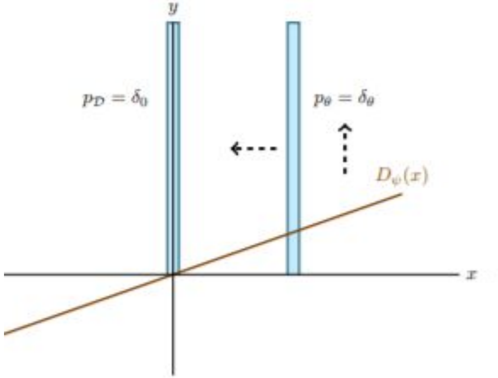
$$F_\gamma(\mathbb{P},\mathbb{Q};\varphi) \;=\; \mathbf{E}_{\mathbb{P}}\left[\ln(\varphi)\right] + \mathbf{E}_{\mathbb{Q}}\left[\ln(1-\varphi)\right] - \frac{\gamma}{2}\Omega_{JS}(\mathbb{P},\mathbb{Q};\varphi)$$

$$\Omega_{JS}(\mathbb{P},\mathbb{Q};\varphi) := \mathbf{E}_{\mathbb{P}}\left[(1-\varphi(\mathbf{x}))^2\|\nabla\phi(\mathbf{x})\|^2\right] + \mathbf{E}_{\mathbb{Q}}\left[\varphi(\mathbf{x})^2\|\nabla\phi(\mathbf{x})\|^2\right]$$

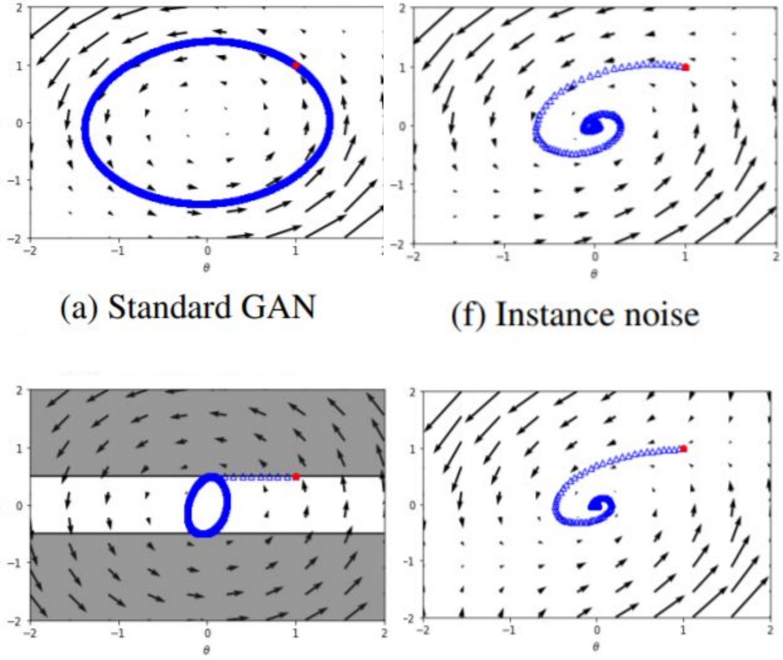
but requires figuring out a good annealing schedule

[**“Stabilizing Training of Generative Adversarial Networks through Regularization”,** Roth et al, NeurIPS’17]
[**“Instance Noise: A trick for stabilising GAN training”**, Ferenc Huszár, inference.vc]

A “toy GAN problem” confirms it.



$$D_\psi(x) = \psi \cdot x$$
$$p_\theta = \delta_\theta$$
$$p_{\mathcal{D}} = \delta_0$$



(a) Standard GAN

(f) Instance noise

(c) WGAN ($n_d = 5$)

(g) Gradient penalty

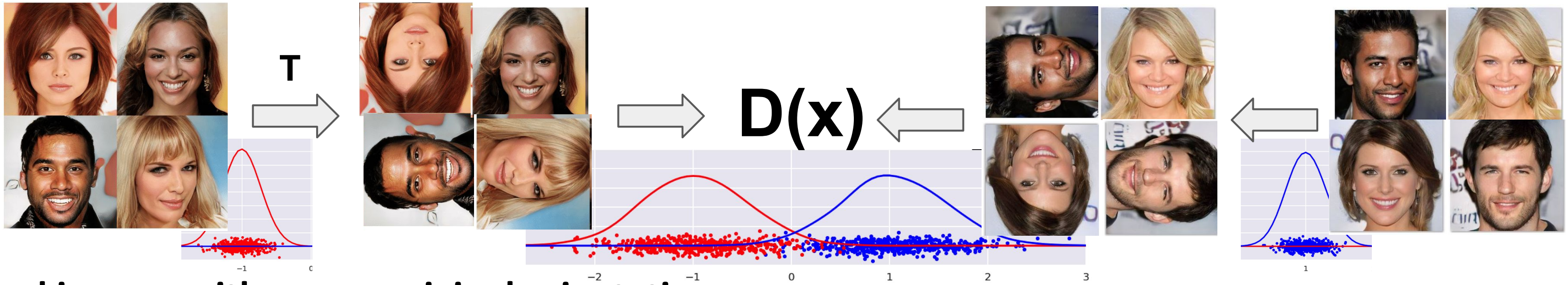
[**“Which Training Methods for GANs do actually Converge?”**, Mescheder et al., ICML’18]

Let’s extend to arbitrary augmentations.

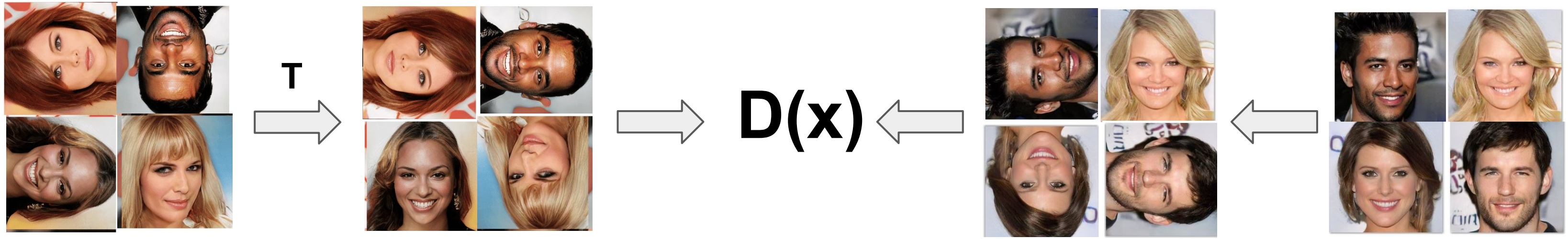
Assume augmentation T(x) randomly flips an image by [0, 90, 180, 270] and we apply T(x) “as instance noise” before passing them to D(x) to make images “less separable”.

“good” generated images

real images



generated images with wrong original orientation



[**“Training Generative Adversarial Networks with Limited Data”**, Karras et al., NeurIPS’20]

Here is what you get - “leaking augmentation”.

T(x) is flip

T(x) is color shift

[**“Training Generative Adversarial Networks with Limited Data”**, Karras et al., NeurIPS’20]

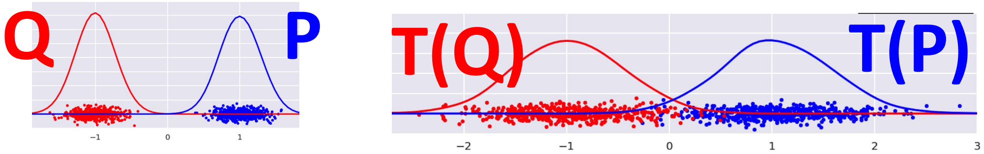
How to avoid “leaking augmentation”?

real images	**T**				**T**	**generated images**

$$P \xrightarrow{\ \mathbf{T}\ } T(P) \rightarrow \mathbf{D(x)} \leftarrow T(Q) \xleftarrow{\ \mathbf{T}\ } Q$$

We want T(x) such that **T(P) = T(Q) ⇔ P = Q ,**
i.e. we want an *invertible* operator “T: distribution □ distribution”.

Not same as an invertible augmentation T(x)!
Example: *T(P) = P ∗ Gaussian(0, 1)*, i.e. *T(x) = x + ε, ε ~ N(0, 1)*.

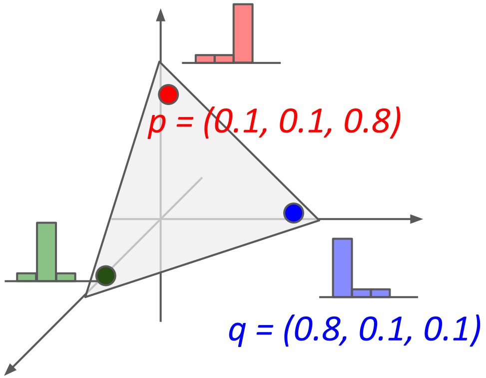


[**“Training Generative Adversarial Networks with Limited Data”**, Karras et al., NeurIPS’20]

Markov operator

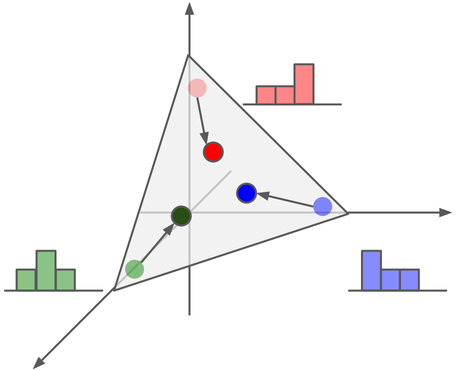
in **observation space** the augmentation function is **random**
e.g. f(1) = {=0 with p=0.2, =1 with p=0.6, and =2 with p=0.2}

X = 1D random variable, supp(X) = {0, 1, 2}
P(X) = a vector in R^3 that lies inside Δ_3



$$T = \begin{pmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{pmatrix}$$

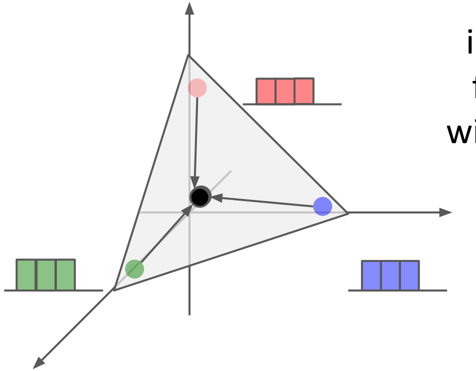
invertible! sums over rows to 1



deterministic linear operator in the distribution space

$$T = \begin{pmatrix} 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \end{pmatrix} * ⅓$$

not invertible! sums over rows to 1



in **observation space**
f(x) is either {0, 1, 2}
with equal probabilities

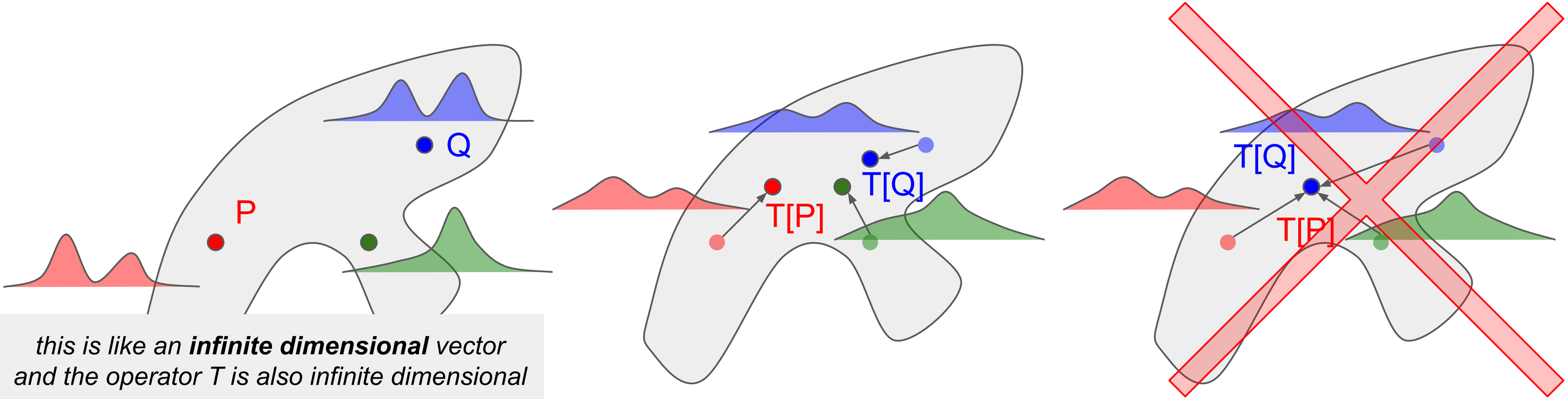
not invertible!

In this case, the
“T: distribution -> distribution”
is just a linear operator “T: Δ_3 -> Δ_3”.

Markov operator

We want T(x) such that **T(P) = T(Q) ⇔ P = Q ,**
i.e. we want an *invertible* operator “T: distribution □ distribution”.

Example: $T(P) = P * N(0, 1)$, i.e. $T(x) = x + \varepsilon$, $\varepsilon \sim N(0, 1)$,
i.e. $T[P](x) = [P * N(0, 1)](x)$, $T^{-1}(T(P)) = P$, $T^{-1}(W) = \mathrm{deconv}(W)$

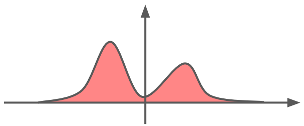


*this is like an **infinite dimensional** vector and the operator T is also infinite dimensional*

How to test invertibility of an infinite-dimensional operator?

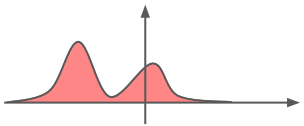
General statements:

1. A **composition** of invertible operators is **invertible** (i.e. a sequence of "good"/"non-leaking" augmentations is still good)
2. A **linear combination** of invertible operators is **not necessarily** invertible [½(T_1 + T_2)](P) means randomly choosing between augmentations T_1 and T_2 and applying it to a single sample from P)

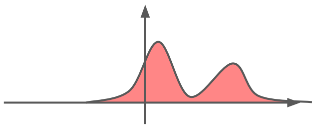


P

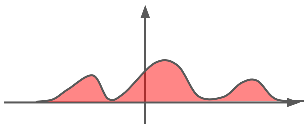
T_1[P](x) = P(x-1)
T_2[P](x) = P(x+1)



T_1(P)



T_2(P)



½(T_1(P) + T_2(P))
= [½(T1 + T2)](P)

[**"Training Generative Adversarial Networks with Limited Data"**, Karras et al., NeurIPS'20]

How to test invertibility of an infinite-dimensional operator?

$$\mathcal{T} = \sum_{i=0}^{N-1} p_i \mathcal{G}^i \qquad \mathcal{U} = \sum_{j=0}^{N-1} q_j \mathcal{G}^j$$

cyclic group $\{\mathcal{G}^i\}_{i=0}^{N-1}$

$\mathcal{G}^0$ is the identity mapping

e.g. {0, 90, 180, 270} img rots

= not just “any” stochastic matrix

$$\begin{aligned}
\mathcal{U}\mathcal{T} &= \left(\sum_{i=0}^{N-1} p_i \mathcal{G}^i\right)\left(\sum_{j=0}^{N-1} q_j \mathcal{G}^j\right) = \sum_{i,j=0}^{N-1} p_i q_j \mathcal{G}^{i+j} \\
&= \sum_{k=0}^{N-1} [p \otimes q]_k \mathcal{G}^k = \sum_{k=0}^{N-1} [\mathbf{F}^{-1}(\hat{p} \odot \hat{q})]_k \mathcal{G}^k \\
&= \sum_{k=0}^{N-1} [\mathbf{F}^{-1}(\hat{p} \odot \hat{p}^{-1})]_k \mathcal{G}^k = \sum_{k=0}^{N-1} [\mathbf{F}^{-1}\mathbf{1}]_k \mathcal{G}^k = \mathcal{G}^0 = \mathcal{I}
\end{aligned}$$

if we set $\hat{q}_i = \frac{1}{\hat{p}_i}$

Inverse operator exists if there are no zeros in operator’s Fourier spectre.

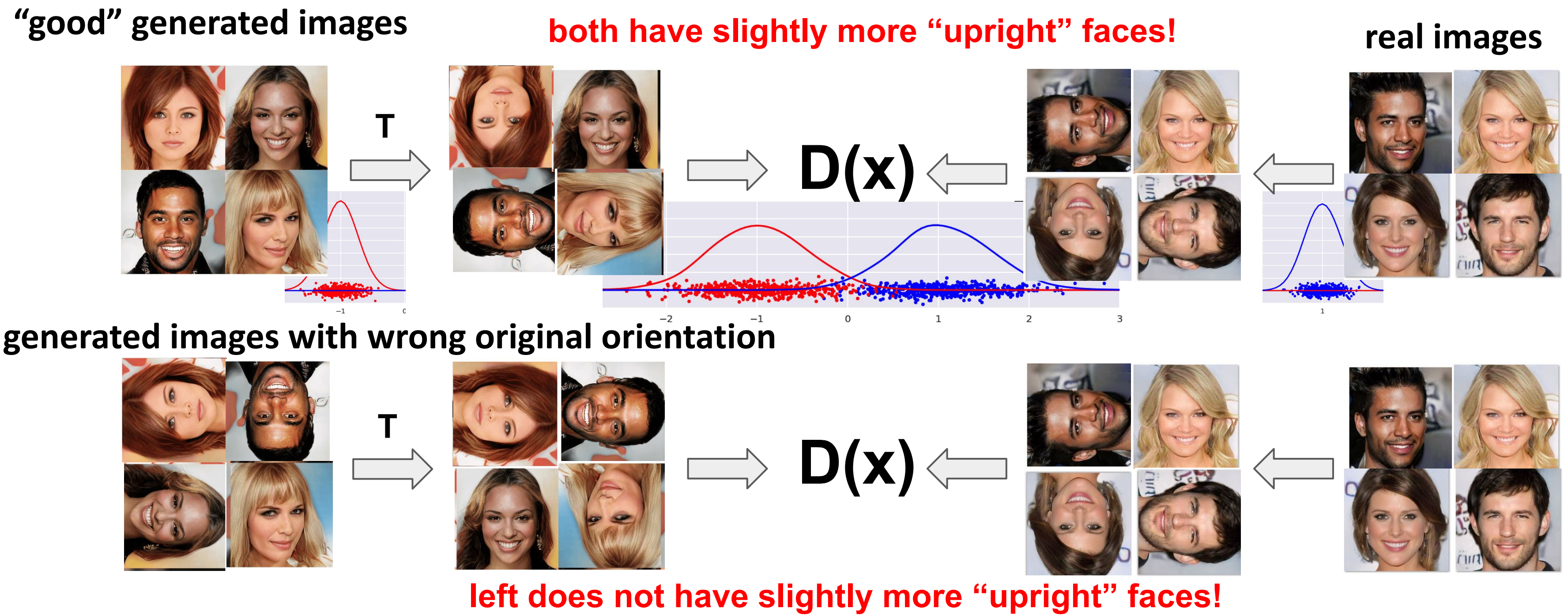
Solution: *uniform* ***but*** *with higher probability of G^0 like [0.28 0.24 0.24 0.24] - has no zeros in spectrum!*

*Essentially T = [(1-α) * Uniform + α * Identity] - e.g. almost like a regularization.*

[**“Training Generative Adversarial Networks with Limited Data”**, Karras et al., NeurIPS’20]

How to test invertibility of an infinite-dimensional operator?

Inverse operator exists if there are no zeros in operator's Fourier spectre.
Solution: *uniform* ***but*** *with higher probability of G^0 like [0.28 0.24 0.24 0.24] - has no zeros in spectrum!*
*Essentially T = [(1-α) * Uniform + α * Identity] - e.g. almost like a regularization.*



["**Training Generative Adversarial Networks with Limited Data**", Karras et al., NeurIPS'20]

How to test invertibility of an infinite-dimensional operator?

Other cases:

1. Non-compact discrete groups (integer shift): also “non-zero Fourier”
2. For continuous groups (e.g. rotations): also “non-zero Fourier” (use Haar measure over that groups under the integral);
3. Additive pixel noise: “non-zero Fourier” of the noise kernel
4. Cropping / blitting / “projection”: requires P(identity) > 0
   Assume $\exists$ **y≠z** s.t. T**y** **=** T**z**, e.g. T(**y-z**) = **0**, e.g. T**x** = **0.**

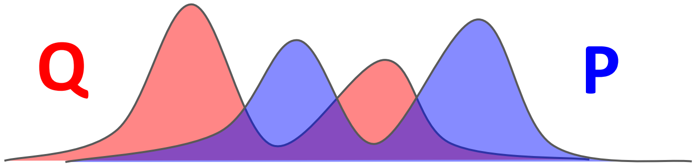
$$\mathcal{T} = p_0\mathcal{I} + \sum_{j=1}^{N} p_j \mathcal{P}_j \qquad 0 = \mathcal{T}\mathbf{x} = p_0\mathbf{x} + \sum_{j=1}^{N} p_j \mathcal{P}_j \mathbf{x} \qquad \sum_{j=1}^{N} p_j \underbrace{\langle \mathbf{x}, \mathcal{P}_j \mathbf{x} \rangle}_{\geq 0} = -p_0 \underbrace{\langle \mathbf{x}, \mathbf{x} \rangle}_{\geq 0}$$

invertible if $p_0 \neq 0$

[**“Training Generative Adversarial Networks with Limited Data”**, Karras et al., NeurIPS’20]

So how do we use it?

Q



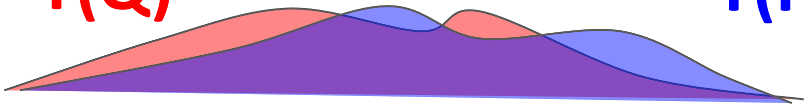
P

$D_{acc} = 0.9$

$T = T_1 \circ T_2 \circ T_3 \ldots$

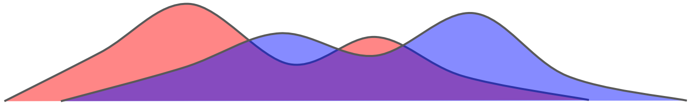
T(Q)

T(P)



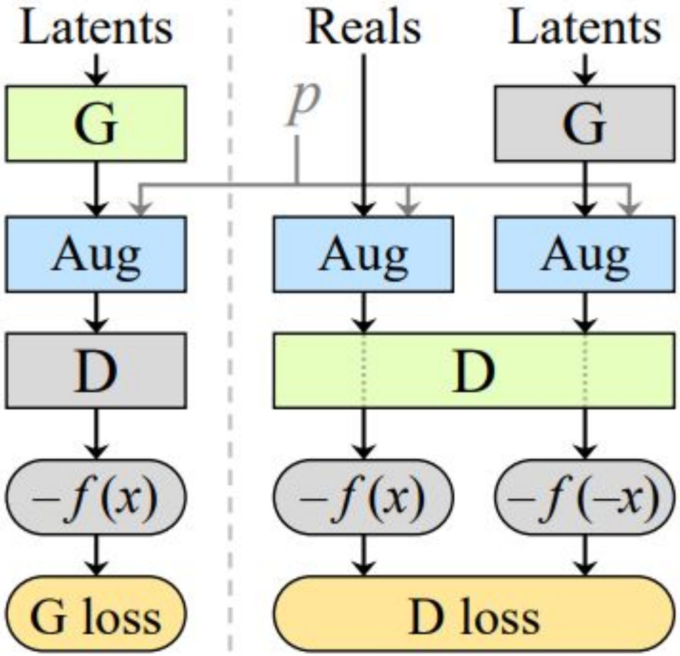
$D_{acc} = 0.55$

(1-p)*Q + p*T(Q)



(1-p)*P + p*T(P)

$D_{acc} = 0.8$



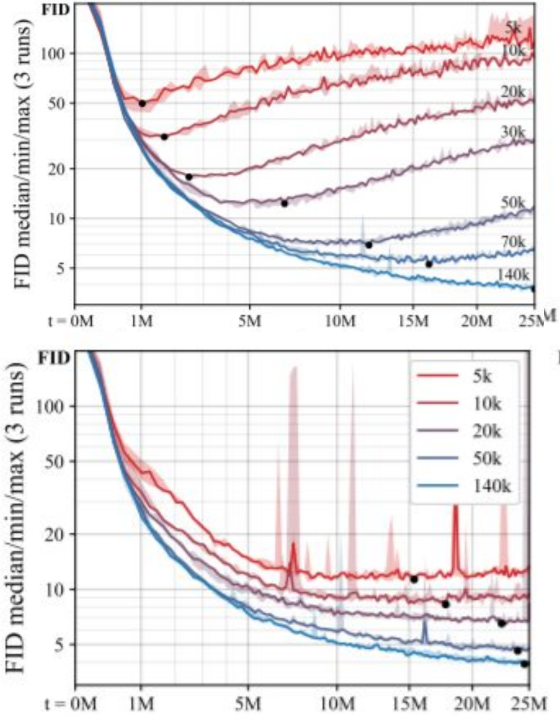
[**"Training Generative Adversarial Networks with Limited Data"**, Karras et al., NeurIPS'20]

Does it help? - yes!

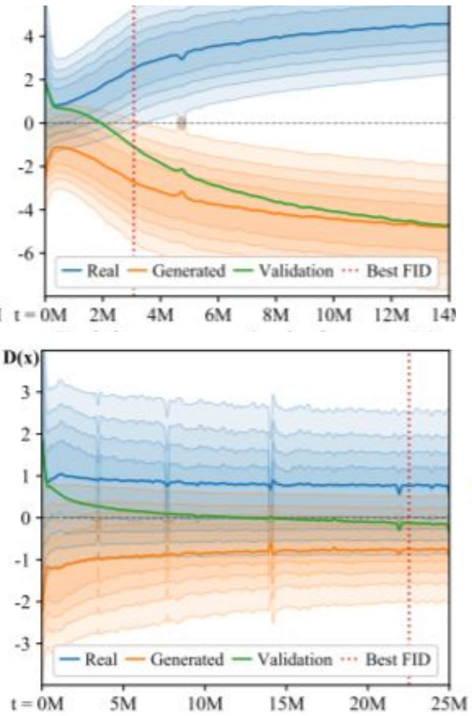
real train / real val / generated
D(x) scores trained on 20k samples

w\o augmentation

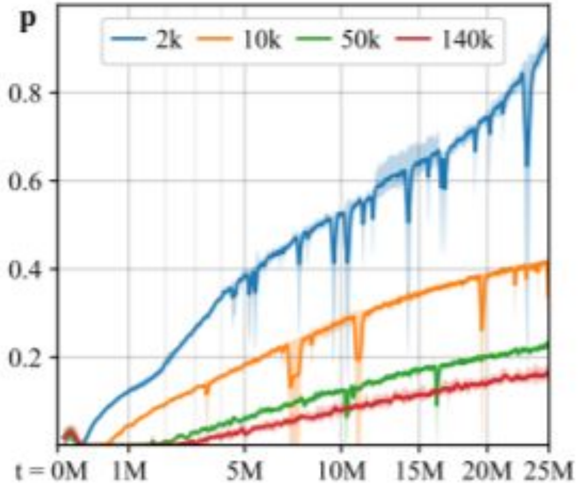
with aug.



FID(train step) for
different amount of real data



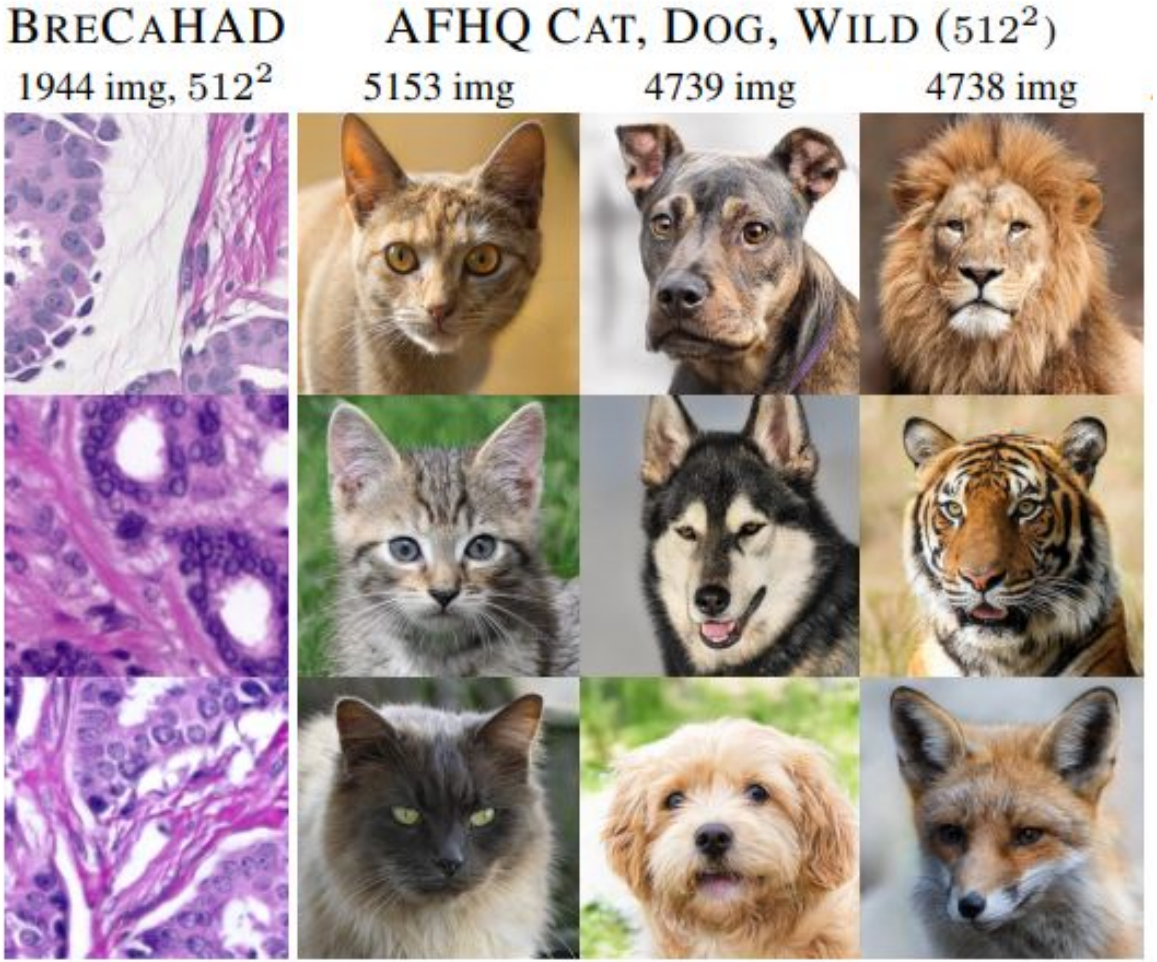
$$r_v = \frac{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{generated}}]}$$



(c) Evolution of p over training

[**"Training Generative Adversarial Networks with Limited Data"**, Karras et al., NeurIPS'20]

Does it help? - yes!



[**"Training Generative Adversarial Networks with Limited Data"**, Karras et al., NeurIPS'20]

Takeaway

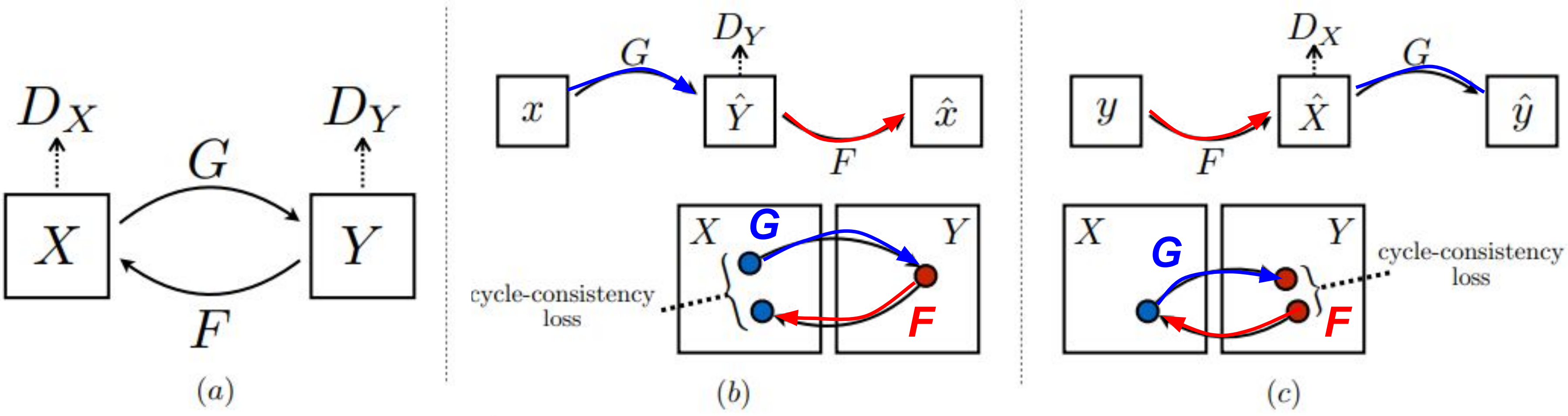
1. Regularizing the discriminator with augmentations helps.
2. But it has to be done in a way that does not “leak” into generated images.
3. For a wide variety of transformations, applying them with a fixed probability “does not leak” into generated examples.

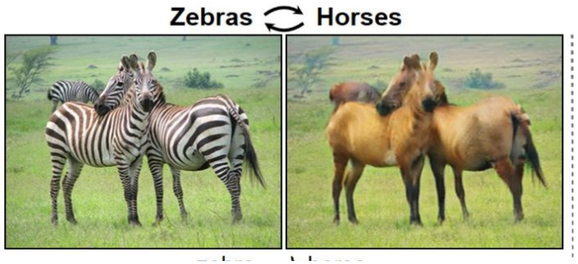
Comments

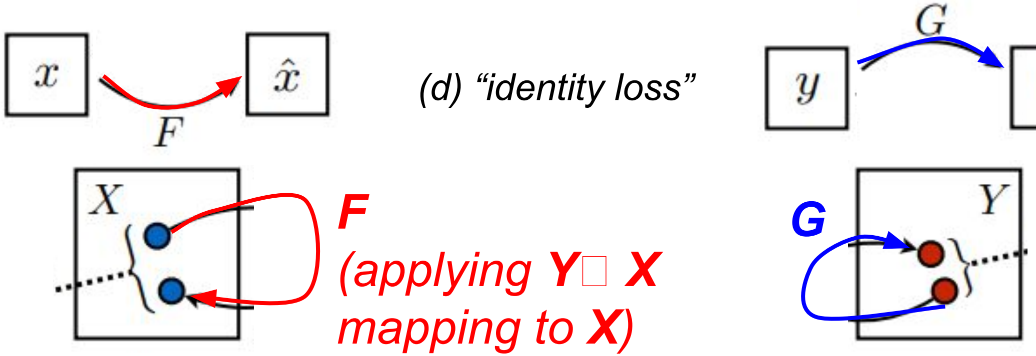
1. All these methods still require careful parameter annealing.
2. As a result we can not reason about the convergence of an objective because there is no single objective! (we change it as we train)

[**“Training Generative Adversarial Networks with Limited Data”**, Karras et al., NeurIPS’20]

CycleGAN overview



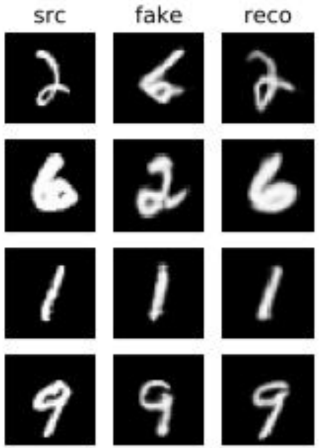




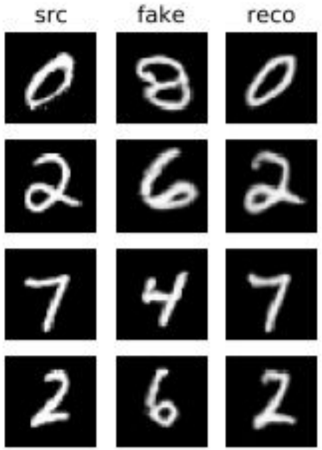
["**Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks**", Zhu et al., ICCV'17]

How to reason about the complexity of the CycleGAN?

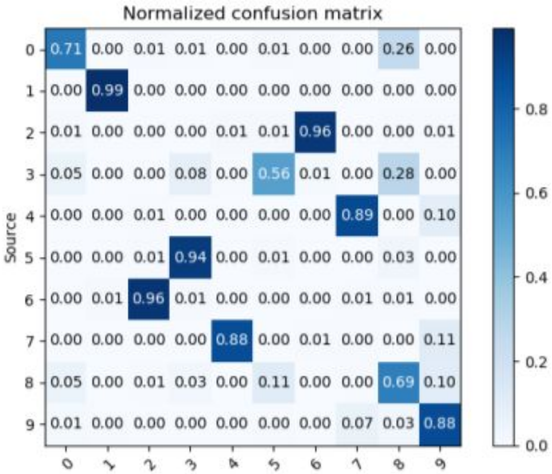
CycleGAN trained to map MNIST train split to the MNIST test split.



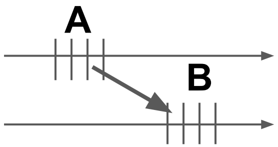
X

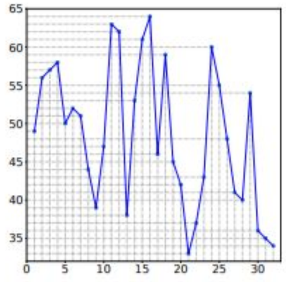


Y



1D-to-1D

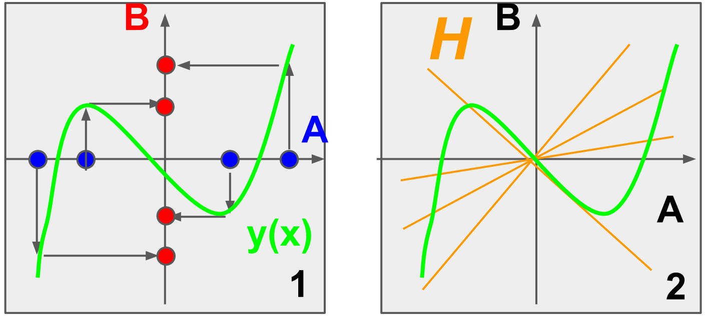


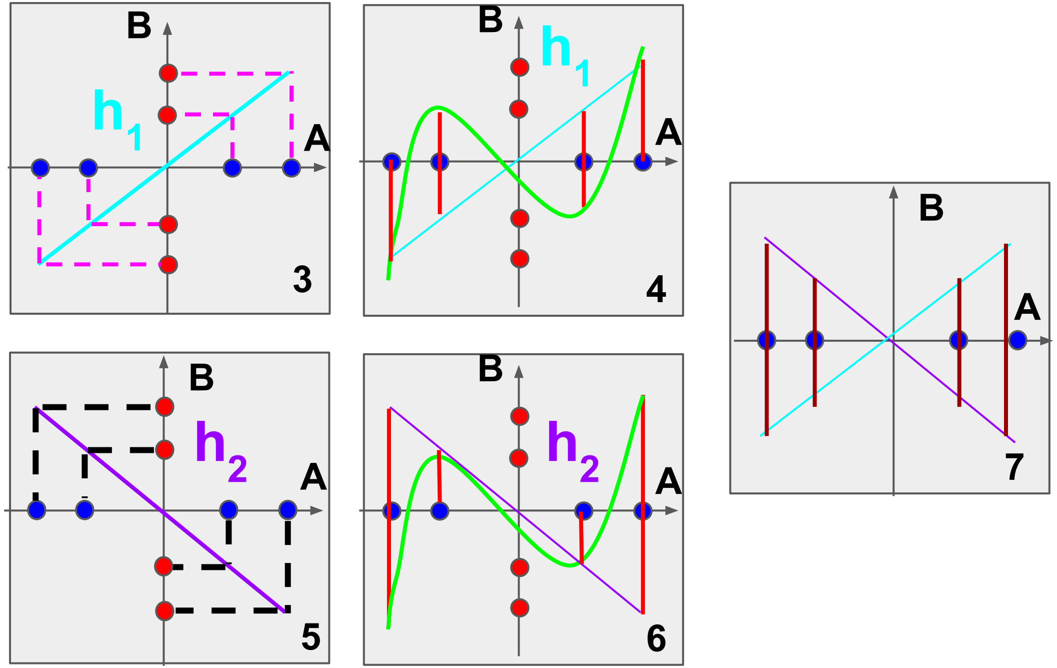


(b) CycleGAN Mapping

[**“Kernel of CycleGAN as a Principle homogeneous space”**, Moriakov et al., ICLR’20]
[**“Guiding the One-to-One Mapping in CycleGAN via Optimal Transport’**, Lu et al., AAAI’19]

How to bound the unsupervised alignment error?





prediction error <
smallest unsupervised **alignment error**
+ smallest **approximation** error in H
+ the **variance** between functions
minimizing the alignment loss.

[**"Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs",** Galanti et al., JMLR'21]
[**"Estimating the Success of Unsupervised Image to Image Translation",** Benaim et al., ECCV'18]
[**"The role of minimal complexity functions in unsupervised learning of semantic mappings",** Galanti et al., ICLR'18]

How to bound the unsupervised alignment error?

$$R_{D_A}[h_1, y] \lesssim \underbrace{\sup_{h_2 \in \mathcal{P}_\omega} R_{\mathcal{S}_A}[h_1, h_2]} + c \underbrace{\inf_{h \in \mathcal{P}_\omega} \rho_C(h \circ \mathcal{S}_A, \mathcal{S}_B)} + \ \ldots$$

R *- pred error;* ***ρ*** *- alignment error;* ***D_A*** *- distribution;* ***S_A*** *- dataset;* ***P_k*** - hypotheses with "low" alignment error

How to use this bound?

A given choice of hyperparameters is evaluated as follows:

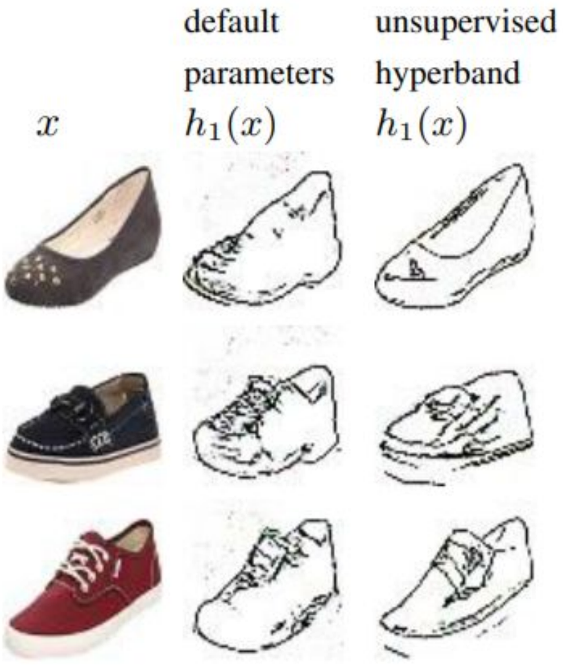
1. $\inf_{h \in \mathcal{P}_k} \rho_C(h \circ \mathcal{S}_A, \mathcal{S}_B)$ **-** we minimize the "GAN loss" to get "the first best" **h1**

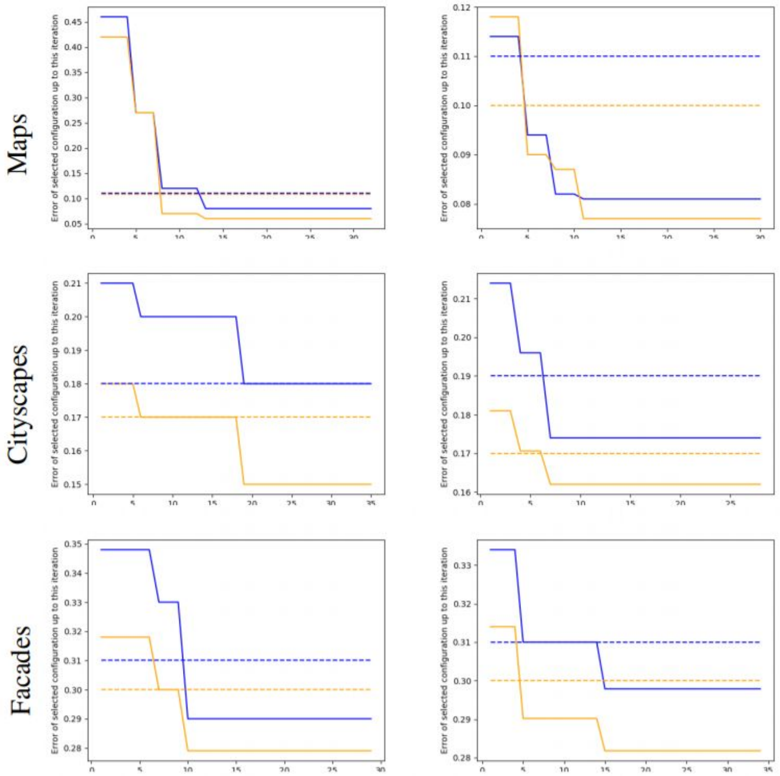
2. $\min_{h_2 \in \mathcal{H}_k} \left\{ \rho_C(h_2 \circ \mathcal{S}_A, \mathcal{S}_B) - \lambda R_{\mathcal{S}_A}[h_1, h_2] \right\}$ **-** then pick the "second best" **h2**

3. $R_{\mathcal{S}_A}[h_1, h_2] + \rho_C(h_1 \circ \mathcal{S}_A, \mathcal{S}_B)$ - and then bound the unknown GT error

[**"Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs",** Galanti et al., JMLR'21]
[**"Estimating the Success of Unsupervised Image to Image Translation",** Benaim et al., ECCV'18]
[**"The role of minimal complexity functions in unsupervised learning of semantic mappings",** Galanti et al., ICLR'18]

Does it help? - yes!





The **ground truth error** and the **theoretical bound** as a function of hyper-parameter optimization steps varying
- encoder and decoder #layers
- batch size
- learning rate
...

[**"Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs",** Galanti et al., JMLR'21]
[**"Estimating the Success of Unsupervised Image to Image Translation",** Benaim et al., ECCV'18]
[**"The role of minimal complexity functions in unsupervised learning of semantic mappings",** Galanti et al., ICLR'18]

translation functions

discriminator class

domains

Theorem 1 (Cross-Domain Mapping with IPMs) *Assume that* $\mathcal{X}_A \subset \mathbb{R}^N$ *and* $\mathcal{X}_B \subset \mathbb{R}^M$ *are convex and bounded sets. Let* $\mathcal{H}$ *be the hypothesis class and* $\mathcal{C}$ *the class of discriminators. Assume that* $\mathcal{C} \subset C^2$ *and* $\sup_{d\in\mathcal{C}} \|d\|_{\infty,\mathcal{X}_A\cup\mathcal{X}_B} < \infty$. *Then, for any* $\delta \in (0,1)$ *and* $c \geq 1$, *with probability at least* $1-\delta$ *over the selection of* $\mathcal{S}_A \sim D_A^{m_1}$ *and* $\mathcal{S}_B \sim D_B^{m_2}$, *for every* $\omega \in \Omega$ *and* $h_1 \in \mathcal{P}_\omega := \mathcal{P}_\omega(\mathcal{S}_A, \mathcal{S}_B)$, *we have:*

single discriminator

finite datasets

hyperparams

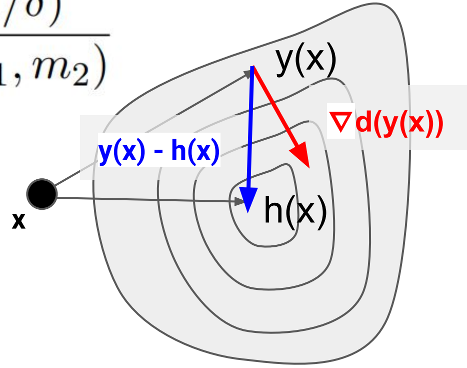
a mapping produced by the learning algorithm

[**“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”,** Galanti et al., JMLR’21]
[**“Estimating the Success of Unsupervised Image to Image Translation”,** Benaim et al., ECCV’18]
[**“The role of minimal complexity functions in unsupervised learning of semantic mappings”,** Galanti et al., ICLR’18]

Theorem 1 (Cross-Domain Mapping with IPMs) *Assume that $\mathcal{X}_A \subset \mathbb{R}^N$ and $\mathcal{X}_B \subset \mathbb{R}^M$ are convex and bounded sets. Let $\mathcal{H}$ be the hypothesis class and $\mathcal{C}$ the class of discriminators. Assume that $\mathcal{C} \subset C^2$ and $\sup_{d \in \mathcal{C}} \|d\|_{\infty, \mathcal{X}_A \cup \mathcal{X}_B} < \infty$. Then, for any $\delta \in (0,1)$ and $c \geq 1$, with probability at least $1 - \delta$ over the selection of $\mathcal{S}_A \sim D_A^{m_1}$ and $\mathcal{S}_B \sim D_B^{m_2}$, for every $\omega \in \Omega$ and $h_1 \in \mathcal{P}_\omega := \mathcal{P}_\omega(\mathcal{S}_A, \mathcal{S}_B)$, we have:*

$$R_{D_A}[h_1, y] \lesssim \sup_{h_2 \in \mathcal{P}_\omega} R_{\mathcal{S}_A}[h_1, h_2] + c \inf_{h \in \mathcal{P}_\omega} \rho_{\mathcal{C}}(h \circ \mathcal{S}_A, \mathcal{S}_B) + \inf_{h \in \mathcal{P}_\omega} \inf_{\substack{d \in \mathcal{C} \\ \beta(d) \leq 1}} \mathcal{K}(h, d; y) \tag{9}$$
$$+ \hat{\mathcal{R}}_{\mathcal{S}_A}(\ell_{\mathcal{H}}) + \hat{\mathcal{R}}_{\mathcal{S}_A}(\mathcal{C} \circ \mathcal{H}) + \hat{\mathcal{R}}_{\mathcal{S}_B}(\mathcal{C}) + \sqrt{\frac{\log(1/\delta)}{\min(m_1, m_2)}}$$

where, $\mathcal{K}(h, d; y) := \mathbb{E}_{x \sim D_A}\left[\|\nabla_{y(x)} d(y(x)) - (h(x) - y(x))\|_2\right]$.



[**"Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs",** Galanti et al., JMLR'21]
[**"Estimating the Success of Unsupervised Image to Image Translation",** Benaim et al., ECCV'18]
[**"The role of minimal complexity functions in unsupervised learning of semantic mappings",** Galanti et al., ICLR'18]

$$\boxed{R_{D_A}[h_1, y]} \lesssim \sup_{h_2 \in \mathcal{P}_\omega} R_{\mathcal{S}_A}[h_1, h_2] + c \inf_{h \in \mathcal{P}_\omega} \rho_{\mathcal{C}}(h \circ \mathcal{S}_A, \mathcal{S}_B) + \inf_{h \in \mathcal{P}_\omega} \inf_{\substack{d \in \mathcal{C} \\ \beta(d) \leq 1}} \mathcal{K}(h, d; y)$$

$$+ \hat{\mathscr{R}}_{\mathcal{S}_A}(\ell_{\mathcal{H}}) + \hat{\mathscr{R}}_{\mathcal{S}_A}(\mathcal{C} \circ \mathcal{H}) + \hat{\mathscr{R}}_{\mathcal{S}_B}(\mathcal{C}) + \sqrt{\frac{\log(1/\delta)}{\min(m_1, m_2)}}$$

Lem 3: $$\boxed{R_{D_A}[h_1, y]} \leq 3 \sup_{h_2 \in \mathcal{P}} \boxed{R_{D_A}[h_1, h_2]} + 3 \inf_{h \in \mathcal{P}} \boxed{R_{D_A}[h, y]}$$

Lem 4: $$\boxed{R_{D_A}[h, y]} \leq \frac{2\rho_C(h \circ D_A, D_B)}{2 - \beta(d)} + \frac{2 \underbrace{\boxed{\sup_{u \in \mathcal{X}_A} \|h(u) - y(u)\|_2}}_{\leq L}}{2 - \beta(d)} \cdot \mathcal{K}(h, d; y)$$

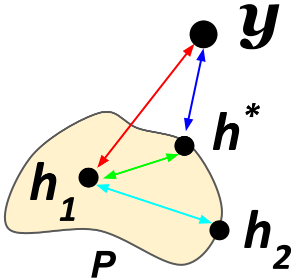
Lem 7: $$\boxed{R_{D_A}[h_1, h_2]} \leq R_{\mathcal{S}_A}[h_1, h_2] + 2\hat{\mathscr{R}}_{\mathcal{S}_A}(\ell_{\mathcal{H}}) + 9K^2 \sqrt{\frac{\log(6/\delta)}{2m_1}}$$

[**“Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs”,** Galanti et al., JMLR’21]

Lem 3 (Triangle inequality and the "set diameter")

$$R_{D_A}[h_1, y] \le 3 \sup_{h_2 \in \mathcal{P}} R_{D_A}[h_1, h_2] + 3 \inf_{h \in \mathcal{P}} R_{D_A}[h, y]$$

$$\boxed{h^* \in \arg\inf_{h \in \mathcal{P}} R_{D_A}[h^*, y]}$$



$$\begin{aligned} \|a-c\|_2^2 &\le (\|a-b\|_2 + \|b-c\|_2)^2 \\ &\le \|a-b\|_2^2 + \|b-c\|_2^2 + 2\max(\|a-b\|_2^2, \|b-c\|_2^2) \\ &\le 3(\|a-b\|_2^2 + \|b-c\|_2^2) \end{aligned}$$

$$\begin{aligned} R_{D_A}[h_1, y] &= \mathbb{E}_{x \sim D_A}[\|h_1(x) - y(x)\|_2^2] \\ &\le \mathbb{E}_{x \sim D_A}\left[3\|h_1(x) - h^*(x)\|_2^2 + 3\|h^*(x) - y(x)\|_2^2\right] \\ &= 3\left[R_{D_A}[h_1, h^*] + \inf_{h \in \mathcal{P}} R_{D_A}[h, y]\right] \end{aligned}$$

$$\boxed{R_{D_A}[h_1, h^*] \le \sup_{h_2 \in \mathcal{P}} R_{D_A}[h_1, h_2]}$$

Lem 4 (Prediction Error via Stat. Distance and Discriminator Capacity)

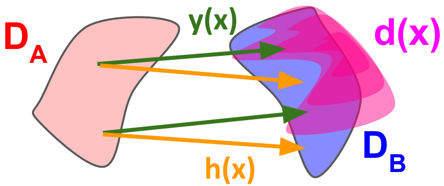
$$\overbrace{R_{D_A}[h,y]}^{\text{avg prediction error}} \leq \frac{\overbrace{2\rho_C(h\circ D_A, D_B)}^{\text{stat. distance}}}{2-\beta(d)} + \frac{\overbrace{2\sup_{u\in X_A}\|h(u)-y(u)\|_2}^{\textit{max}\text{ prediction error} < L}}{2-\beta(d)} \cdot \overbrace{\mathcal{K}(h,d;y)}^{\text{discriminator error}}$$

$$\textit{where, } \mathcal{K}(h,d;y) := \mathbb{E}_{x\sim D_A}\left[\|\nabla_{y(x)} d(y(x)) - (h(x)-y(x))\|_2\right]$$

$$\underbrace{\rho_C(h\circ D_A, D_B)}_{\text{stat. distance between } h(P_A) \text{ and } P_B} = \sup_{d\in C}\left\{\mathbb{E}_{u\sim h\circ D_A}[d(u)] - \mathbb{E}_{v\sim D_B}[d(v)]\right\} = \sup_{d\in C}\left\{\mathbb{E}_{x\sim D_A}[d\circ h(x) - d\circ y(x)]\right\}$$

$f(x) - f(y) = \langle\nabla f(x), x - y\rangle + \dots$
$= \|x - y\|^2 + \langle\nabla f(x) - (x - y), x - y\rangle + \dots$

$$\geq \mathbb{E}_{x\sim D_A}[d(h(x)) - d(y(x))] = \mathbb{E}_{x\sim D_A}\left[\|h(x)-y(x)\|_2^2\right]$$
$$+\, \mathbb{E}_{x\sim D_A}\left[\left\langle \nabla_{y(x)} d(y(x)) - (h(x)-y(x)), h(x)-y(x)\right\rangle\right]$$
$$+\, \frac{1}{2}\mathbb{E}_{x\sim D_A}\left[\left\langle (h(x)-y(x))^\top \cdot \mathrm{H}_d(u^*_{d,x}), h(x)-y(x)\right\rangle\right]$$



$$= \mathbb{E}_{x\sim D_A}\left[\|h(x)-y(x)\|_2^2\right] - \frac{1}{2}\mathbb{E}_{x\sim D_A}\left[\beta(d)\cdot\|h(x)-y(x)\|_2^2\right]$$
$$- \sup_{u\in X_A}\|h(u)-y(u)\|_2 \cdot \underbrace{\mathbb{E}_{x\sim D_A}\left[\|\nabla_{y(x)} d(y(x)) - (h(x)-y(x))\|_2\right]}_{\text{discriminator approx error (capacity)}}$$

$$= \left(1-\frac{\beta(d)}{2}\right)\underbrace{R_{D_A}[h,y]}_{\text{prediction error}} - \underbrace{\sup_{u\in X_A}\|h(u)-y(u)\|_2}_{\text{max pred error} < L} \cdot \mathcal{K}(h,d;y)$$

[**"Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs"**, Galanti et al., JMLR'21]

Definition 3.1 (Empirical Rademacher complexity)

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(z_i)\right]$$

Theorem 3.3 *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z}$ to $[0,1]$.* *with probability at least $1-\delta$*

$$\mathbb{E}[g(z)] \le \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\widehat{\mathfrak{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}$$

$$\begin{aligned}
\mathbb{E}_S[\Phi(S)] &= \mathbb{E}_S\left[\sup_{g\in\mathcal{G}}\left(\mathbb{E}[g] - \widehat{\mathbb{E}}_S(g)\right)\right] = \mathbb{E}_{S,S'}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m}(g(z_i') - g(z_i))\right] \\
&= \mathbb{E}_{\boldsymbol{\sigma},S,S'}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i(g(z_i') - g(z_i))\right] \\
&\le \mathbb{E}_{\boldsymbol{\sigma},S'}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z_i')\right] + \mathbb{E}_{\boldsymbol{\sigma},S}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m}-\sigma_i g(z_i)\right] \\
&= 2\,\mathbb{E}_{\boldsymbol{\sigma},S}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z_i)\right] = 2\mathfrak{R}_m(\mathcal{G}).
\end{aligned}$$

sup g - a function with the **largest** deviation of its value from average across possible splits S, S'

with probability at least $1-\delta$ by McDiarmid's inequality $\Phi(S) \le \mathbb{E}_S[\Phi(S)] + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}$ ***and*** $\mathfrak{R}_m(\mathcal{G}) \le \widehat{\mathfrak{R}}_S(\mathcal{G}) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}$

[**"Foundations of Machine Learning"** Mohri, Rostamizadeh, Talwalkar, 2nd Edition, 2018]

Lem 7 (Sample complexity)

$$\boxed{\mathbb{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\widehat{\mathfrak{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}}$$

$$R_{D_A}[h_1, h_2] \leq R_{\mathcal{S}_A}[h_1, h_2] + 2\hat{\mathscr{R}}_{\mathcal{S}_A}(\ell_{\mathcal{H}}) + 9K^2\sqrt{\frac{\log(6/\delta)}{2m_1}}$$

$$\rho_{\mathcal{C}}(D_B, \mathcal{S}_B) = \sup_{d \in \mathcal{C}}\left\{\mathbb{E}_{x\sim D_B}[d(x)] - \frac{1}{m_2}\sum_{x\in\mathcal{S}_B} d(x)\right\} \lesssim \hat{\mathscr{R}}_{\mathcal{S}_B}(\mathcal{C}) + \sqrt{\frac{\log(1/\delta)}{m_2}}$$

$$\rho_{\mathcal{C}}(h \circ D_A, h \circ \mathcal{S}_A) \lesssim \hat{\mathscr{R}}_{\mathcal{S}_A}(\mathcal{C} \circ \mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{m_1}}$$

$$\boxed{R_{D_A}[h_1, y]} \lesssim \sup_{h_2 \in \mathcal{P}_\omega} R_{\mathcal{S}_A}[h_1, h_2] + c \inf_{h \in \mathcal{P}_\omega} \rho_{\mathcal{C}}(h \circ \mathcal{S}_A, \mathcal{S}_B) + \inf_{h \in \mathcal{P}_\omega} \inf_{\substack{d \in \mathcal{C} \\ \beta(d) \leq 1}} \mathcal{K}(h, d; y)$$

$$+ \hat{\mathcal{R}}_{\mathcal{S}_A}(\ell_{\mathcal{H}}) + \hat{\mathcal{R}}_{\mathcal{S}_A}(\mathcal{C} \circ \mathcal{H}) + \hat{\mathcal{R}}_{\mathcal{S}_B}(\mathcal{C}) + \sqrt{\frac{\log(1/\delta)}{\min(m_1, m_2)}}$$

Lem 3: $\boxed{R_{D_A}[h_1, y]} \leq 3 \sup_{h_2 \in \mathcal{P}} \boxed{R_{D_A}[h_1, h_2]} + 3 \inf_{h \in \mathcal{P}} \boxed{R_{D_A}[h, y]}$

Lem 4: $\boxed{R_{D_A}[h, y]} \leq \frac{2\rho_{\mathcal{C}}(h \circ D_A, D_B)}{2 - \beta(d)} + \frac{2 \boxed{\sup_{u \in \mathcal{X}_A} \|h(u) - y(u)\|_2} \boxed{\leq L}}{2 - \beta(d)} \cdot \mathcal{K}(h, d; y)$

Lem 7: $\boxed{R_{D_A}[h_1, h_2]} \leq R_{\mathcal{S}_A}[h_1, h_2] + 2\hat{\mathcal{R}}_{\mathcal{S}_A}(\ell_{\mathcal{H}}) + 9K^2 \sqrt{\frac{\log(6/\delta)}{2m_1}}$

[**"Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs"**, Galanti et al., JMLR'21]

Takeaway

The **prediction error** of the **unsupervised** image translation method (wrt the ground truth output) can be bounded via
- **minimal statistical distance** attainable by the network and
- variance between solutions that attain that lowest statistical distance.
And this bound actually works in practice!

Comments

1. Regression CNNs can fit almost random (x,y) pairs - can I2I networks fit random (x1, x2) pairs? if so why doesn't R[h1, h2] explode?
2. The empirical **Rademacher complexity** of the discriminator class seems related to the "**expected statistical distance between random splits** of that dataset"?

Recap

1. “Neural GAN distances” between datasets seem to have have **better sample complexity** than “classical” distances. We used an ϵ-net over NN weights and Chernoff bound on each element of ϵ-net to show that.
2. These neural distances can be “smoothened” via **instance noise and augmentations** to make gradient descent iterations more stable. By treating random augmentations as Markov operators we showed that in most cases **skipping augmentations with fixed probability** ensures that the neural distance remain “non-leaking” even under augmentations.
3. The **prediction error** of the unsupervised alignment method can be bounded via the **variance between solutions** attaining similar GAN loss.

Thank you for your time!

Main papers:

1. **"Generalization and Equilibrium in Generative Adversarial Nets"** by Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, Yi Zhang, Proceedings International Conference on Machine Learning (PMLR) 2017.

2. **"Training Generative Adversarial Networks with Limited Data"** by Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, Timo Aila; Advances in Neural Information Processing Systems (NeurIPS) 2020.

3. **"Risk Bounds for Unsupervised Cross-Domain Mapping with IPMs"** by Tomer Galanti, Sagie Benaim, Lior Wolf; JMLR 2021. *// paper series (2017-2021)*