

Manipulating Natural Images by Learning Relationships between Visual Domains

Ben Usman, Ph.D. Candidate

Department of Computer Science
Boston University

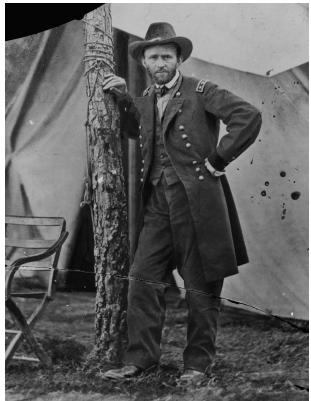
Cross-Domain Image Manipulation: Motivation



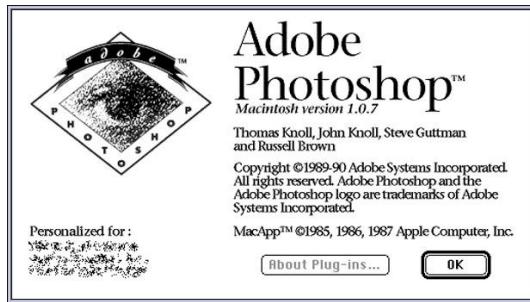
[Lenin and Trotsky during Red Square Demonstration, 1919]



[General Ulysses S. Grant, 1864]

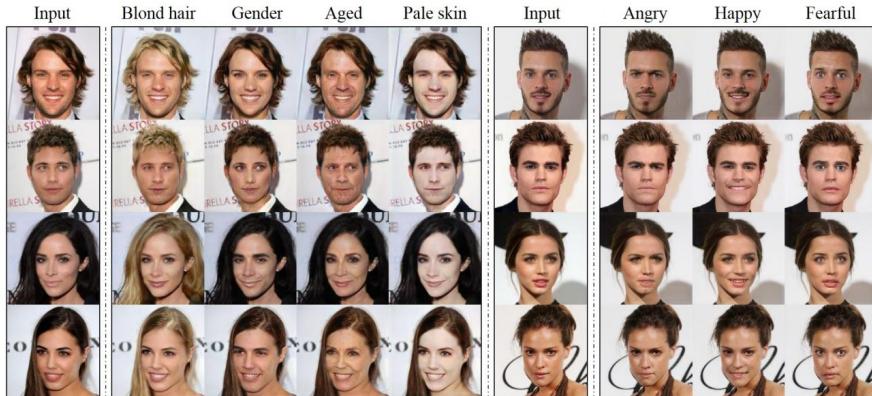
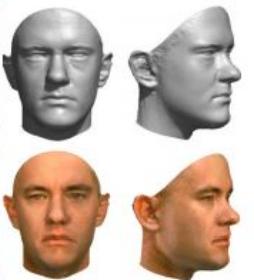


Cross-Domain Image Manipulation: Motivation



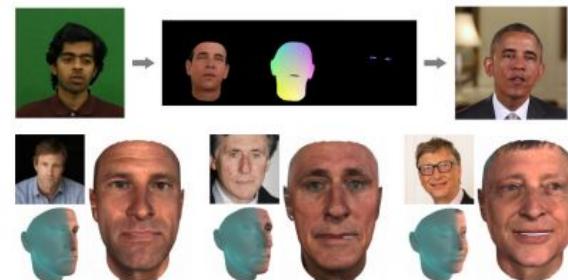
collections of **hand-crafted**
manipulations

parametric models (e.g. face 3DMM)



supervised neural models

?



1999

2009

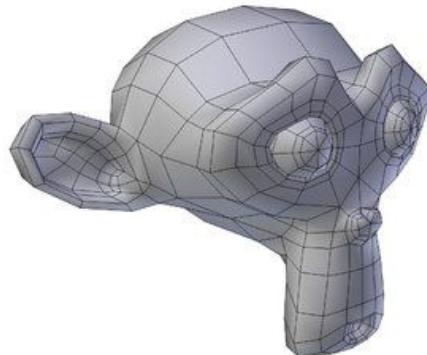
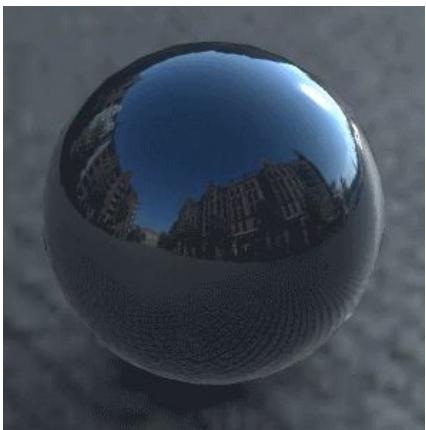
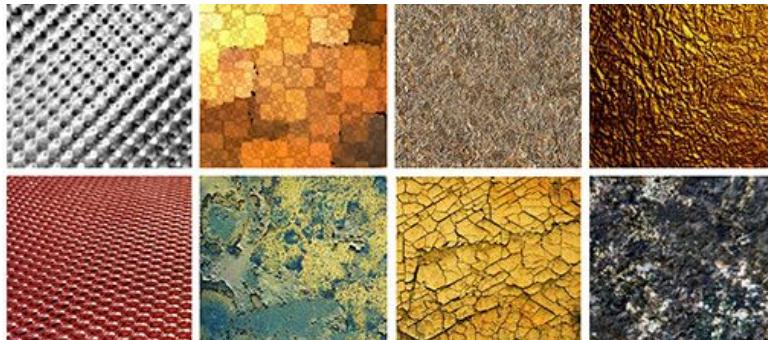
2019

2029

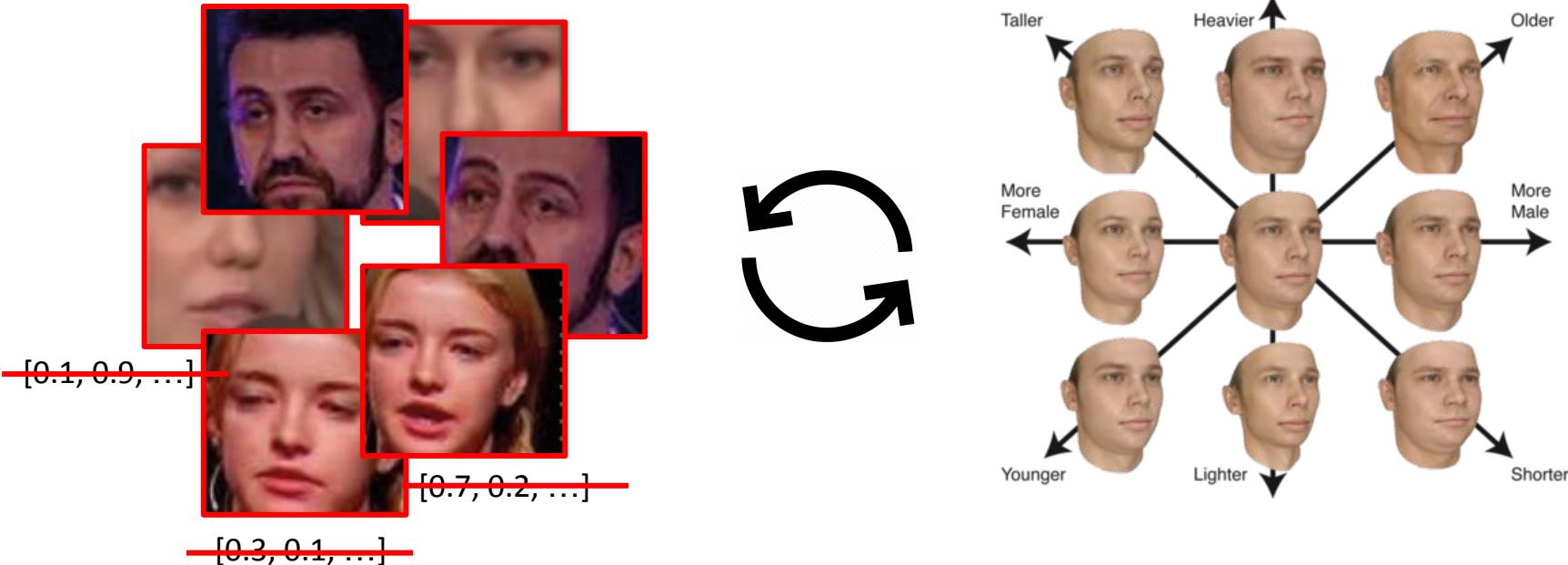
[1] "3D Morphable Face Models - Past, Present and Future", Egger et. al, 2019

[2] "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", Choi et. al, 2018

Cross-Domain Image Manipulation: Motivation



Cross-Domain Image Manipulation: Motivation

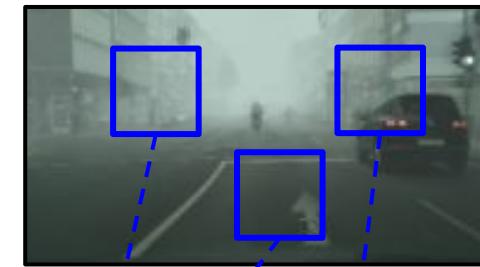
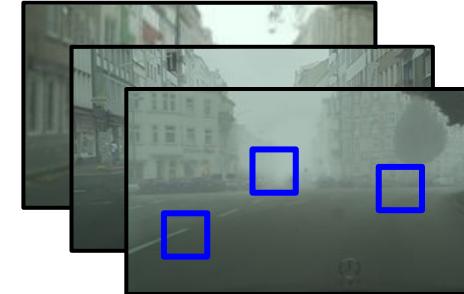
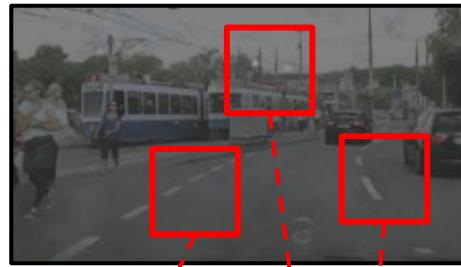
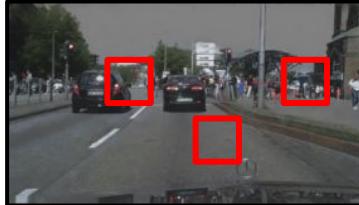


precise control over all attributes!

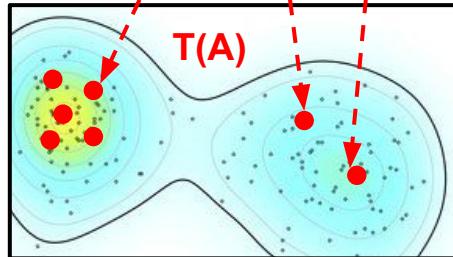
Solution: Unsupervised Image Translation / Domain Alignment



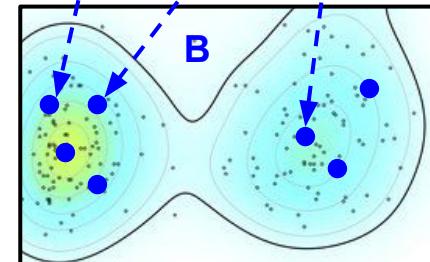
$T(x)$
→



find T that minimizes
distinguishability (



,



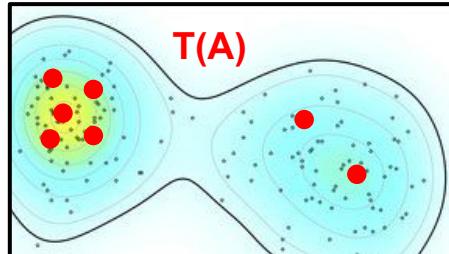
Solution: Unsupervised Image Translation / Domain Alignment

MMD, EMD, CORAL , etc.
✓ easy to optimize
✗ work poorly in higher dims

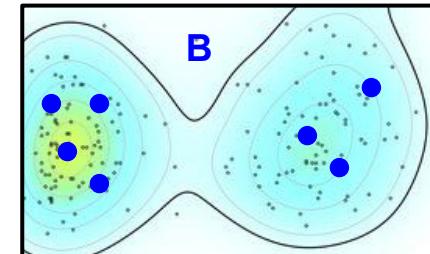


neural adversarial
✓ expressive
✗ unstable training

find T that minimizes
distinguishability (



,



)

Solution: Unsupervised Image Translation / Domain Alignment

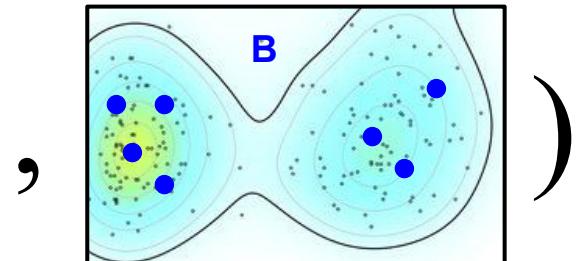
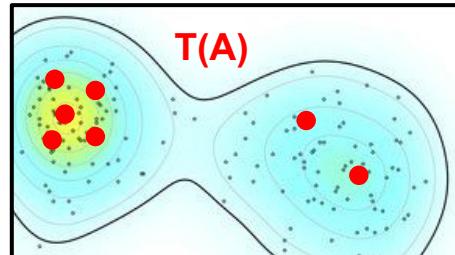


easy to
MMD, EM

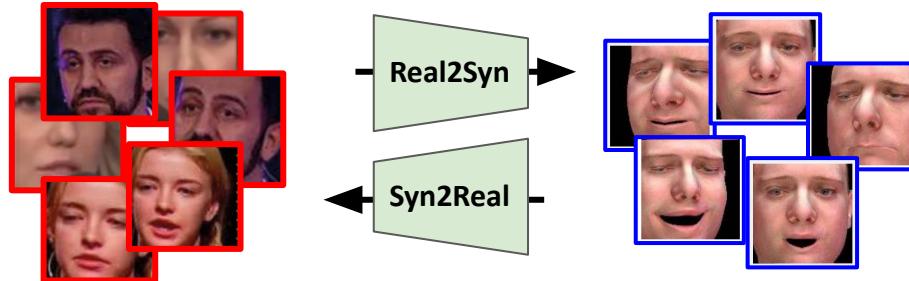
e:
sarial

Topic 1: Stable and expressive distribution alignment.

find T that minimizes
distinguishability (

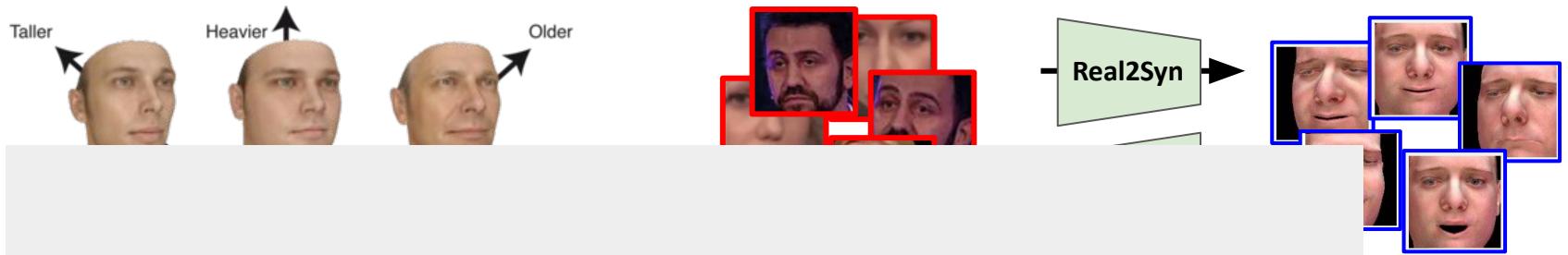


Domain Alignment: Synthetic-to-Real



no precise control over attributes!

Domain Alignment: Synthetic-to-Real

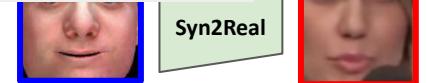


**Topic 2: Precise manipulation of real
data using synthetic supervision.**

precise control over all attributes!

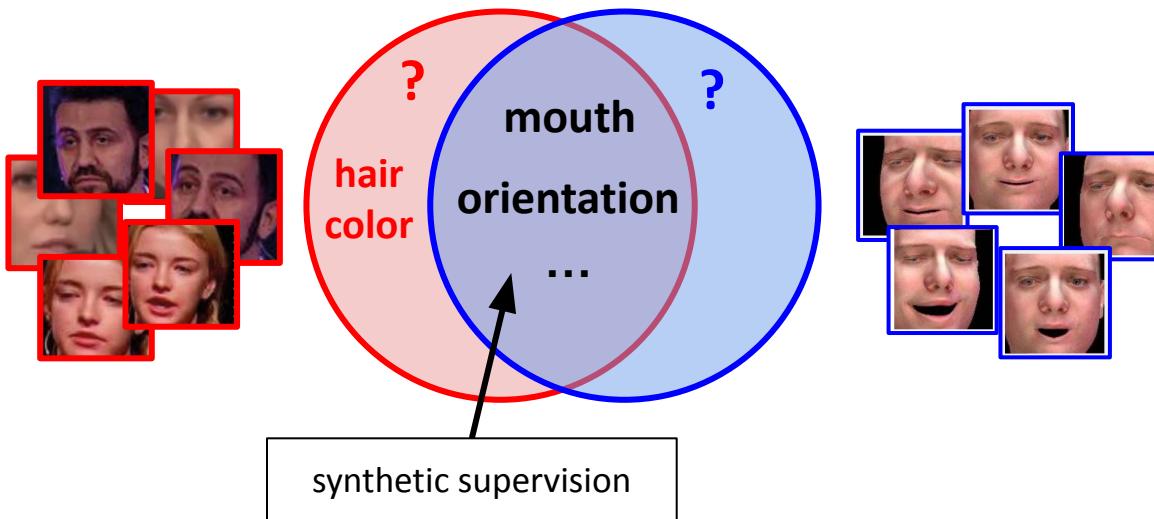
no precise control over attributes!

nthetic
ain



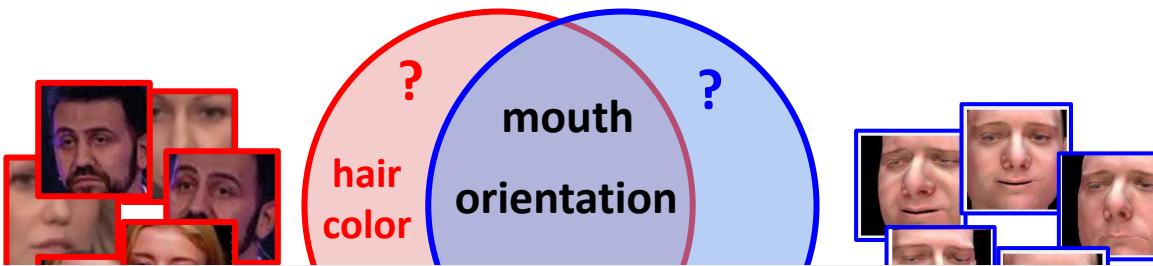
Domain Alignment: Domain-specific factors

factors of variation
in **source** and **target** domains



Domain Alignment: Domain-specific factors

factors of variation
in **source** and **target** domains



**Topic 3: Disentanglement and manipulation
of domain-specific and shared factors
in isolation without supervision.**

Task

Source Distribution

Target Distribution

P_A

P_B

F

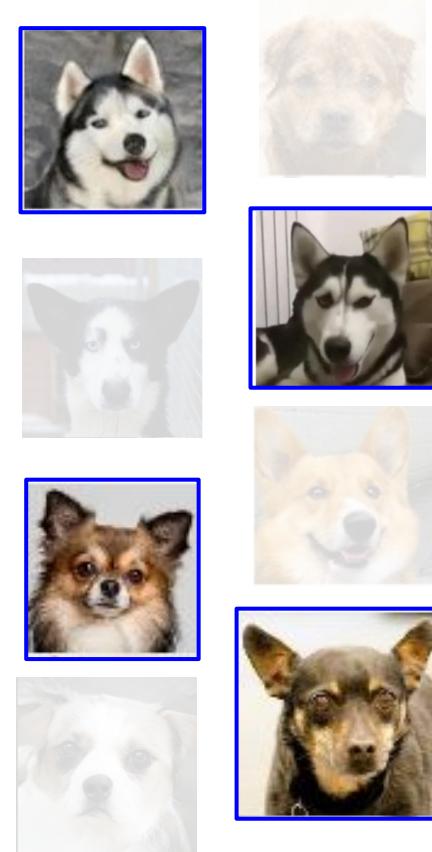
ground truth 1-to-1 cross-domain mapping

Task

Source Samples



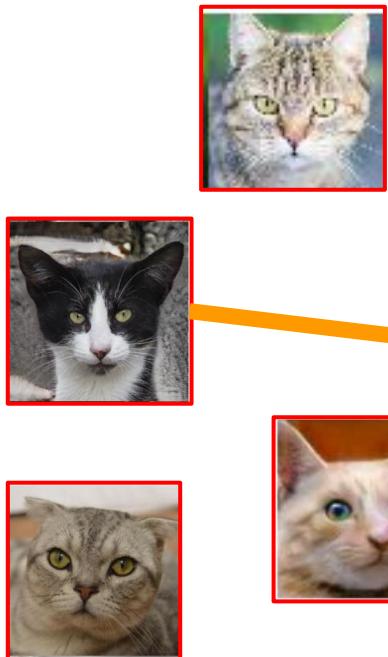
Target Samples



Goal:
reconstruct F from
unpaired samples

Task

Source Samples (Cats)



Target Samples (Dogs)



F



- ✓ is a dog
- ✓ same coat color
- ✓ same pose

...

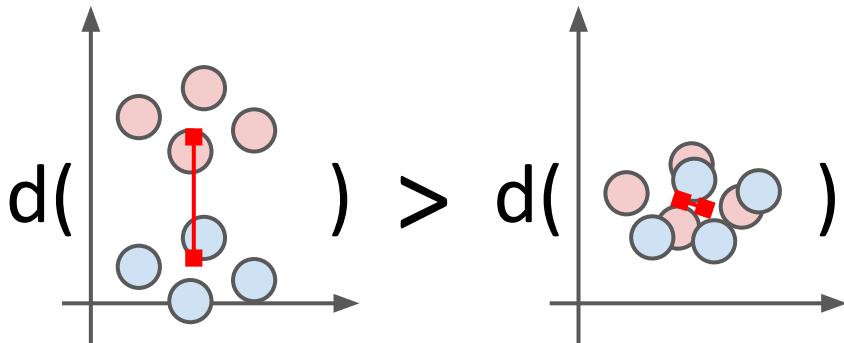


How to find a good F?

(An empirical estimate of)

a statistical distance

$$d(A, B) : \text{Set, Set} \rightarrow \mathbb{R}$$

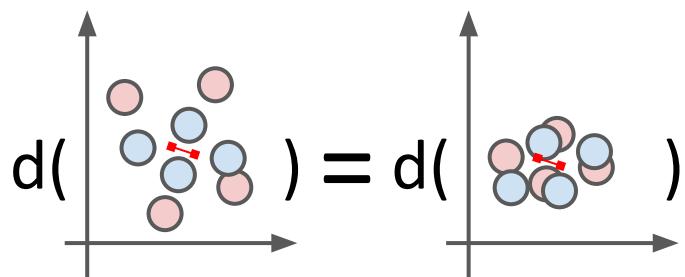


“looks like **A** and **B** are coming from different distributions”

“looks like **A** and **B** might be coming from the same distribution”

Example: difference of means

$$d(A, B) = \|\hat{\mu}(A) - \hat{\mu}(B)\|$$



“look same to me”

How to find a good F?

Source Samples

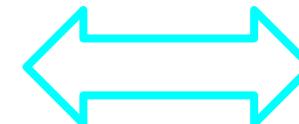


Translated Source Samples

F
→



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



Target Samples



$$A = \{a_i\}$$

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

$$B = \{b_j\}$$

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

How to find a good F? - what we expect

Source Samples



Translated Source Samples



$F_{t=0}$

... optimizing F ...

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

$F_{t=T}$

Target Samples



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

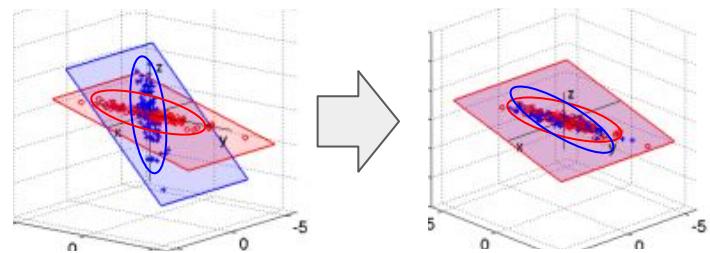


LOW
statistical
distance
 $d(F(A), B)$



What could go wrong?

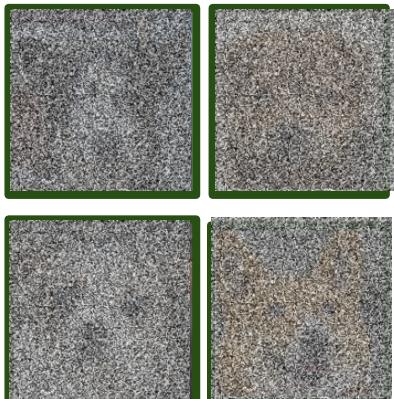
Simple parametric models are “too weak”



Source Samples



Translated Source Samples



LOW
statistical
distance

Target Samples



$$A = \{a_i\}$$

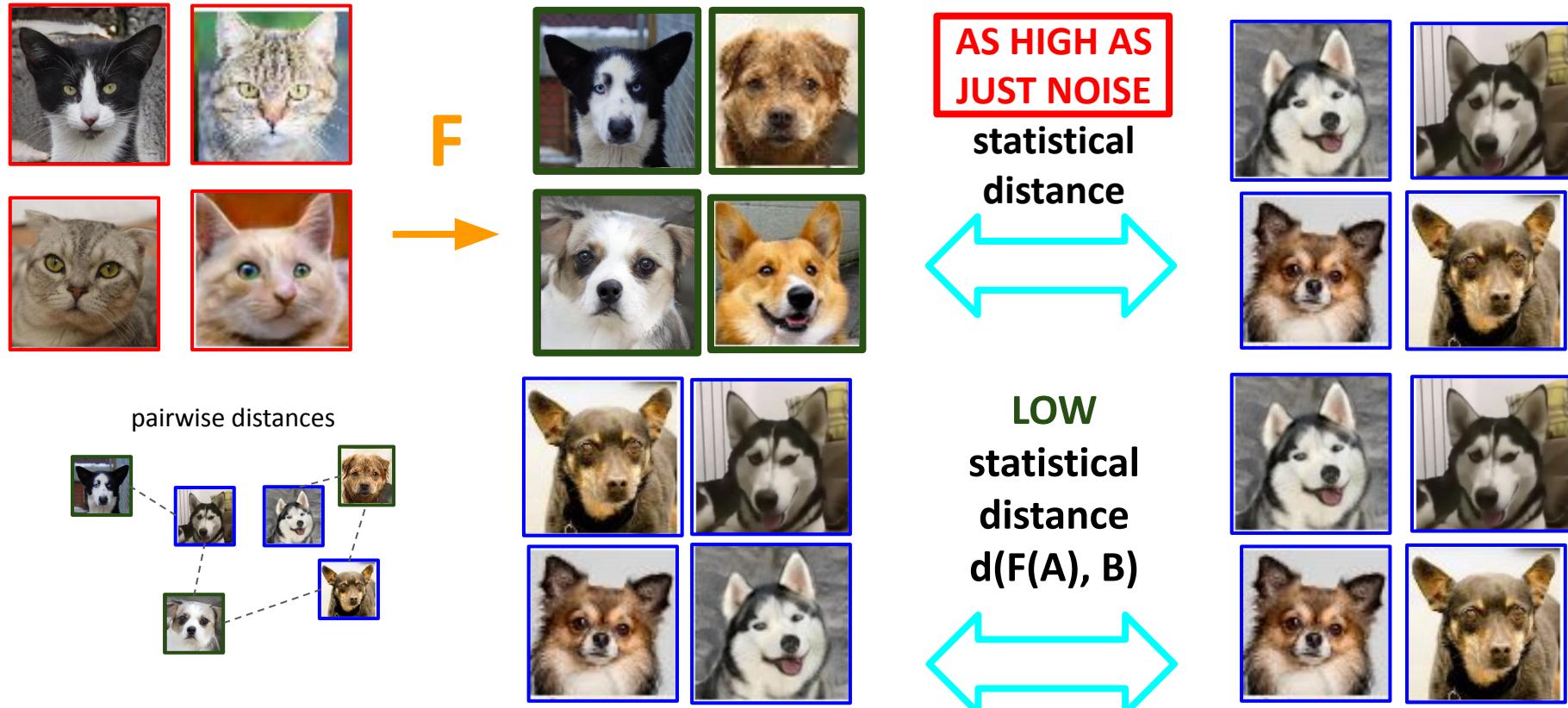
$$F(A) = \{F(a_i) : a_i \in A\}$$

$$B = \{b_i\}$$

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

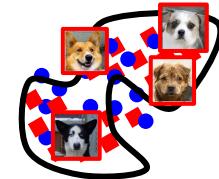
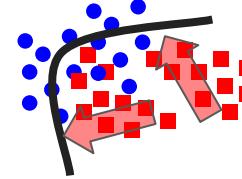
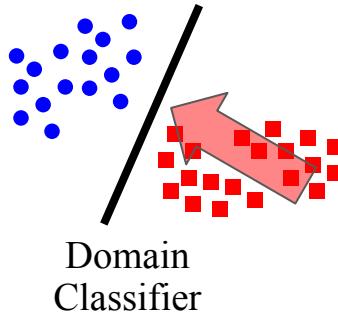
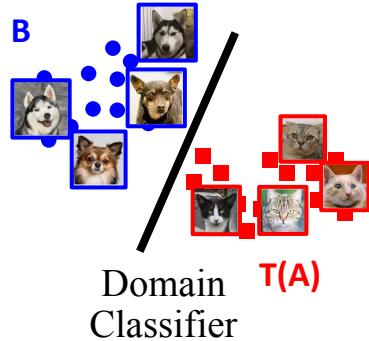
What could go wrong?

Non-parametric models (MMD, EMD) “do not generalize” well



What could go wrong?

Adversarial alignment (GANs) are unstable and fail silently



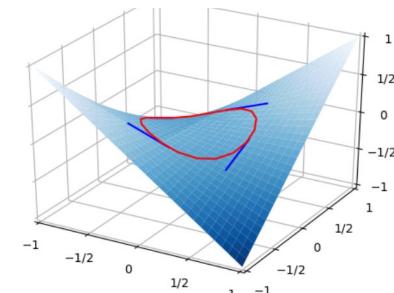
This problem is min-max!

$$\min_G \max_D V(D, G)$$

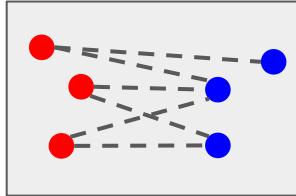
Solving min-max with 1st order methods is **hard**!

$$f(x, y) = xy$$
$$\min_x \max_y f(x, y)$$

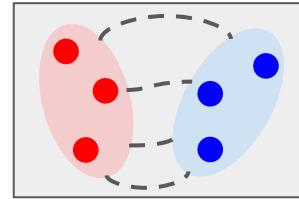
$$g(x, y) = (x, -y)$$
$$x_{t+1} = x_t + \alpha g(x, y)$$



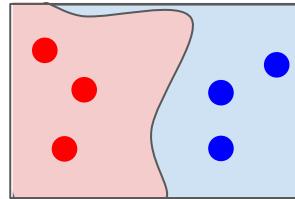
Bounding Likelihood Ratios with Normalizing Flows: Motivation



Non-parametric
(MMD, EMD)



Simple Parametric
(CORAL)



Adversarial (GAN,
Monge–Kantorovich)

closed-form
estimators exists



can be minimized
via gradient descent



model-free
(metric-based)

closed-form
estimators exists



can be minimized
via gradient descent



simple data model
(e.g. normal)

min-max objective

adversarial training

powerful implicit
data model (NN)

Learning better one-to-one mappings

We can get **stable** alignment dy
dualizing the logistic discriminator!
(ICLR-W'18)

We can get **stable** alignment wrt
powerful discriminator families using
normalizing flows! (NeurIPS20)

Defending models against
performing adversarial attacks **on**
themselves improves semantic
consistency! (NeurIPS19)

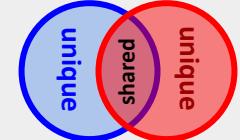
Manipulating factors with cross-domain supervision

We can alter a **single specific attribute** of
real images using **only synthetic**
supervision! (ICCV19 Oral)

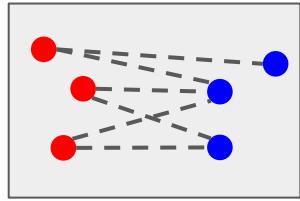


We can manipulate attributes **unique** to
each domain independently from those
shared across domains!

(in submission)



Stable Alignment using Dual Adversarial Distance: Motivation

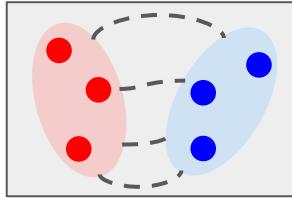


Non-parametric
(MMD, EMD)

closed-form
estimators exists

can be minimized
via gradient descent

model-free
(metric-based)

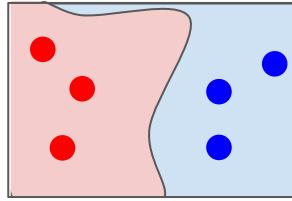


Simple Parametric
(CORAL)

closed-form
estimators exists

can be minimized
via gradient descent

simple data model
(e.g. normal)



Adversarial (GAN,
Monge–Kantorovich)

min-max objective

adversarial training

powerful implicit
data model (NN)

Objective Dualization

min-min objective

gradient descent

Stable Alignment using Dual Adversarial Distance: Our Solution

Adversarial alignment loss **for the logistic discriminator**:

$$d(A, B') = \max_w \sum_{x_i \in A} \log(\sigma(w^T x_i)) + \sum_{x_j \in B'} \log(1 - \sigma(w^T x_j)) - \frac{\lambda}{2} w^T w$$

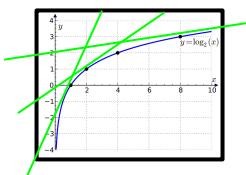
Contribution: an equivalent dual adversarial alignment loss for the logistic D(x).

$$\begin{aligned} d(A, B') &= \min_{0 \leq \alpha_i \leq 1/\lambda} \frac{1}{2} \alpha_A^T Q_{AA} \alpha_A + \frac{1}{2} \alpha_B^T Q_{BB} \alpha_B - \alpha_A^T Q_{AB} \alpha_B + H(\alpha_A) + H(\alpha_B) \\ &\quad \text{s.t. } \|\alpha_A\|_1 = \|\alpha_B\|_1 \end{aligned}$$

$$Q_{AB} = A^T B \quad H(\alpha) = \alpha^T \log \alpha + (1 - \alpha)^T \log(1 - \alpha)$$

Stable Alignment using Dual Adversarial Distance: Our Solution

$$\max_{w,b} \sum_{x_i,y_i \in C_\theta} \log(\sigma(y_i(w^T x_i + b))) - \frac{\lambda}{2} w^T w$$



$$\begin{aligned} \log(\sigma(u)) &\leq \alpha^T u + H(\alpha), \quad \alpha_i \in [0, 1] \\ H(\alpha) &= \alpha^T \log \alpha + (1 - \alpha)^T \log(1 - \alpha) \end{aligned}$$

$$\min_{0 \leq \alpha \leq 1} \max_{w,b} \sum_{x_i,y_i \in C_\theta} \alpha_i y_i (w^T x_i + b) + H(\alpha) - \frac{\lambda}{2} w^T w \quad w^* = \frac{1}{\lambda} (\sum_j x_j y_j \alpha_j)$$

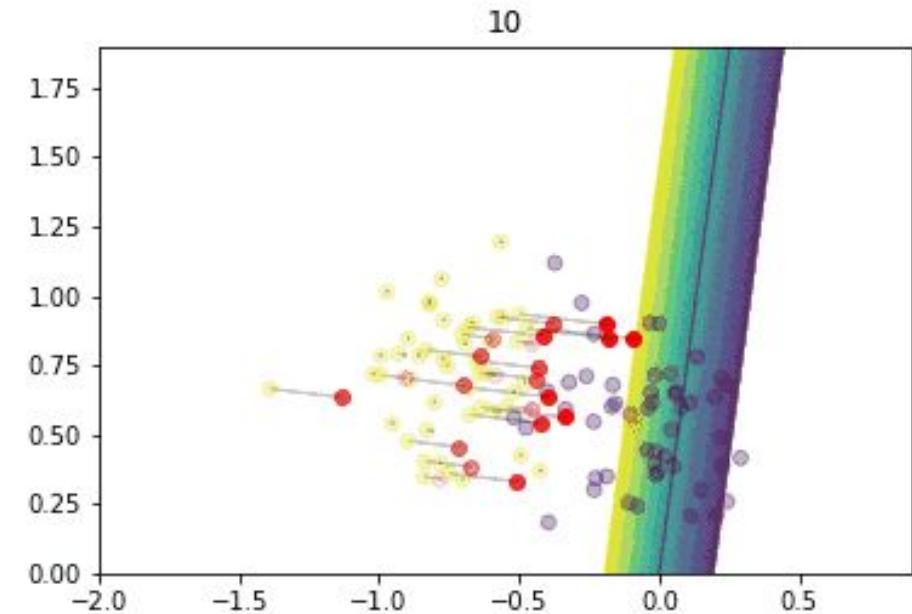
$$= \min_{0 \leq \alpha_i \leq 1} \frac{1}{2\lambda} \sum_{ij} \alpha_i \alpha_j (y_i x_i)^T (y_j x_j) + H(\alpha) = \min_{0 \leq \alpha_i \leq 1} \frac{1}{2\lambda} \alpha^T Q \alpha + H(\alpha) =$$

$$\min_{0 \leq \alpha_i \leq 1/\lambda} \frac{1}{2} \alpha_A^T Q_{AA} \alpha_A + \frac{1}{2} \alpha_B^T Q_{BB} \alpha_B - \alpha_A^T Q_{AB} \alpha_B + H(\alpha_A) + H(\alpha_B)$$

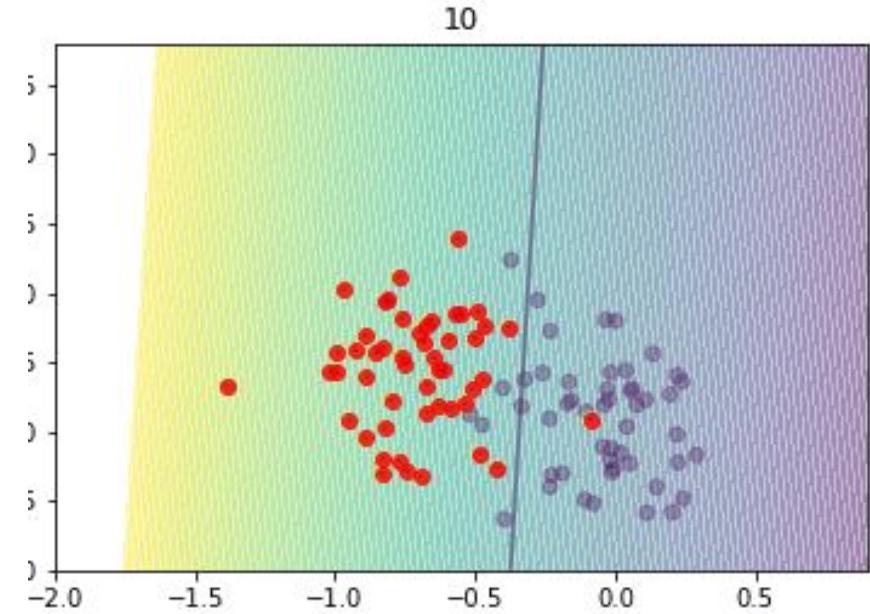
$$\text{s.t. } \|\alpha_A\|_1 = \|\alpha_B\|_1$$

Stable Alignment using Dual Adversarial Distance: Experiments

Linear dual

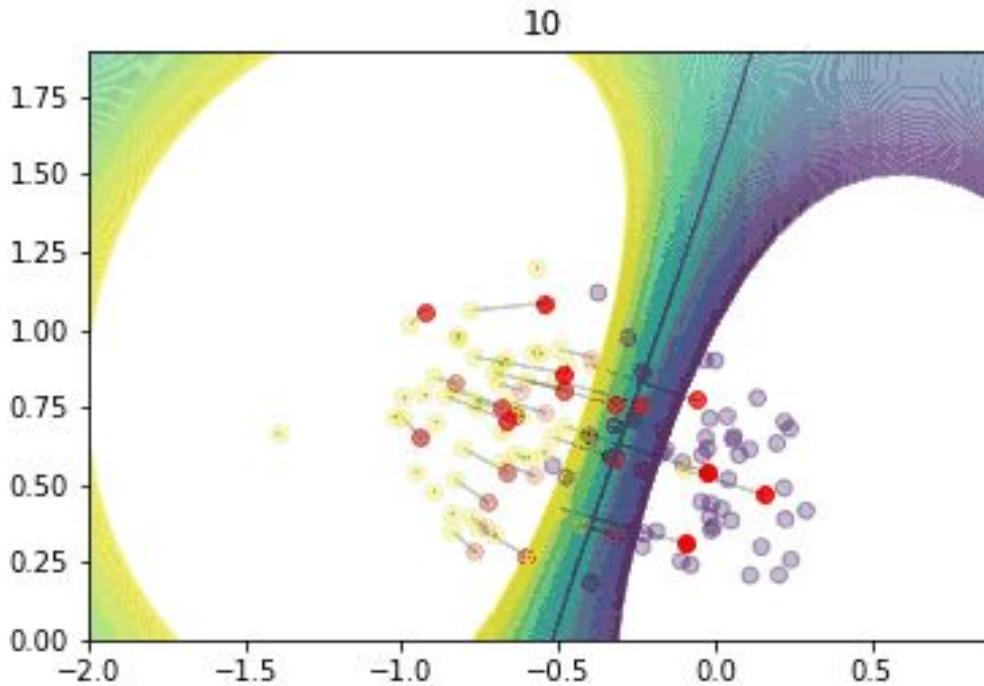


Linear min-max

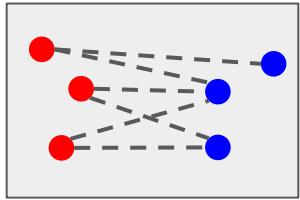


Stable Alignment using Dual Adversarial Distance: Experiments

Kernel dual



Stable Alignment using Dual Adversarial Distance: Takeaway

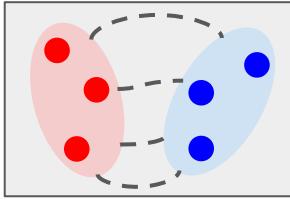


Non-parametric
(MMD, EMD)

closed-form
estimators exists

can be minimized
via gradient descent

model-free
(metric-based)

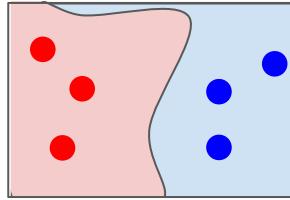


Simple Parametric
(CORAL)

closed-form
estimators exists

can be minimized
via gradient descent

simple data model
(e.g. normal)



Adversarial (GAN,
Monge–Kantorovich)

min-max objective

adversarial training

powerful implicit
data model (NN)

Dual Adversarial
Distance

min-min objective

gradient descent

again, relatively
simple data model

Learning better one-to-one mappings

We can get **stable** alignment dy
dualizing the logistic discriminator!
(ICLR-W'18)

We can get **stable** alignment wrt
powerful discriminator families using
normalizing flows! (NeurIPS20)

Defending models against
performing adversarial attacks **on**
themselves improves semantic
consistency! (NeurIPS19)

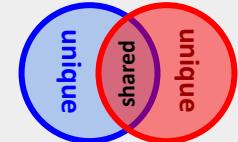
Manipulating factors with cross-domain supervision

We can alter a **single specific attribute** of
real images using **only synthetic**
supervision! (ICCV19 Oral)

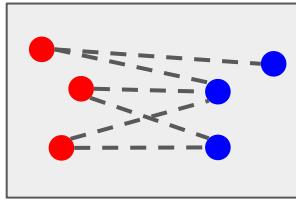


We can manipulate attributes **unique** to
each domain independently from those
shared across domains!

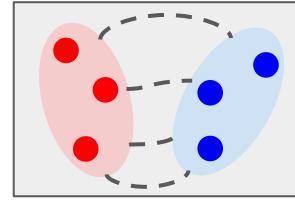
(in submission)



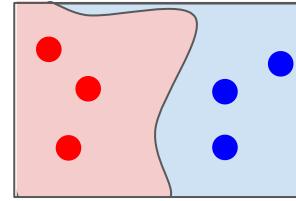
Bounding Likelihood Ratios with Normalizing Flows: Motivation



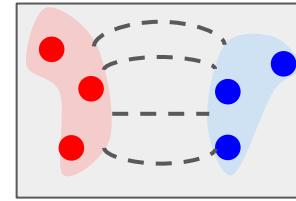
Non-parametric
(MMD, EMD)



Simple Parametric
(CORAL)



Adversarial (GAN,
Monge–Kantorovich)



Log-Likelihood Ratio
Minimizing Flows (new!)

closed-form
estimators exists

can be minimized
via gradient descent

model-free
(metric-based)

closed-form
estimators exists

can be minimized
via gradient descent

simple data model
(e.g. normal)

min-max objective

closed-form
upper bound

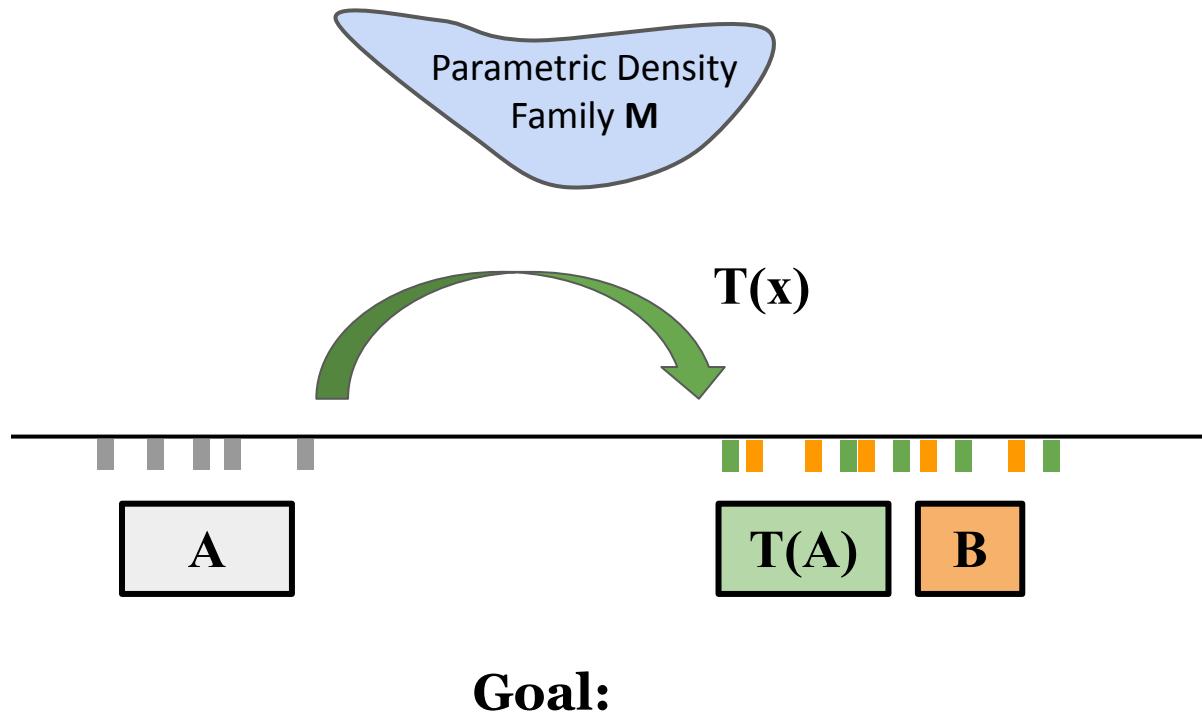
adversarial training

can be minimized
via gradient descent

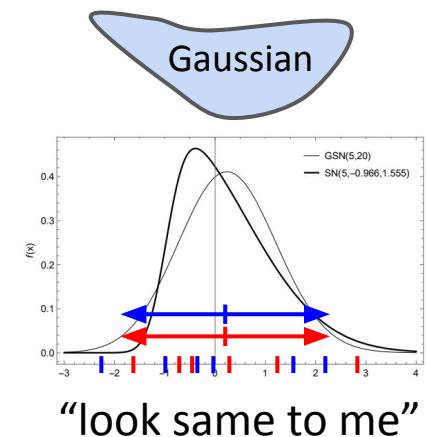
powerful implicit
data model (NN)

any tractable density
+ normalizing flow

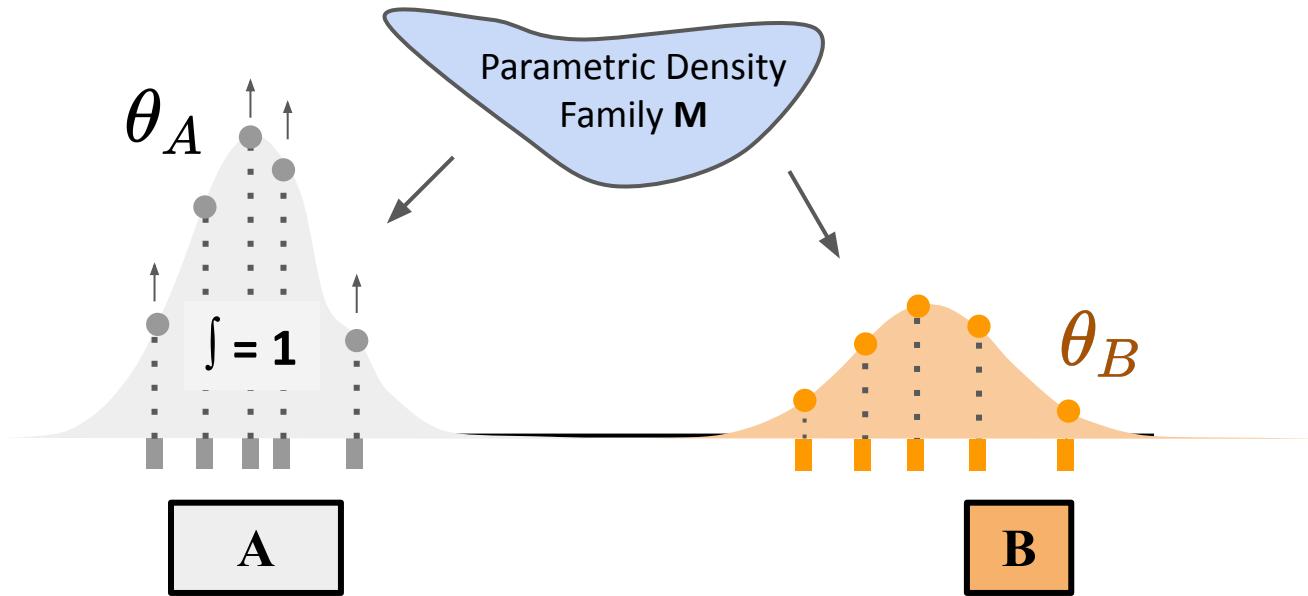
Bounding Likelihood Ratios with Normalizing Flows: Method



find T such that **$T(A)$** and **B** are **indistinguishable for M** .



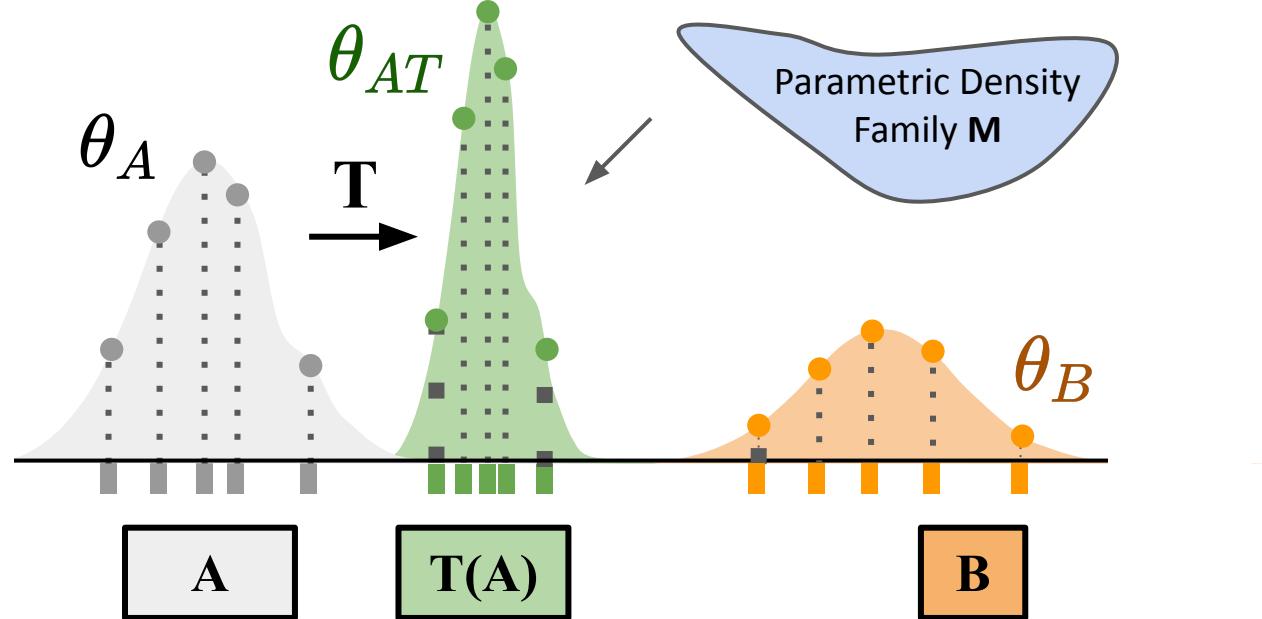
Bounding Likelihood Ratios with Normalizing Flows: Method



We start with two dataset **A** and **B** that we want to align.

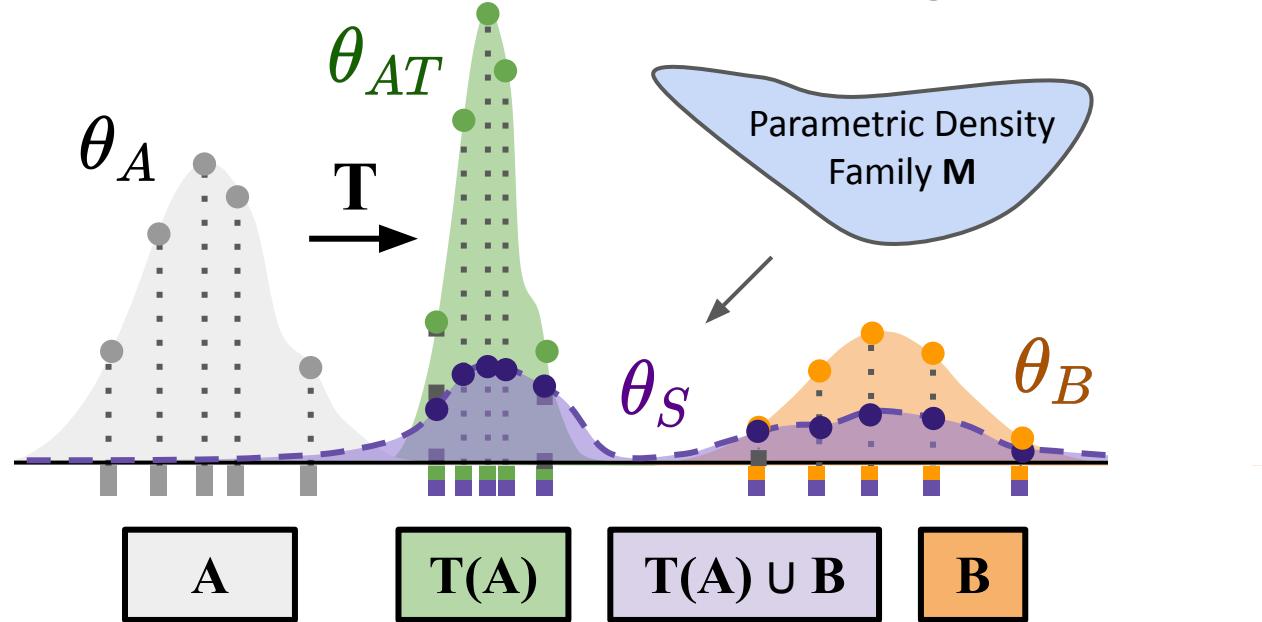
We assume that we fitted two separate density models with parameters θ_A and θ_B to each dataset individually.

Bounding Likelihood Ratios with Normalizing Flows: Method



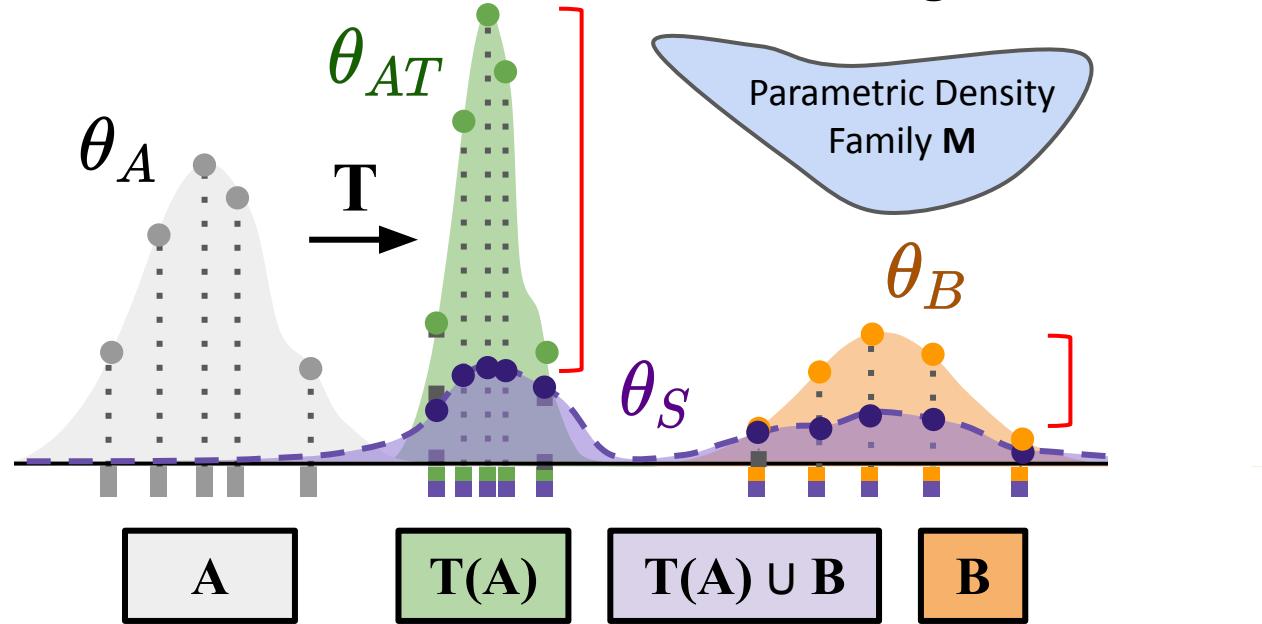
Then we introduce the “transformed” distribution $\mathbf{T}(\mathbf{A})$ and fit a density model to it.

Bounding Likelihood Ratios with Normalizing Flows: Method



Then we introduce the “shared” density model \mathbf{S}
fit to the “combined” dataset $\mathbf{T}(\mathbf{A}) \cup \mathbf{B}$.

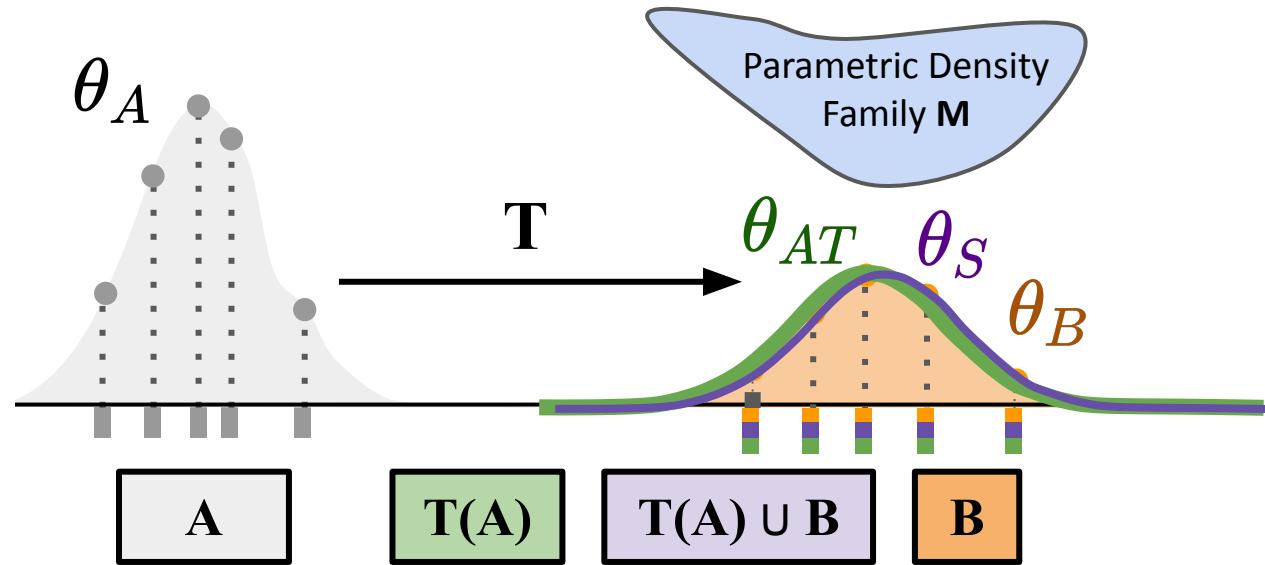
Bounding Likelihood Ratios with Normalizing Flows: Method



Observation 1 (\Rightarrow Lemma 2.1):

The likelihood of the “shared” model (S) trained on the “combined” dataset is always **lower** than likelihoods of “private” models trained on each dataset alone (T(A), B), **unless** both datasets are from the **same distribution**.

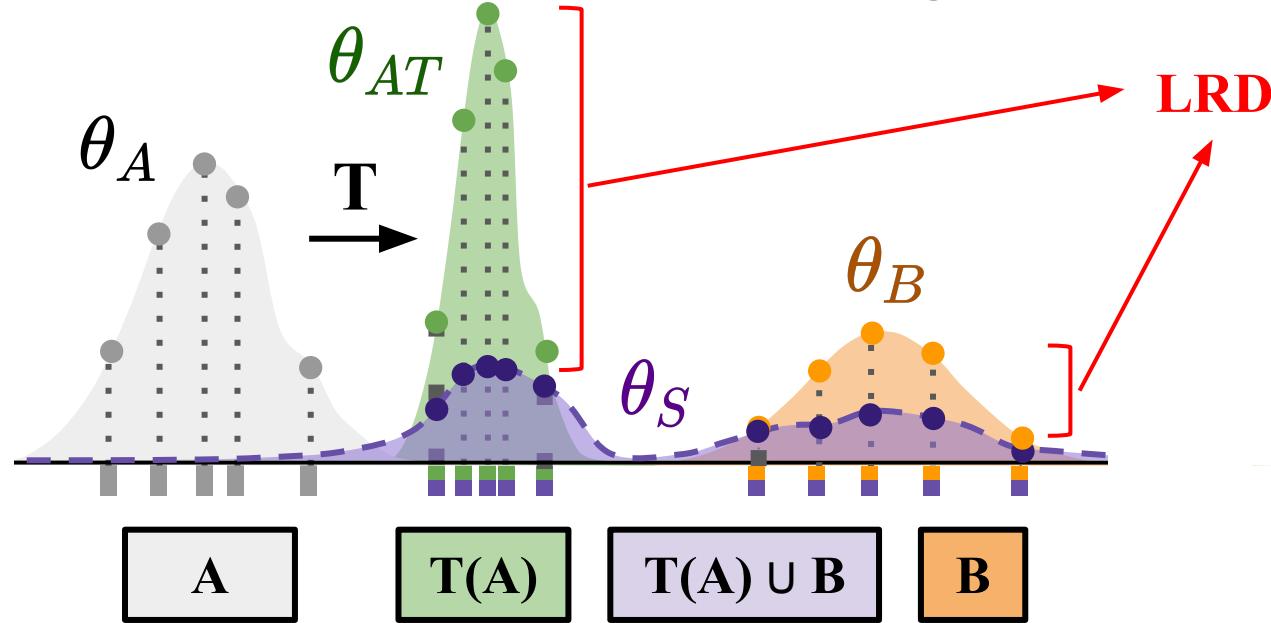
Bounding Likelihood Ratios with Normalizing Flows: Method



Observation 1 (\Rightarrow Lemma 2.1):

The likelihood of the “shared” model (\mathbf{S}) trained on the “combined” dataset is always **lower** than likelihoods of “private” models trained on each dataset alone ($\mathbf{T(A)}$, \mathbf{B}), **unless** both datasets are from the **same distribution**.

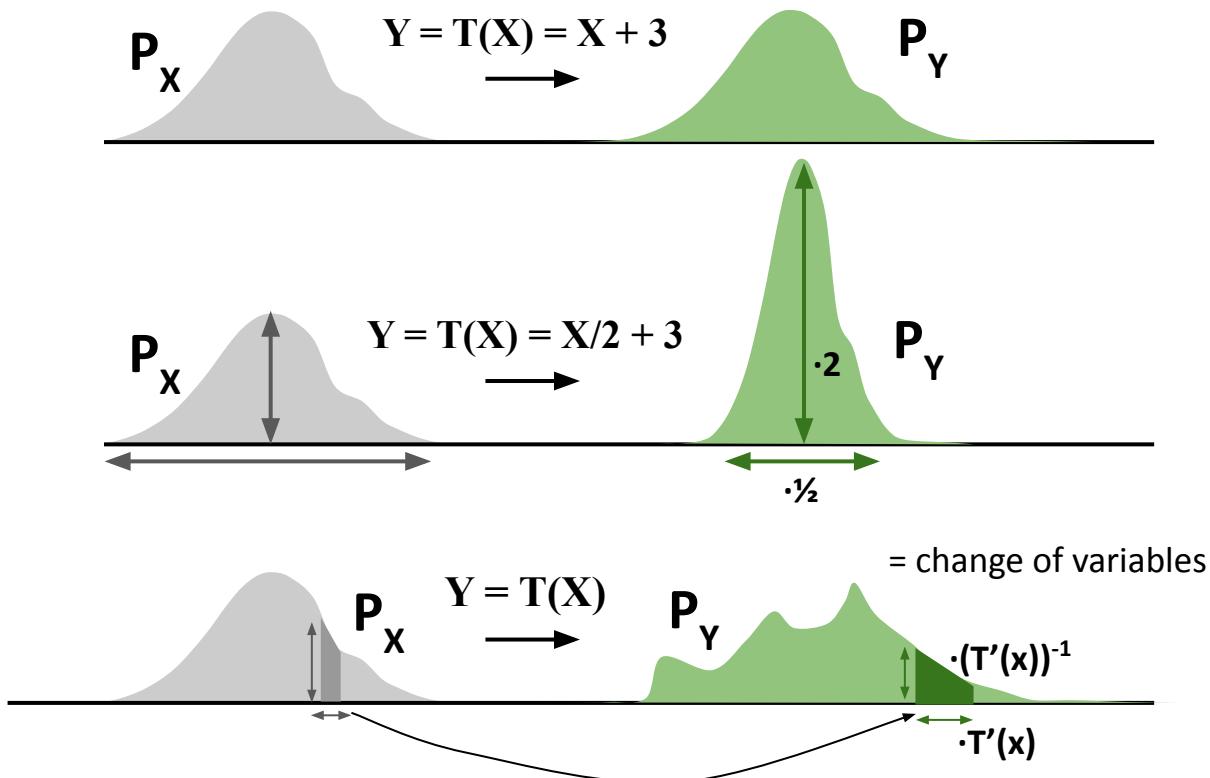
Bounding Likelihood Ratios with Normalizing Flows: Method



Definition 1 (Likelihood-Ratio Distance):

LR-distance between **T(A)** and **B** equals the difference between log-likelihoods of the optimal “shared” density **S** fit to the combined **T(A) \cup B** and two optimal “private” densities fit to **T(A)** and **B** independently.

Bounding Likelihood Ratios with Normalizing Flows: Background

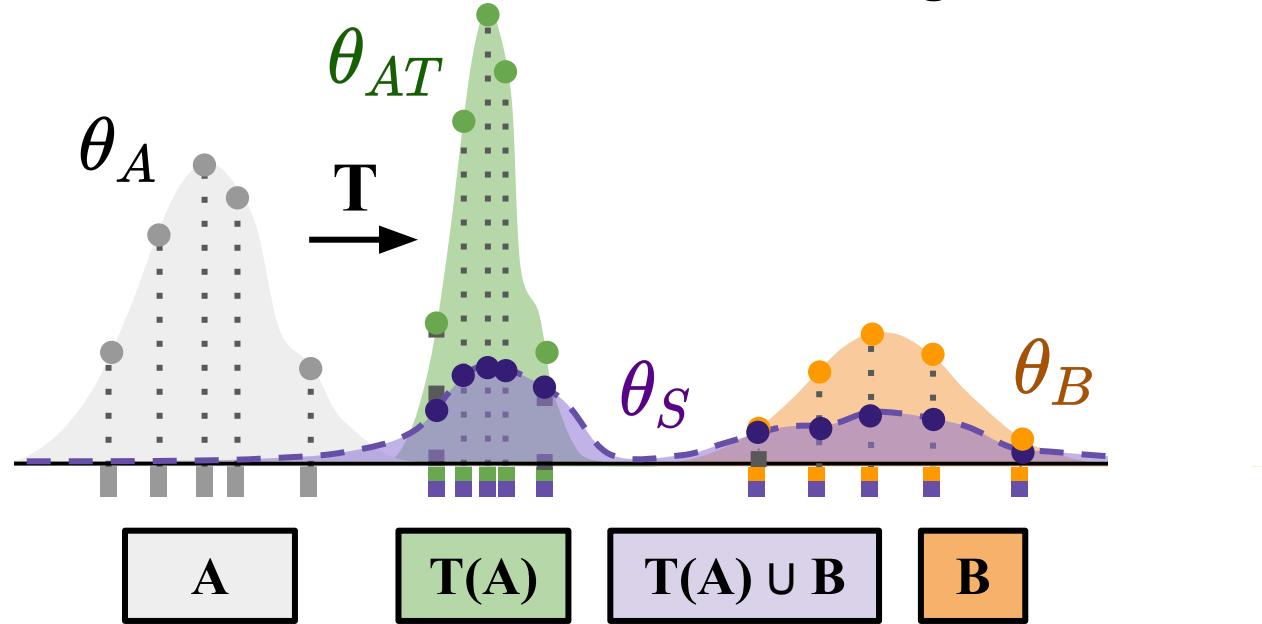


Normalizing flows:

- efficiently invertible
- efficient computation of $\det \text{Jac}[T](x)$

⇒ can apply change of variables formula efficiently!

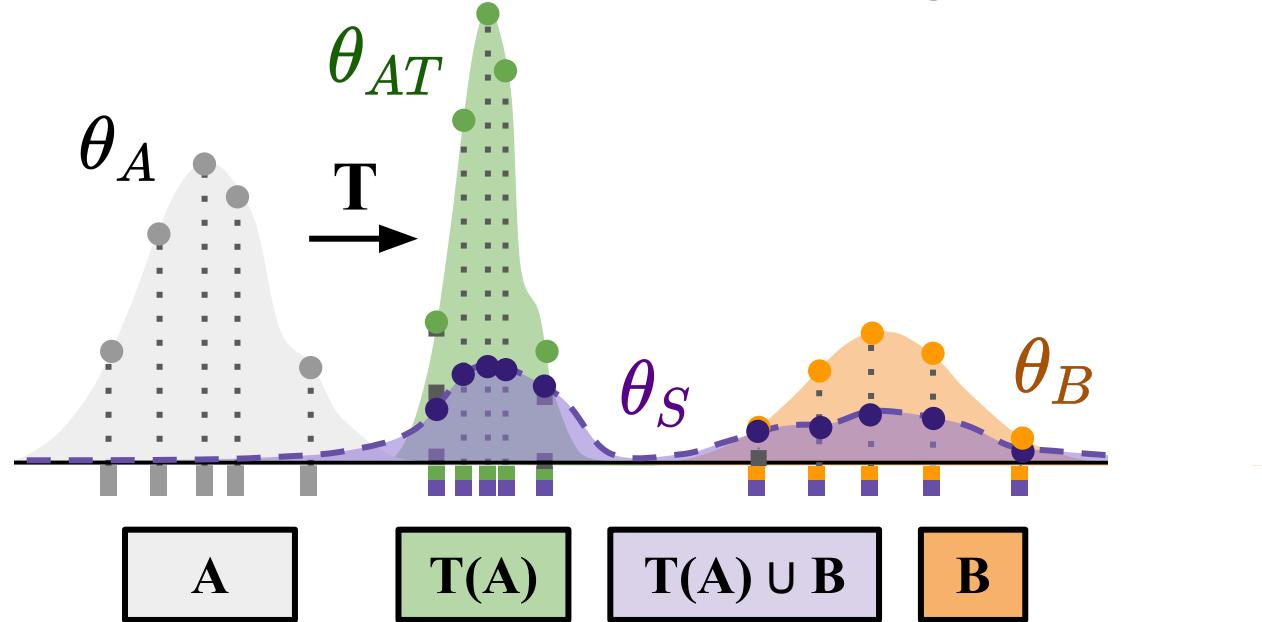
Bounding Likelihood Ratios with Normalizing Flows: Method



Observation 2 (\Rightarrow Lemma 2.2):

The optimal likelihood of the transformed dataset $T(A)$ can be approximated
in closed-form if $T(x)$ is a **normalizing flow**.

Bounding Likelihood Ratios with Normalizing Flows: Method



Conclusion (\Rightarrow Theorem 2.3):

We can find the optimal flow T^* that minimizes the adversarial LR-distance (the “gap” between shared private likelihoods) by minimizing a

$$\mathcal{L}_{\text{LRMF}}(A, B, \phi, \theta_S) = -\log \det |\nabla_x T(A; \phi)| - \log P_M(T(A; \phi); \theta_S) - \log P_M(B; \theta_S) + c(A, B)$$

Bounding Likelihood Ratios with Normalizing Flows: Method

Lemma 2.2. *If $T(x; \phi)$ is a normalizing flow, then the first term in the objective (1) can be bounded in closed form as a function of ϕ up to an approximation error \mathcal{E}_{bias} . The equality in (2) holds when the approximation term vanishes, i.e. if M approximates both A and $T(A; \phi)$ equally well; P_A is the true distribution of A and $T[P_A, \phi]$ is the push-forward distribution of the transformed dataset.*

$$\max_{\theta_{AT}} \log P_M(T(A; \phi); \theta_{AT}) \leq \max_{\theta_A} \log P_M(A; \theta_A) - \log \det |\nabla_x T(A; \phi)| + \mathcal{E}_{bias}(A, T, M) \quad (2)$$

$$\mathcal{E}_{bias}(A, T, M) \triangleq \max_{\phi} \left[\min_{\theta} \mathcal{D}_{KL}(P_A; M(\theta)) - \min_{\theta} \mathcal{D}_{KL}(T[P_A, \phi]; M(\theta)) \right]$$

Bounding Likelihood Ratios with Normalizing Flows: Method

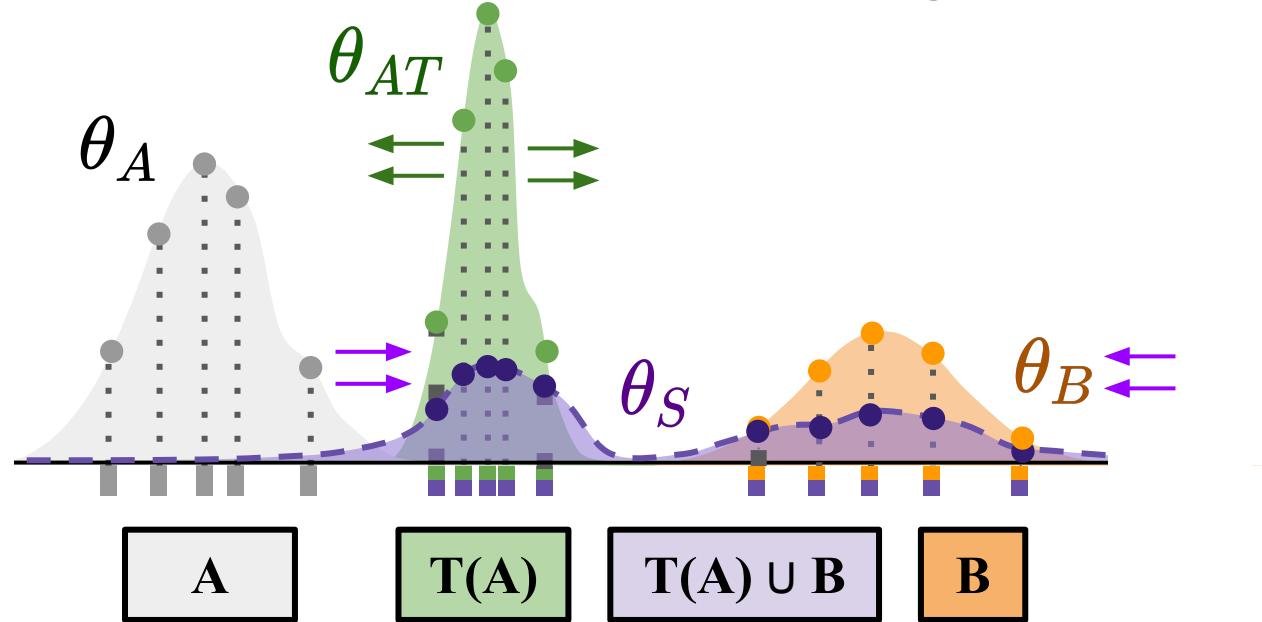
First, we add and remove the true (unknown) entropy $H[P_A] = -\mathbb{E}_{a \sim P_A} \log P_A(a)$:

$$\begin{aligned}\max_{\theta_A} \mathbb{E}_{a \sim P_A} \log P_M(a; \theta_A) &= \max_{\theta_A} \left[\mathbb{E}_{a \sim P_A} \log P_A(a) - \mathbb{E}_{a \sim P_A} \log \frac{P_A(a)}{P_M(a; \theta_A)} \right] \\ &= H[P_A] - \min_{\theta_A} \mathbb{E}_{a \sim P_A} \left[\log \frac{P_A(a)}{P_M(a; \theta_A)} \right] = H[P_A] - \min_{\theta} \mathcal{D}_{KL}(P_A; M(\theta)).\end{aligned}\quad (\star)$$

And then add and remove the (unknown) entropy of the transformed distribution $H[T[P_A, \phi]]$. We also use the change of variable formula $T[P_A](x) = P_A(T^{-1}(x)) \cdot \det |\nabla_x T^{-1}(x)|$, and substitute the expression for $H[P_A]$ from the previous line (\star) :

$$\begin{aligned}\max_{\theta_{AT}} \log P_M(T(A; \phi); \theta_{AT}) &= \max_{\theta_{AT}} \mathbb{E}_{a' \sim T[P_A, \phi]} \log P_M(a'; \theta_{AT}) \\ &= \max_{\theta_{AT}} \left[\mathbb{E}_{a' \sim T[P_A, \phi]} \log T[P_A](a') - \mathbb{E}_{a' \sim T[P_A, \phi]} \log \frac{T[P_A, \phi](a')}{P_M(a'; \theta_{AT})} \right] \\ &= \max_{\theta_{AT}} \left[\mathbb{E}_{a \sim P_A} P_A(T^{-1}(T(a, \phi), \phi)) + \right. \\ &\quad \left. + \log \det |\nabla_x T^{-1}(T(a, \phi), \phi)| - \mathcal{D}_{KL}(T[P_A, \phi]; M(\theta_{AT})) \right] \\ &= H[P_A] - \log \det |\nabla_x T(A, \phi)| - \min_{\theta} \mathcal{D}_{KL}(T[P_A, \phi]; M(\theta)) \\ &\leq \max_{\theta_A} \log P_M(A; \theta_A) - \log \det |\nabla_x T(A, \phi)| + \mathcal{E}_{bias}(A, T, M).\end{aligned}$$

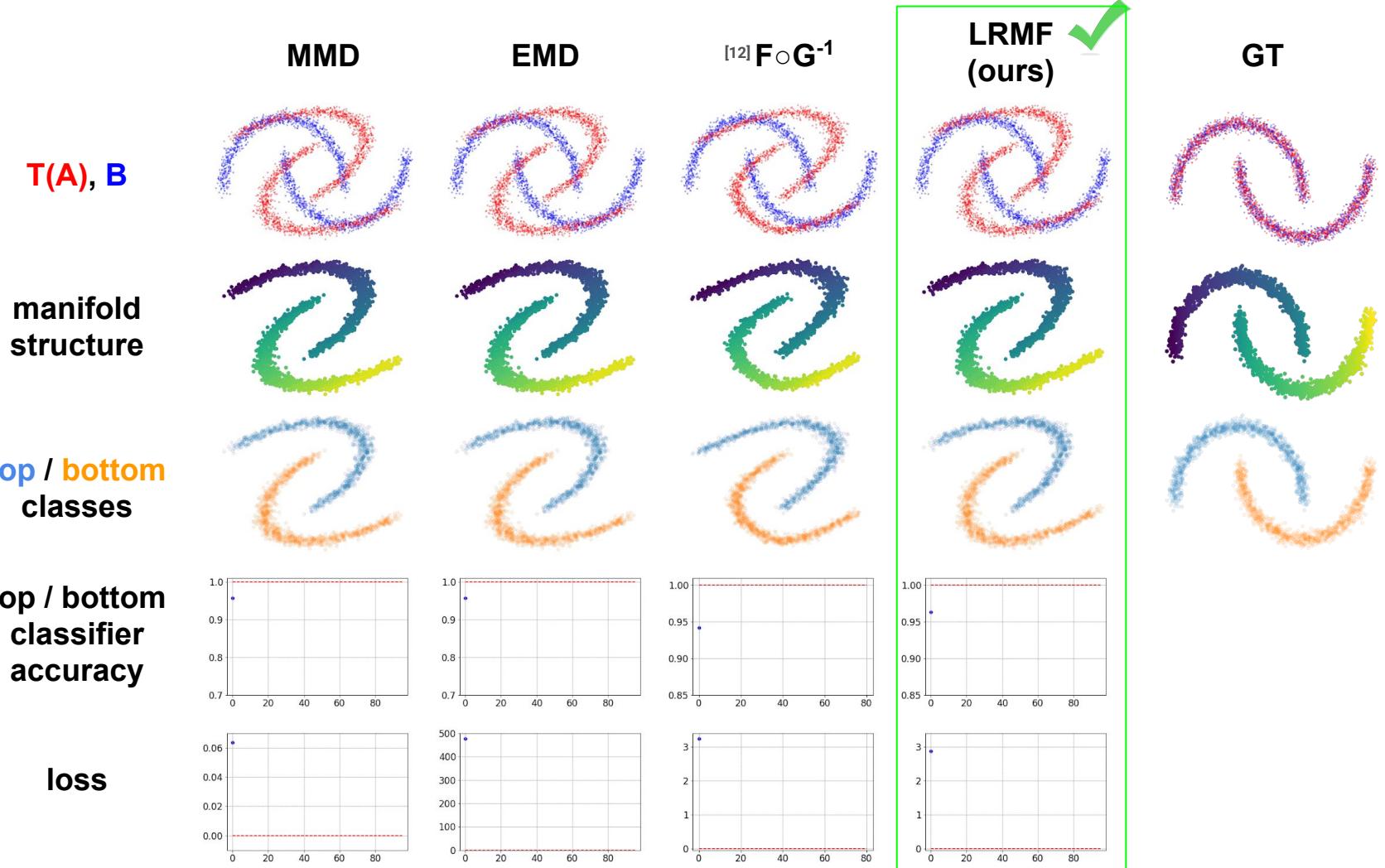
Bounding Likelihood Ratios with Normalizing Flows: Method



Conclusion (\Rightarrow Theorem 2.3):

We can find the optimal flow T^* that minimizes the adversarial LR-distance (the “gap” between shared private likelihoods) by minimizing a

$$\mathcal{L}_{\text{LRMF}}(A, B, \phi, \theta_S) = -\log \det |\nabla_x T(A; \phi)| - \log P_M(T(A; \phi); \theta_S) - \log P_M(B; \theta_S) + c(A, B)$$



Bounding Likelihood Ratios with Normalizing Flows: Special Cases

Special cases:

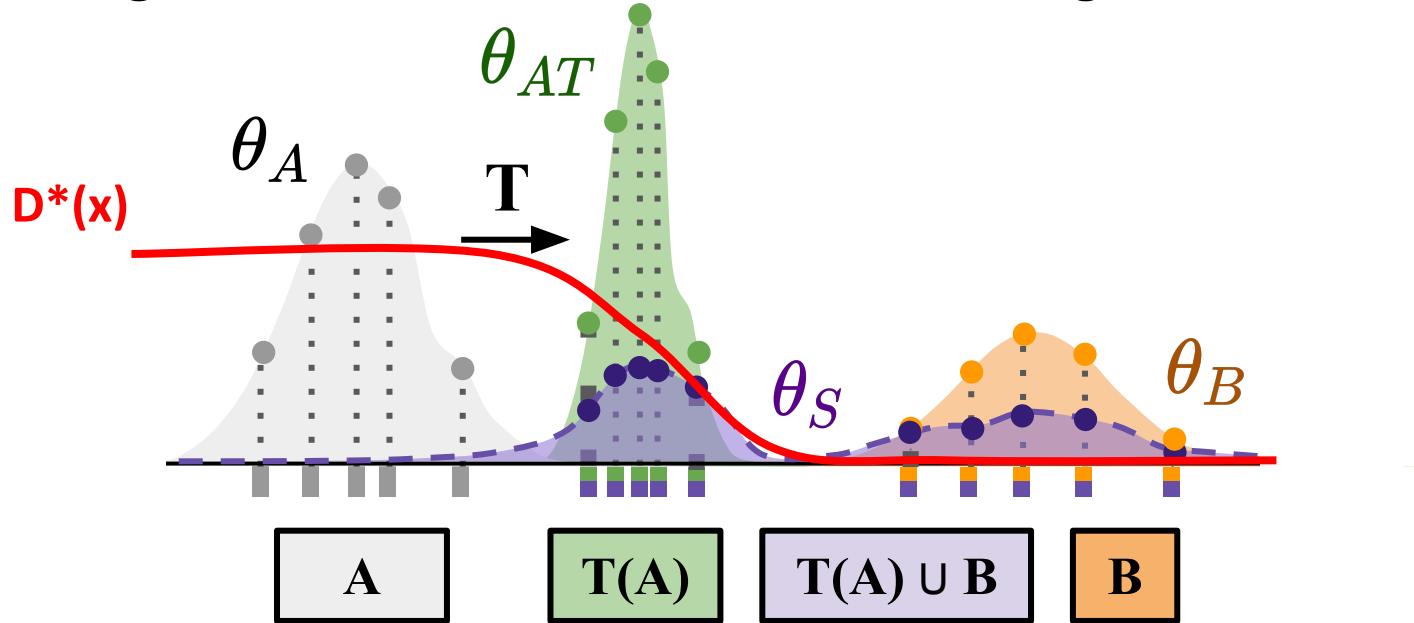
1. Gaussian LRMF \Leftrightarrow matching mean and variance.
2. Minimizing an infinite capacity LRMF loss \Leftrightarrow minimizing JSD

$$d_{\Lambda}(A, B) = 2 \cdot \text{JSD}(A, B) - \mathcal{D}_{KL}(A, M) - \mathcal{D}_{KL}(B, M) + 2 \cdot \mathcal{D}_{KL}((A + B)/2, M)$$

3. Minimizing LRMF \Leftrightarrow training a GAN with a particular discriminator class

$$\max_{D \in \mathcal{H}} [\log D(T(A)) + \log (1 - D(B)) + \log 4], \quad \mathcal{H}(\theta, \theta') = \left\{ \frac{P_M(x; \theta)}{P_M(x; \theta) + P_M(x; \theta')} \right\}$$

Bounding Likelihood Ratios with Normalizing Flows: Method

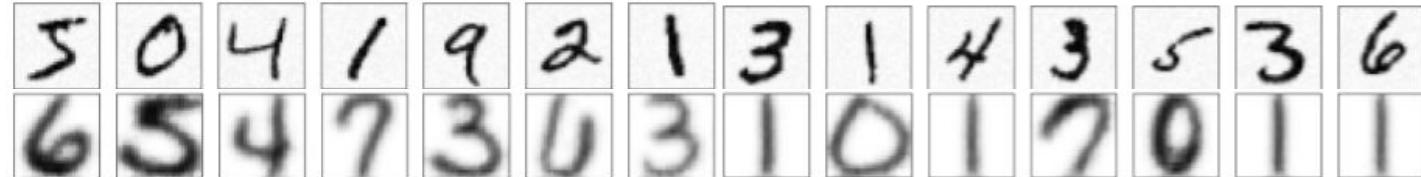


The **Optimal Bayes Classifier** $D^*(x)$ is defined in closed form

$$D^*(x) = \frac{P(X|\theta_{AT})}{P(X|\theta_{AT}) + P(X|\theta_B)}$$

DATA

MNIST (B):

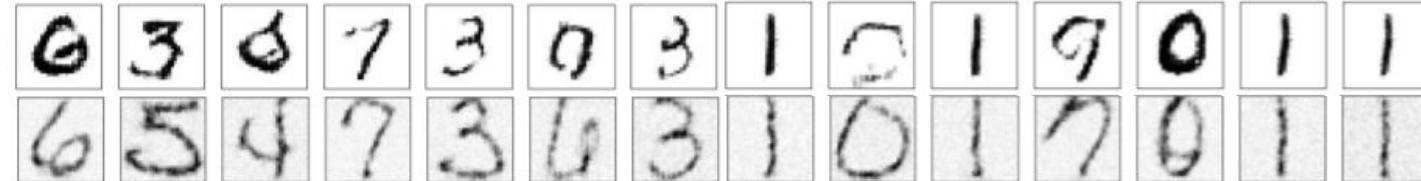


USPS (A):



LRMF T(A)

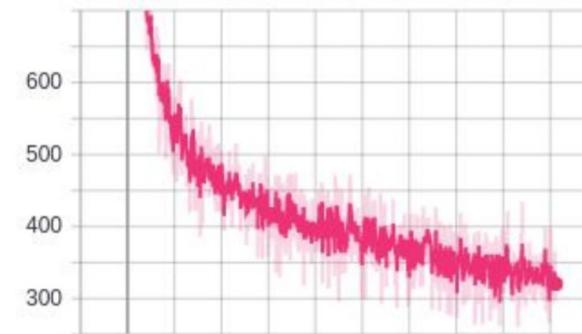
in emb space of VAEGAN:



in pixel space (GLOW):



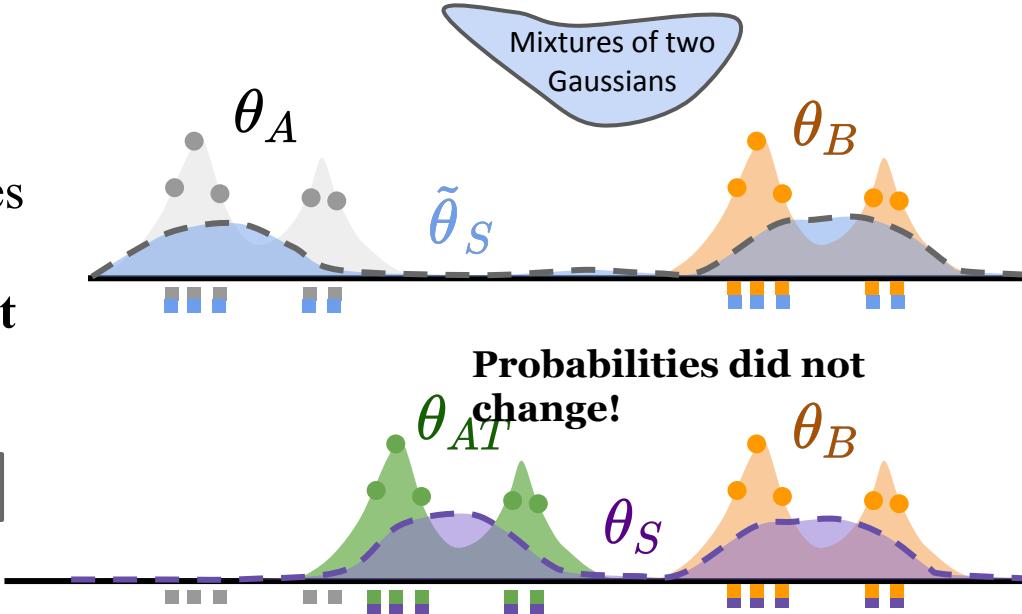
Explicit failure signal:
final LRMF loss $\neq 0$



Bounding Likelihood Ratios with Normalizing Flows: Limitations

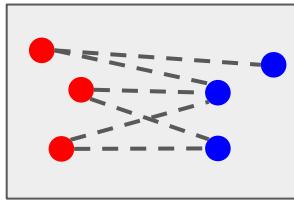
If A and B are far apart, a **shift** $T(x; b)$ does not affect the likelihood of $T(A)$ or S, so the LRMF objective is **locally constant** w.r.t. the transformation parameter b .

$$||[\partial \mathcal{L}_{\text{LRMF}}(A + \mu, B, \phi, \theta) / \partial \phi]|| \propto \exp(-\mu^2)$$



Gradients of LRMF (wrt the **transformation**) between two-gaussian mixtures **vanish** as distribution means become further away from each other.

Bounding Likelihood Ratios with Normalizing Flows: Takeaway



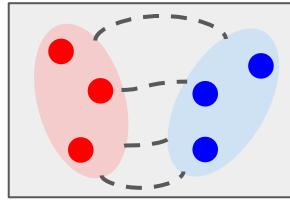
Non-parametric
(MMD, EMD)

model-free
(metric-based)

stable minimization

no mode collapse

stable gradients



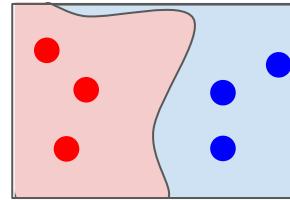
Simple Parametric
(CORAL)

simple data model
(e.g. normal)

stable minimization

no mode collapse

stable gradients



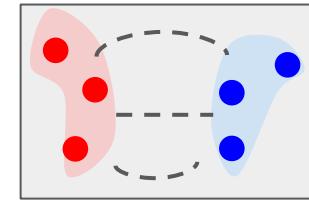
Adversarial (GAN,
Monge–Kantorovich)

powerful implicit
data model (NN)

unstable min-max

mode collapse

vanishing generator
gradients



Log-Likelihood Ratio
Minimizing Flows

any tractable density
+ normalizing flow

stable minimization

no mode collapse

vanishing generator
gradients

Learning better one-to-one mappings

We can get **stable** alignment dy
dualizing the logistic discriminator!
(ICLR-W'18)

We can get **stable** alignment wrt
powerful discriminator families using
normalizing flows! (NeurIPS20)

Defending models against
performing adversarial attacks **on**
themselves improves semantic
consistency! (NeurIPS19)

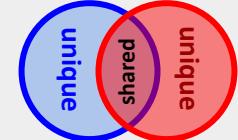
Manipulating factors with cross-domain supervision

We can alter a **single specific attribute** of
real images using **only synthetic**
supervision! (ICCV19 Oral)



We can manipulate attributes **unique** to
each domain independently from those
shared across domains!

(in submission)



Disclaimer

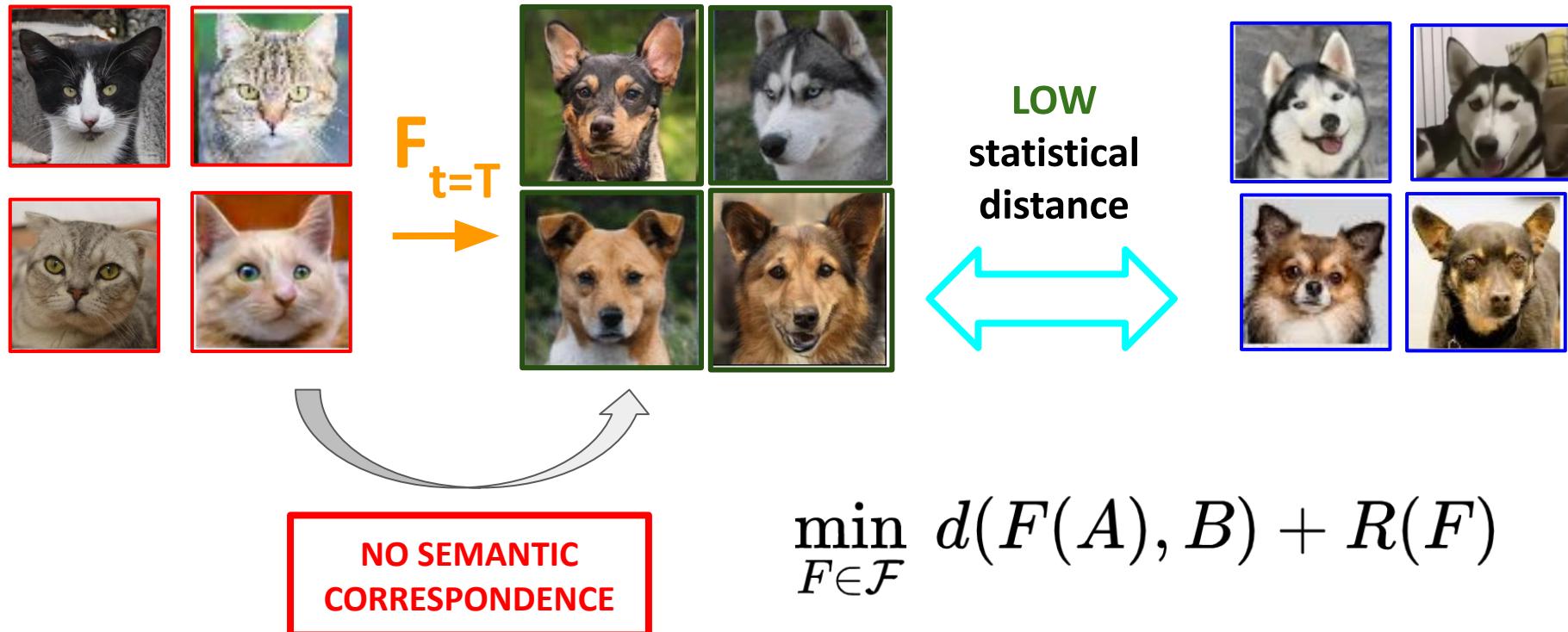
I am the second author, and my contribution is limited mostly to technical help:

“Adversarial Self-Defense for Cycle-Consistent GANs”,
Bashkirova, Usman, Saenko (NeurIPS’19)

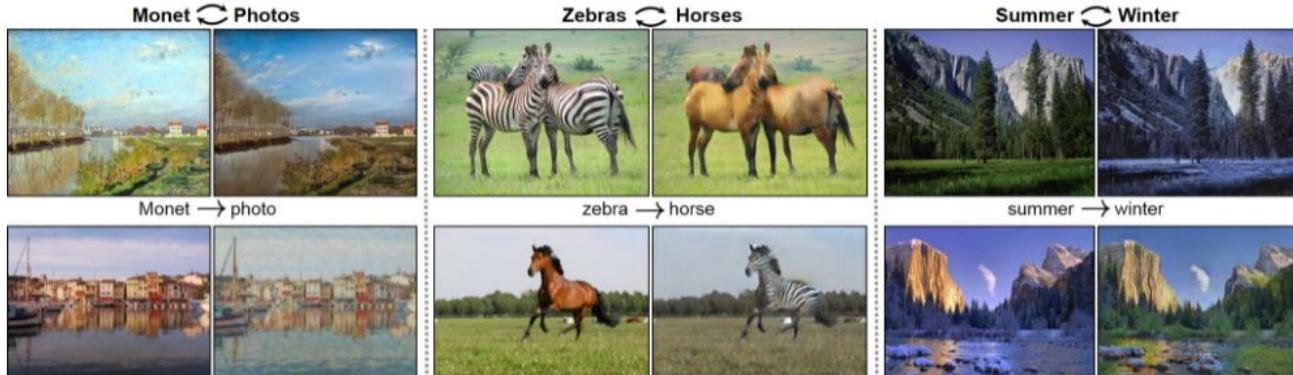
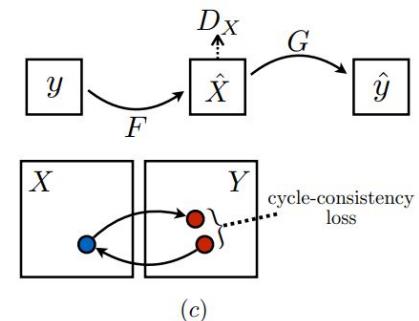
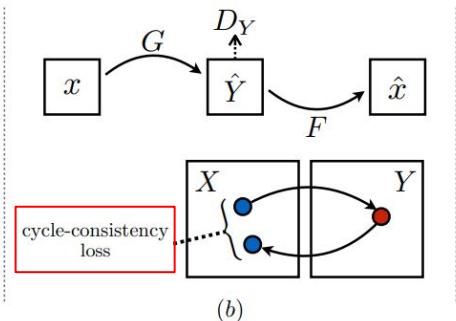
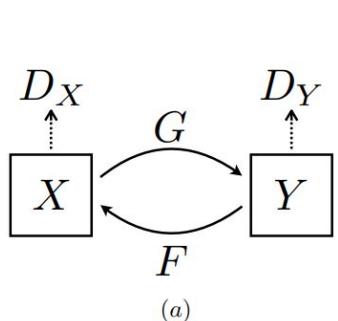
I include this paper in this presentation, because the method proposed in this paper is essential to two remaining papers I talk about in this presentation.

What could go wrong?

The found mapping might be nonsensical

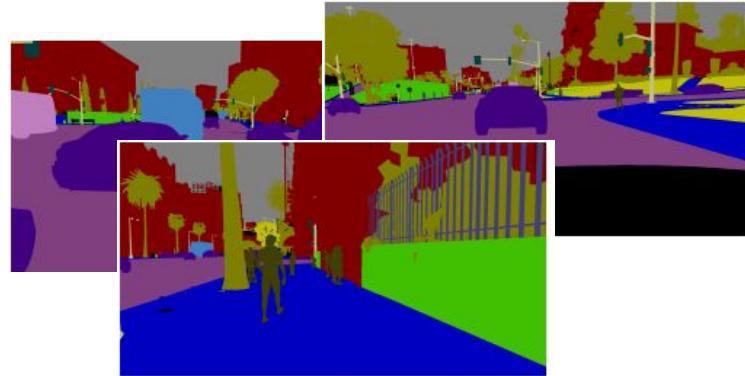


Cycle Reconstruction Improves Semantic Consistency



Improving Semantic Consistency with Translation Honesty

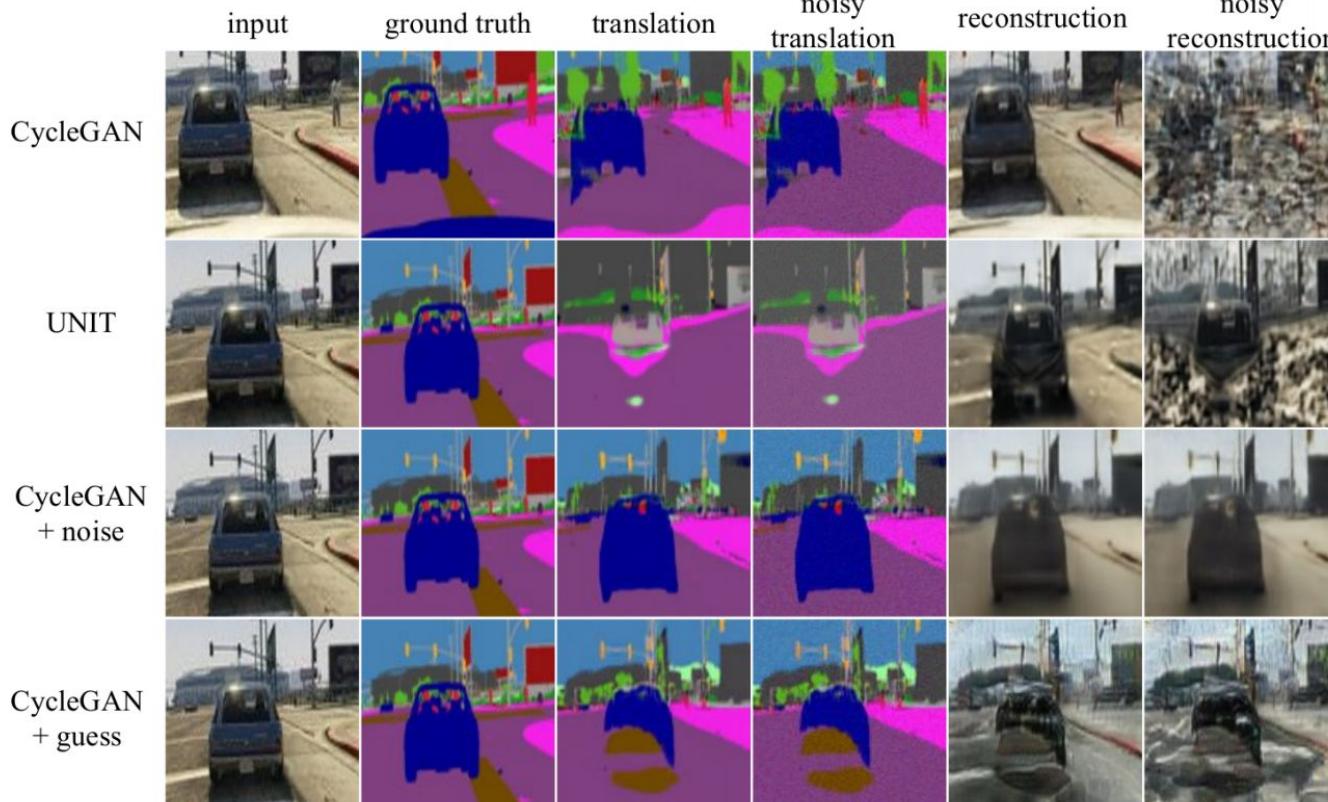
RGB  Segmentation



CycleGAN
UNIT

...

Improving Semantic Consistency with Translation Honesty



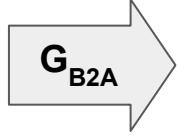
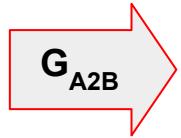
Observation: CycleGAN reconstructs input images perfectly by **cheating** - it embedded **structured noise** into generated translations.

Solution: we propose defence techniques that prevent this “cheating”, and, consequently, **improve the semantic consistency** of outputs.

Improving Semantic Consistency with Translation Honesty

a

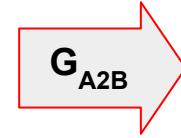
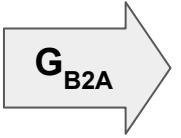
$$b' = G_{A2B}(a)$$



since we **know** that the network
 G_{A2B} adds adversarial noise

b

$$a' = G_{B2A}(b)$$



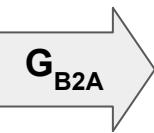
$$b_{cyc} = G_{A2B}(G_{B2A}(b))$$



the cycle-reconstruction b_{cyc}
will **also** have embedded
adversarial noise!

Improving Semantic Consistency with Translation Honesty

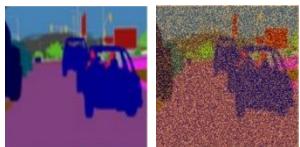
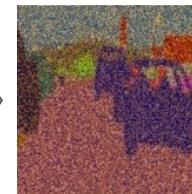
b



$a' = G_{B2A}(b)$

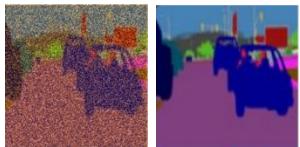


$b_{cyc} = G_{A2B}(G_{B2A}(b))$



D_{noise}

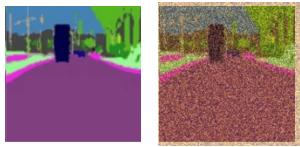
= 0 - “the **first** image is the **original**,
the **second** is a **cycle reconstruction**”



D_{noise}

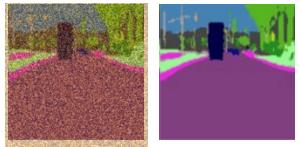
= 1 - “the **first** image is a **cycle reconstruction**,
the **second** is the **original**”

...



D_{noise}

= 0



D_{noise}

= 1

Use the adversarial noise detector
to penalize the model!

Improving Semantic Consistency with Translation Honesty

Reconstruction honesty - how much the performance decreases if we **quantize segmentations**?

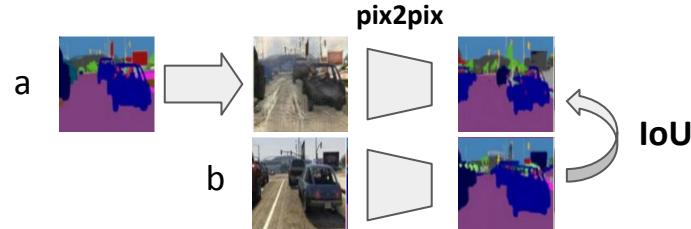
$$RH = \frac{1}{N} \sum_{i=1}^N \{\|G_A(\lfloor G_B(X_i) \rfloor) - Y_i\|_2 - \|G_A(G_B(X_i)) - Y_i\|_2\},$$

Sensitivity to noise - how much the output changes if we **add noise**?

$$SN(\sigma) = \frac{1}{N} \sum_{i=1}^N \|G_A(G_B(X_i) + \mathcal{N}(0, \sigma)) - G_A(G_B(X_i))\|_2$$

Pix2Pix IoU - do generated images produce same segmentation maps as the original?

$$\text{IoU}(\text{pix}(G_A(B_i)), \text{pix}(A_i))$$



Improving Semantic Consistency with Translation Honesty

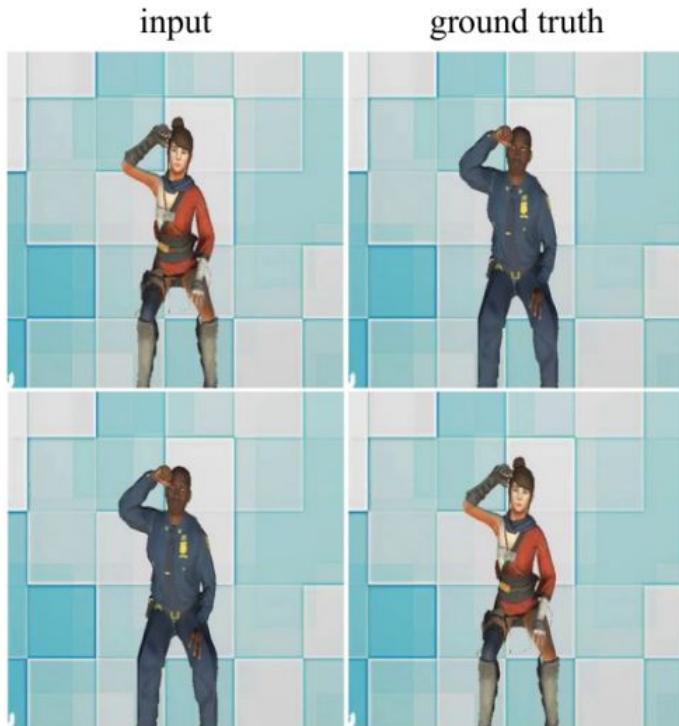
Method	acc. segm↑	IoU segm↑	IoU p2p↑	RH↓	SN↓
CycleGAN	0.23	0.16	0.20	27.43 ± 6.1	446.9
CycleGAN + noise*	0.24	0.17	0.23	9.17 ± 7.4	94.2
CycleGAN + guess*	0.24	0.17	0.21	11.4 ± 7.0	212.6
CycleGAN + guess + noise*	0.236	0.17	0.24	6.1 ± 5.9	150.6
UNIT	0.08	0.04	0.06	6.4 ± 11.7	361.5
MUNIT + cycle	0.13	0.08	0.17	2.5 ± 8.9	244.9
pix2pix (supervised)	0.4	0.34	—	—	—

Table 2: Results on the GTA V dataset.

Method	acc. segm↑	IoU segm↑	IoU p2p↑	RH↓	SN↓
CycleGAN	0.23	0.18	0.21	21.8 ± 5.2	251.2
CycleGAN + noise*	0.24	0.19	0.22	12.27 ± 4.42	222.2
CycleGAN + guess*	0.24	0.184	0.224	7.5 ± 2.4	235.4
CycleGAN + guess + noise*	0.25	0.19	0.22	-0.45 ± 2.3	238.3
UNIT	0.21	0.15	0.12	19.6 ± 6.1	528.2
MUNIT + cycle	0.15	0.09	0.12	21.4 ± 7.9	687.3
pix2pix (supervised)	0.3	0.23	—	—	—

Table 3: Results on the Google Maps dataset.

Improving Semantic Consistency with Translation Honesty



Method	MSE \downarrow	SN \downarrow
CycleGAN	32.55	6.5
CycleGAN+noise*	22.18	1.1
CycleGAN+guess*	23.57	2.4
CycleGAN+guess+noise*	23.13	1.35

Table 1: Results on SynAction dataset: mean square error of the translation and sensitivity to noise.

Improving Semantic Consistency with Translation Honesty: Takeaway

Cycle-consistent models hide information
in the form of adversarial noise.

If we prevent them from doing this,
the semantic consistency of the alignment
improves.

Learning better one-to-one mappings

We can get **stable** alignment dy
dualizing the logistic discriminator!
(ICLR-W'18)

We can get **stable** alignment wrt
powerful discriminator families using
normalizing flows! (NeurIPS20)

Defending models against
performing adversarial attacks **on**
themselves improves semantic
consistency! (NeurIPS19)

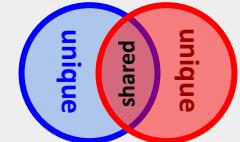
Manipulating factors with cross-domain supervision

We can alter a **single specific attribute** of
real images using **only synthetic**
supervision! (ICCV19 Oral)



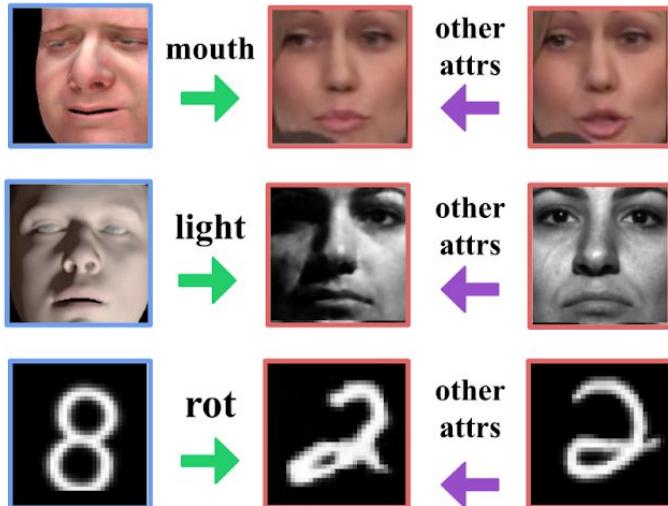
We can manipulate attributes **unique** to
each domain independently from those
shared across domains!

(in submission)

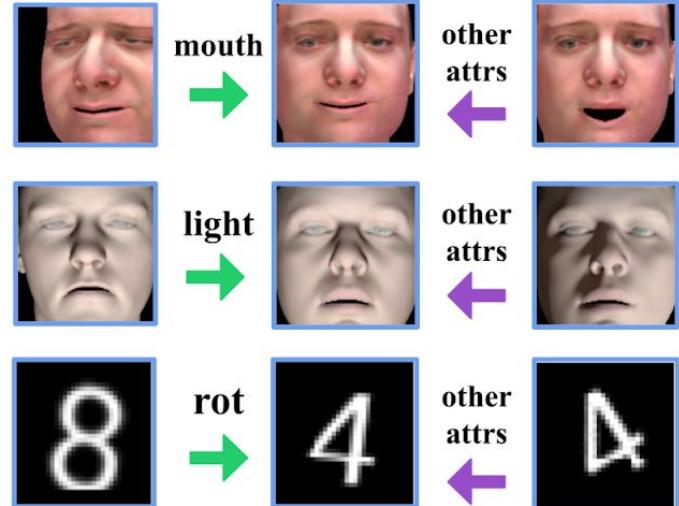


Learning from Cross-Domain Demonstrations: Task

manipulate a **single**
specific **attribute** of a **real** image
using a **synthetic** reference.



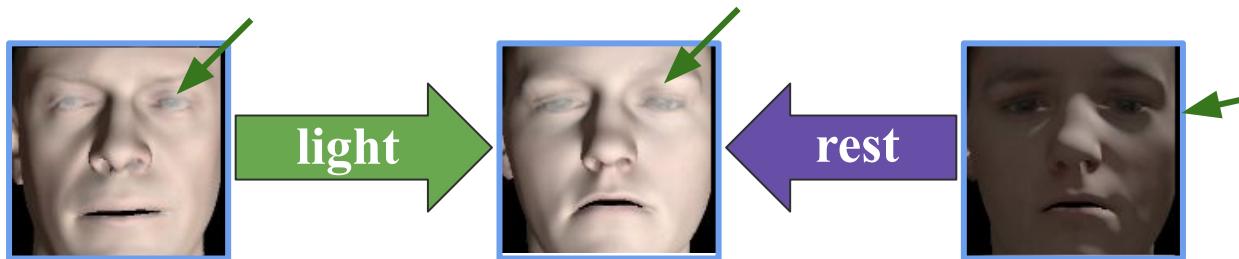
trained exclusively on
synthetic demonstrations
and unlabeled real images.



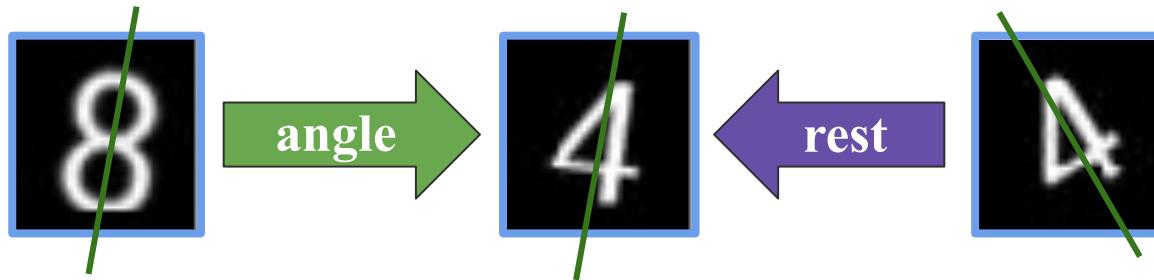
Learning from Cross-Domain Demonstrations: Demo



Learning from Cross-Domain Demonstrations: Demo



Learning from Cross-Domain Demonstrations: Demo



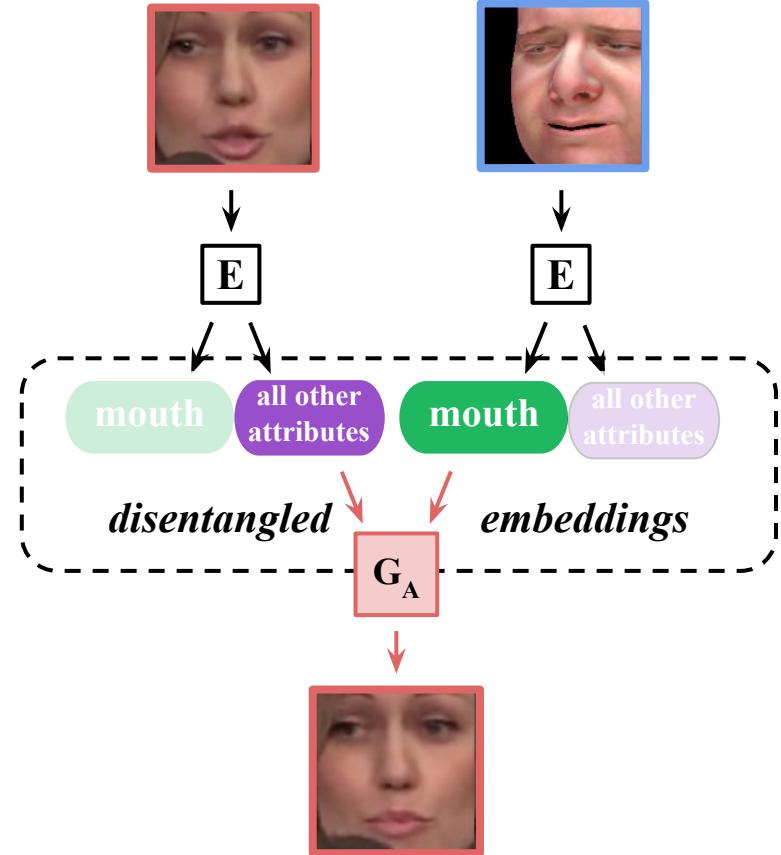
PuppetGAN: Method

Our goal is to train a model that **splits** the embedding into **two parts**:

- **one** to represent the attribute we manipulate (mouth),
- the **other** to represent all other attributes (hair, mic, etc).

↓ ↓ **real decoder**

↓ ↓ **shared encoder**

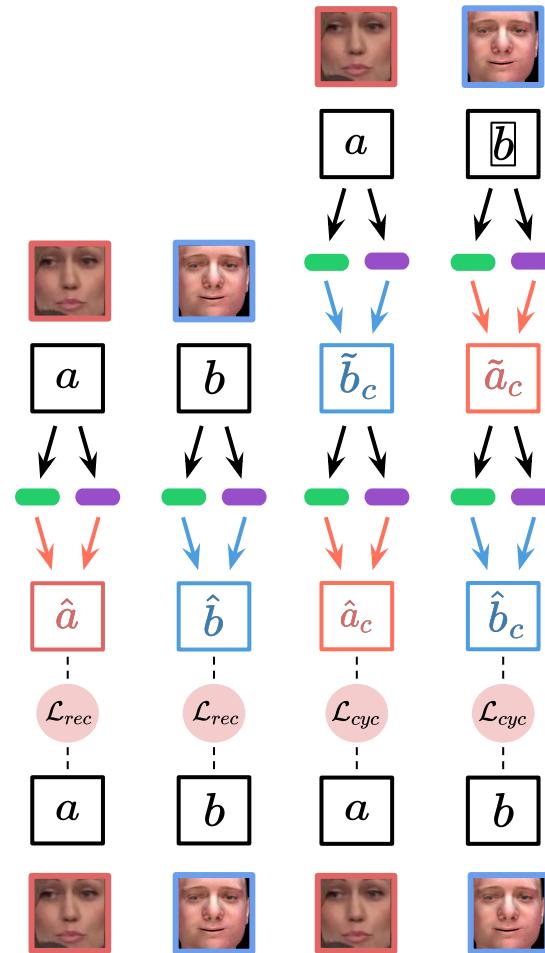


PuppetGAN: Method

We used **autoencoder** and **cycle losses** on both domains.

↓ ↓ **real decoder**
↓ ↓ **synthetic decoder**

↓ ↓ **shared encoder**

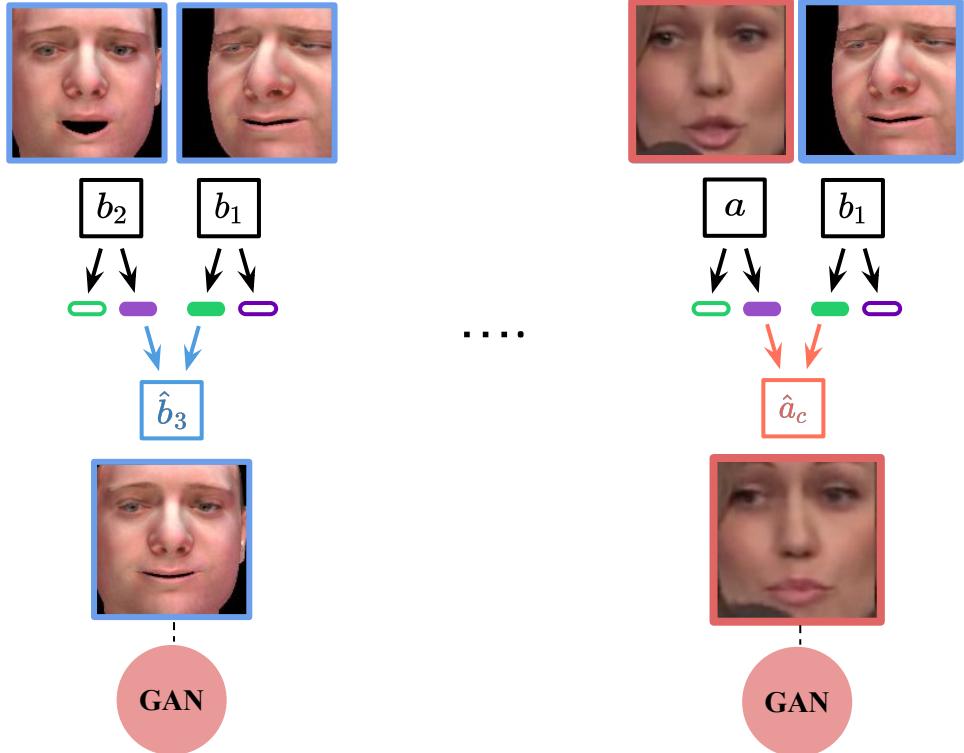


PuppetGAN: Method

And **GAN losses** on all outputs.

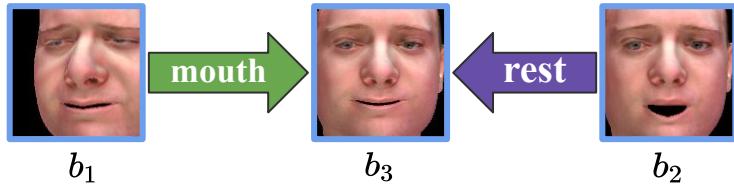
↓ ↓ **real
decoder**
↓ ↓ **synthetic
decoder**

↓ ↓ **shared
encoder**



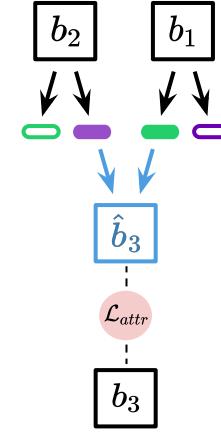
PuppetGAN: Method

We used **supervised losses**
on synthetic data.

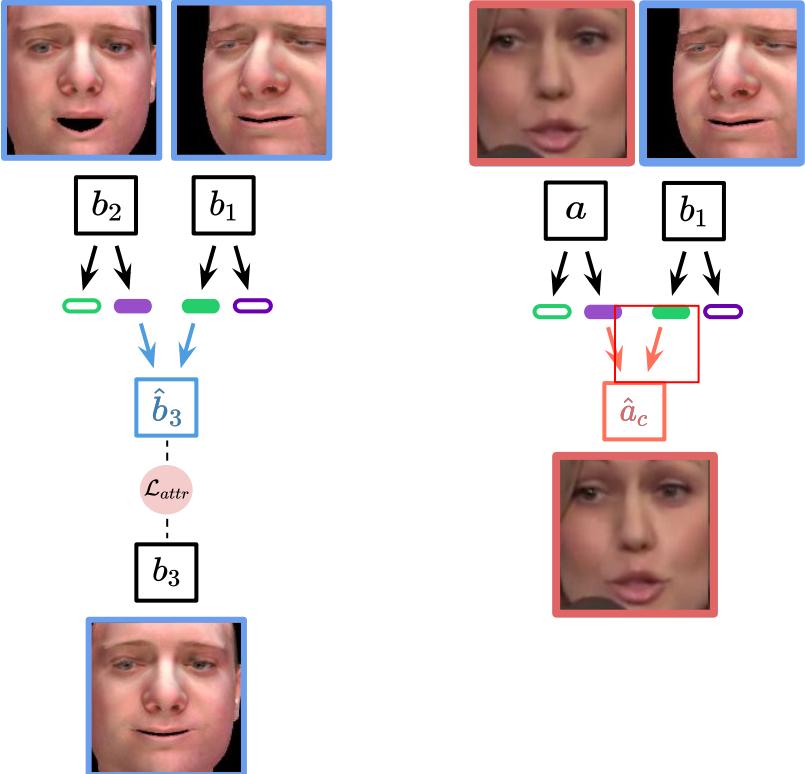


real
decoder
synthetic
decoder

shared
encoder



PuppetGAN: Method



Problem:
The real decoder might **ignore** one input.

real decoder
 synthetic decoder

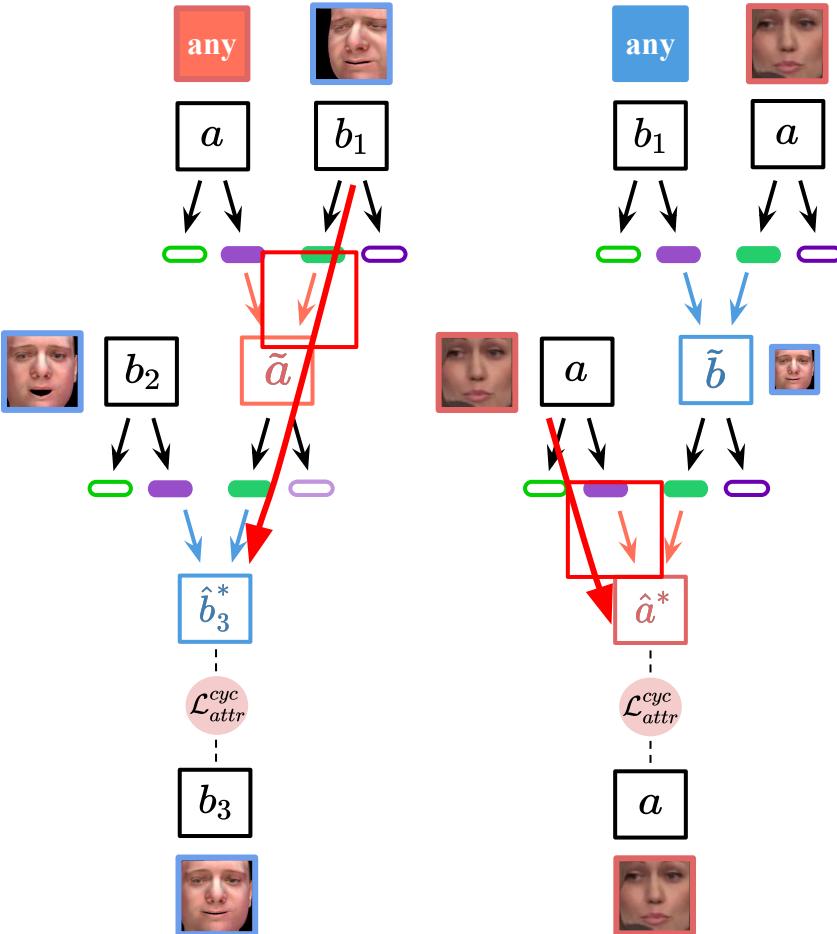
shared encoder

PuppetGAN: Method

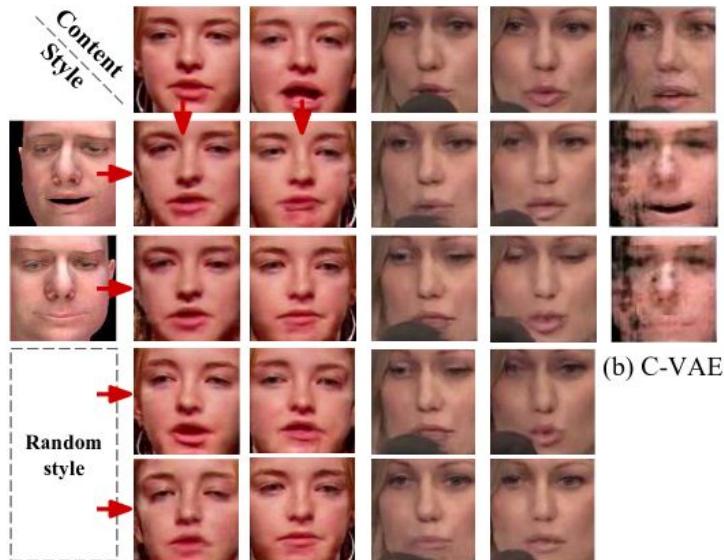
We used **compositional constraint losses** to ensure that **all embeddings** are used.

↓ ↓ real
decoder
synthetic
decoder

↓ ↓ shared
encoder



PuppetGAN: Comparing to related work



Rotation						Size					
attr	6	0	1	2	9	9	6	0	3	6	6
+ rest	4	8	3	7	6	6	1	5	5	0	2
=	6	0	1	2	9	9	1	5	5	0	2

(c) DiDA

5	0	1	8	3	6	3	6	0	8	6	9
1	7	6	1	9	8	1	7	6	1	9	8
5	0	1	3	8	6	3	6	4	8	4	9

(d) MUNIT

5	0	5	1	2	6	5	6	8	8	3	5
0	2	6	3	4	8	4	8	7	6	0	1
5	0	0	3	1	8	9	2	2	9	9	1

(e) Cycle-Consistent VAE

PuppetGAN: Comparing to related work

Model	Disentanglement Quality								Input Domain Discrepancy			
	Size				Rotation				Size		Rot	
	Acc ↑	$r_{\text{attr}}^{\text{syn}} \uparrow$	$r_{\text{rest}}^{\text{syn}} \downarrow$	$V_{\text{rest}} \downarrow$	Acc ↑	$r_{\text{attr}}^{\text{syn}} \uparrow$	$r_{\text{rest}}^{\text{syn}} \downarrow$	$V_{\text{rest}} \downarrow$	$J_{\text{attr}}^{\text{syn}}$	$J_{\text{rest}}^{\text{syn}}$	$J_{\text{attr}}^{\text{syn}}$	$J_{\text{rest}}^{\text{syn}}$
PuppetGAN	0.73	0.85	0.02	0.02	0.97	0.40	0.11	0.01				
CycleGAN [28]	0.10	0.28	0.06	0.28	0.11	0.54	0.37	0.33				
DiDA [2]	0.71	0.18	0.09	0.02	0.86	0.04	0.35	0.02	0.27	0.78	0.05	2.20
MUNIT [10]	0.96	0.06	0.09	0.01	1.00	0.00	0.15	0.01				
Cycle-VAE [8]	0.17	0.92	0.16	0.01	0.29	0.45	0.10	0.01				
PuppetGAN [†]	0.64	0.28	0.07	0.01	0.10	0.06	0.04	0.01	0.90	0.92	0.06	108

PuppetGAN: Takeaway

We can manipulate specific attributes of
real images using supervision from crude
synthetic simulations!

Learning better one-to-one mappings

We can get **stable** alignment dy
dualizing the logistic discriminator!
(ICLR-W'18)

We can get **stable** alignment wrt
powerful discriminator families using
normalizing flows! (NeurIPS20)

Defending models against
performing adversarial attacks **on**
themselves improves semantic
consistency! (NeurIPS19)

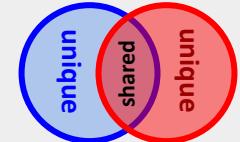
Manipulating factors with cross-domain supervision

We can alter a **single specific attribute** of
real images using **only synthetic**
supervision! (ICCV19 Oral)

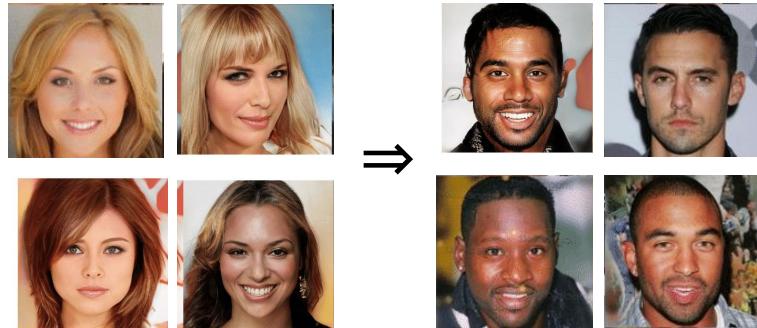


We can manipulate attributes **unique** to
each domain independently from those
shared across domains!

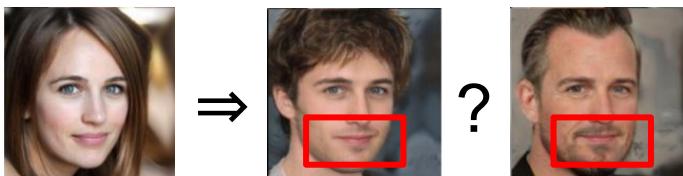
(in submission)



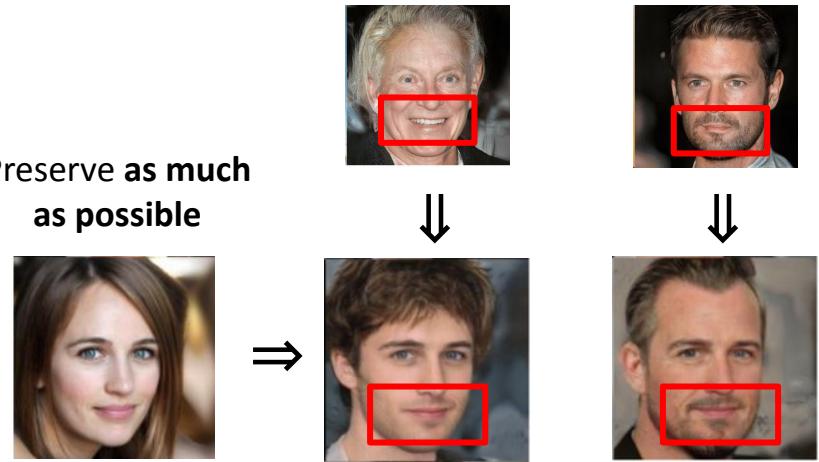
Disentangling Domain-Specific and Domain-Invariant Factors of Variation



1-to-1 alignment problem
is not well defined!



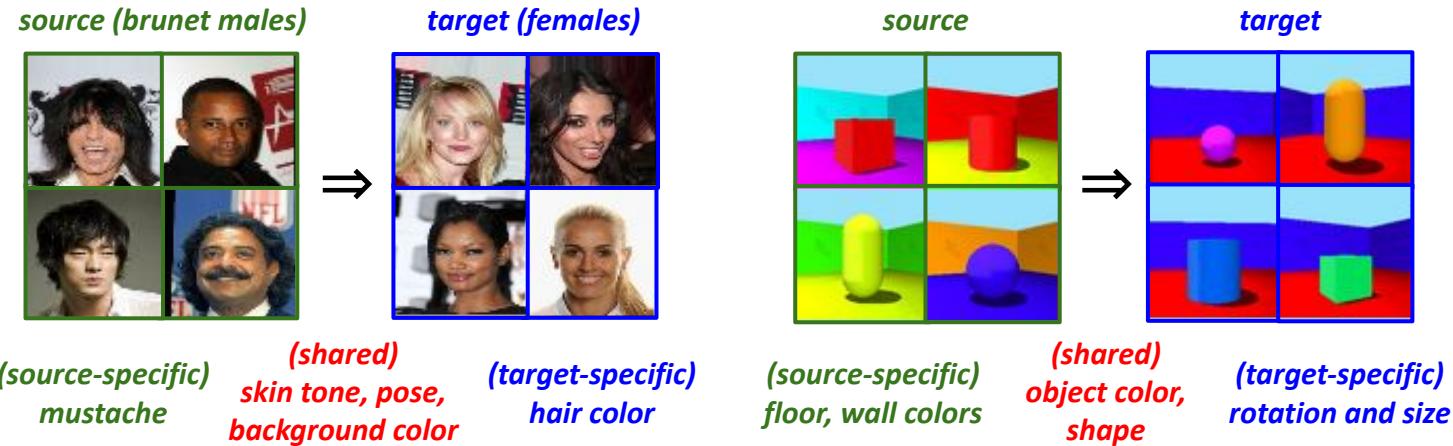
Preserve as much
as possible



many-to-many alignment problem is well defined!

Disentangling Domain-Specific and Domain-Invariant Factors of Variation

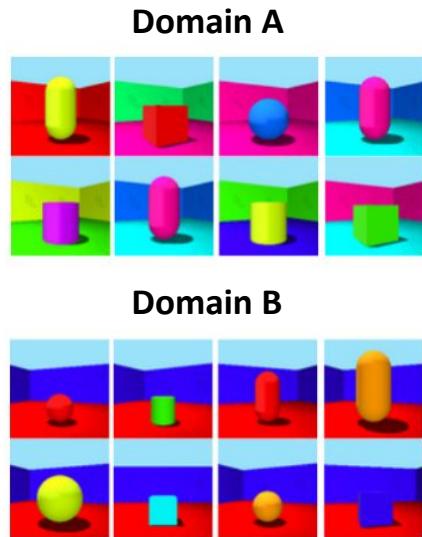
Goal 1: learn which factors of variation are shared vs domain-specific from data



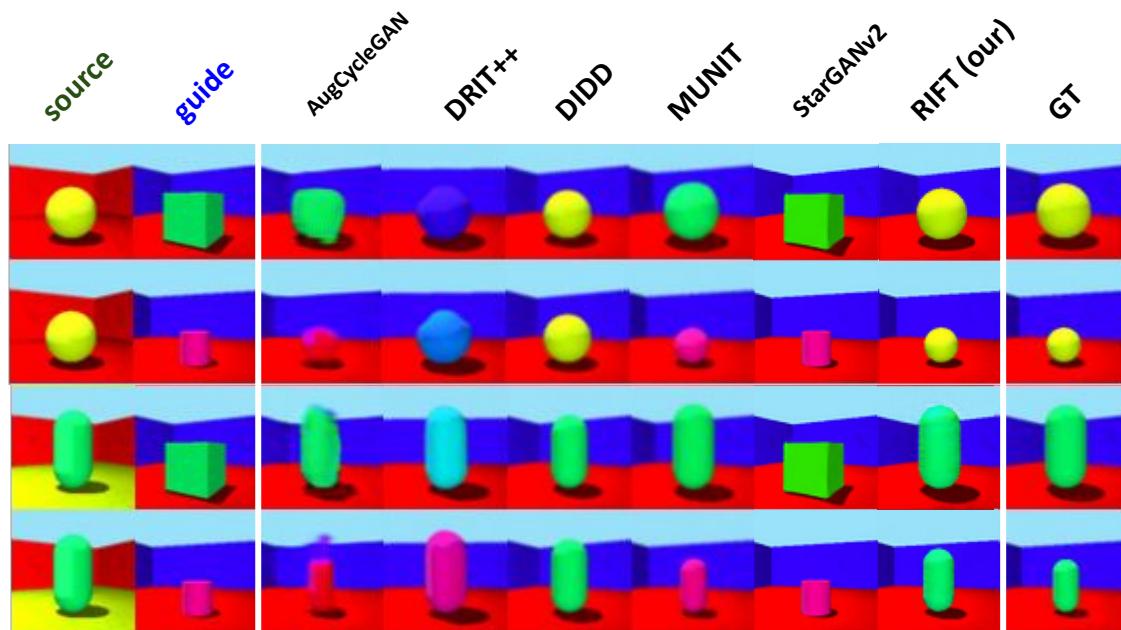
Goal 2: translate a source image to the target domain guided by a target example



Disentangling Domain-Specific and Domain-Invariant Factors of Variation



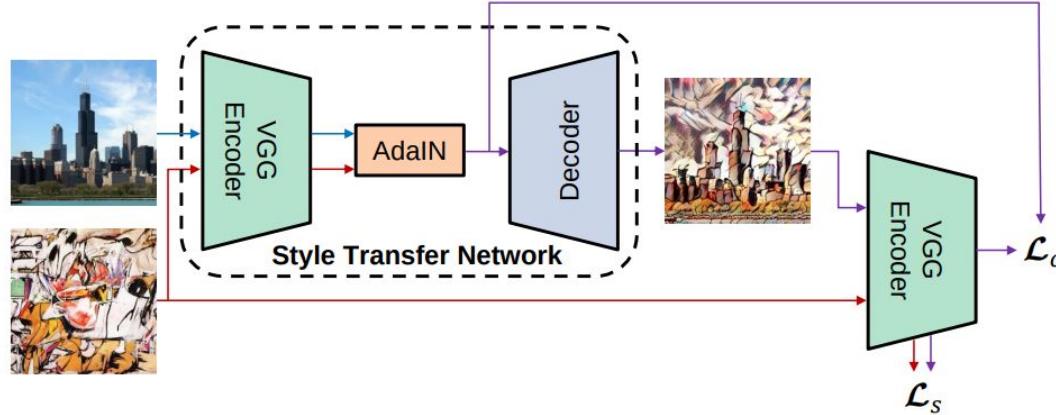
shared: object color, shape
source: floor, wall color
target: size, orientation



“Evaluation of Correctness in Unsupervised Many-to-Many Image Translation” by Bashkirova, Usman, Saenko (WACV22)

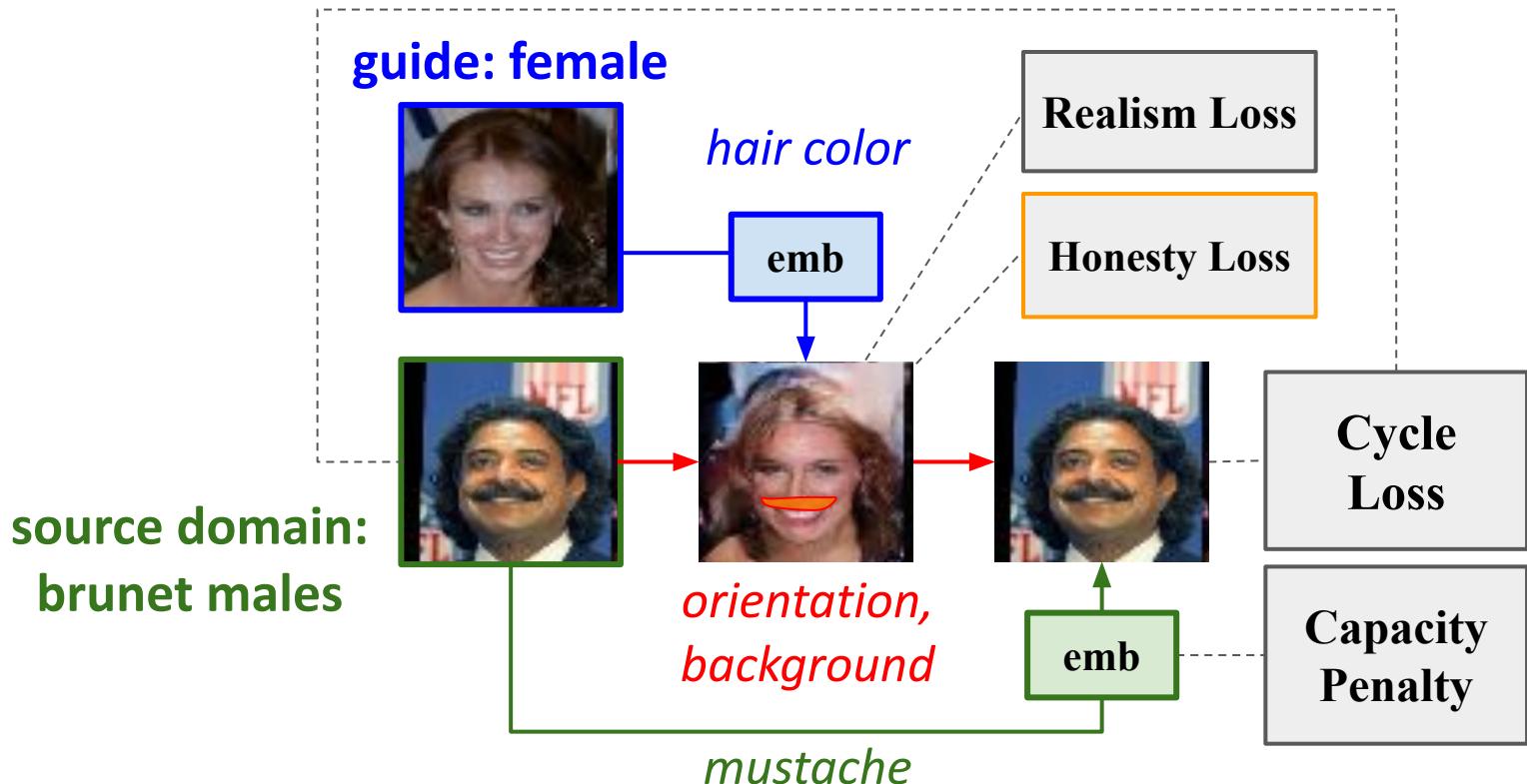
“RIFT: Disentangled Unsupervised Image Translation via Restricted Information Flow” by Usman*, Bashkirova*, Saenko (in submission)

Disentangling Domain-Specific and Domain-Invariant Factors of Variation

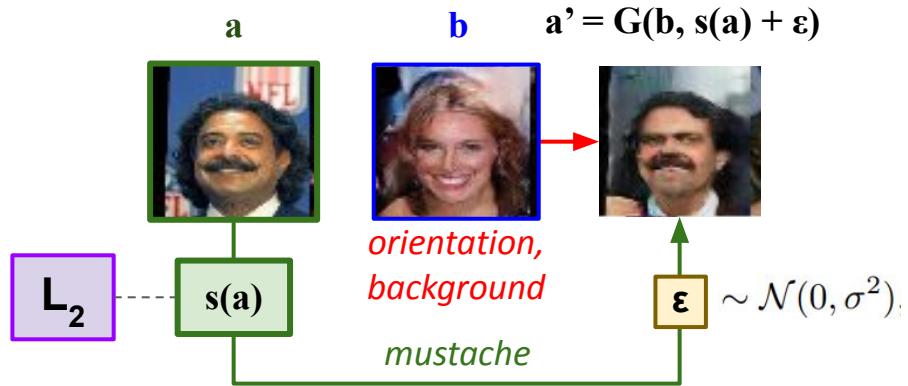


$$\text{AdaIN}(x, y) = \sigma(y) \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y)$$

RIFT: Translation via Restricted Information Flow



Capacity Loss



Theorem 1. *The effective capacity of the guided embedding, i.e. the capacity of the $a \rightarrow a'$ channel, i.e. the mutual information $MI(a; a')$ is bounded by:*

$$MI(a; a') \lesssim \dim(s(a)) \cdot \log_2 (1 + L/\sigma^2),$$

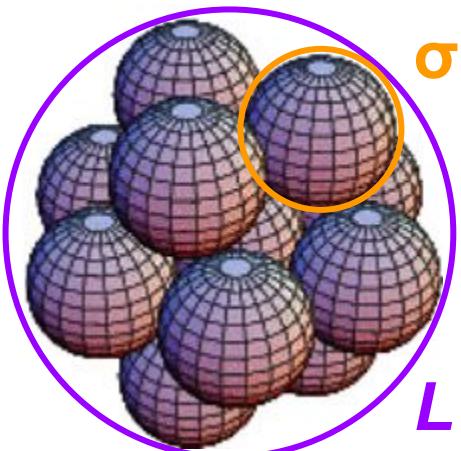
where $a' = G(b, s(a) + \varepsilon)$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$,

and $L = \mathbb{E}\|s(a)\|_2^2$, $a \sim A$, $b \sim B$

Corollary:

$L \rightarrow \infty \Rightarrow MI \rightarrow \infty$

$\sigma \rightarrow 0 \Rightarrow MI \rightarrow \infty$



Capacity Loss

Theorem 1. *The effective capacity of the guided embedding, i.e. the capacity of the $a \rightarrow a'$ channel, i.e. the mutual information $\text{MI}(a; a')$ is bounded by:*

$$\begin{aligned} \text{MI}(a; a') &\lesssim \dim(s(a)) \cdot \log_2 (1 + L/\sigma^2), \\ \text{where } a' &= G(b, s(a) + \varepsilon), \quad \varepsilon \sim \mathcal{N}(0, \sigma^2), \\ \text{and } L &= \mathbb{E}\|s(a)\|_2^2, \quad a \sim A, \quad b \sim B \end{aligned}$$

Proof. Applying the data processing inequality

$$X \rightarrow Y \rightarrow Z \Rightarrow \text{MI}(X; Z) \leq \text{MI}(X; Y) \wedge \text{MI}(X; Z) \leq \text{MI}(Y; Z)$$

twice to following Markov chains

$$a \rightarrow (s(a) + \varepsilon) \rightarrow a', \quad a \rightarrow s(a) \rightarrow (s(a) + \varepsilon)$$

gives us

$$\text{MI}(a; a') \leq \text{MI}(a; s(a) + \varepsilon) \leq \text{MI}(s(a); s(a) + \varepsilon)$$

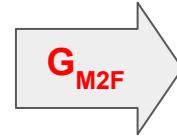
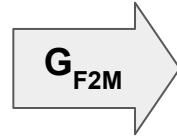
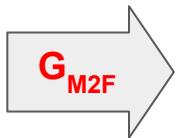
intuitively meaning that the overall pipeline always loses at least as much information as each of its steps. Then expanding the mutual information in terms of the differential entropy $h(X)$ gives us

$$\begin{aligned} \text{MI}(s(a); s(a) + \varepsilon) &= h(s(a) + \varepsilon) - h(s(a) + \varepsilon | s(a)) \\ &= h(s(a) + \varepsilon) - h(\varepsilon) \end{aligned}$$

Since the the second raw moment (aka power) of $s(a)$ is bounded by L , the entropy $h(s(a) + \varepsilon)$ will be maximized if $s(a)$ is a k -dimensional spherical multivariate normal with variance L , where $k = \dim(s(a))$ therefore

$$\begin{aligned} \text{MI}(s(a); s(a) + \varepsilon) &\leq h(\mathcal{N}_k(0; L + \sigma^2)) + h(\mathcal{N}_k(0; \sigma^2)) \\ &= \frac{1}{2} \ln \left(\frac{(L + \sigma^2)^k}{\sigma^{2k}} \right) \leq k \cdot \log_2 (1 + L/\sigma^2). \end{aligned}$$

Honesty Loss

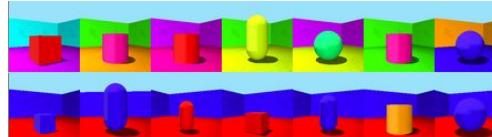


...



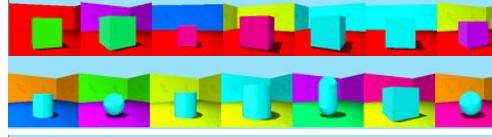
Datasets

Shapes-3D-A



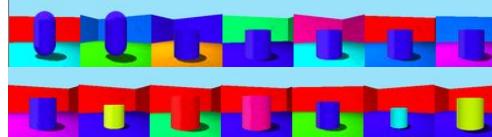
shared: object color, shape
source: floor, wall color
target: size, orientation

Shapes-3D-B



shared: wall color, size
source: object color, orient.
target: shape, floor color

Shapes-3D-C



shared: floor color, orient.
source: wall color, shape
target: size, object color

SynAction



shared: pose
source: background
target: identity/clothing

CelebA



shared: pose, background
source: hair color
target: facial hair

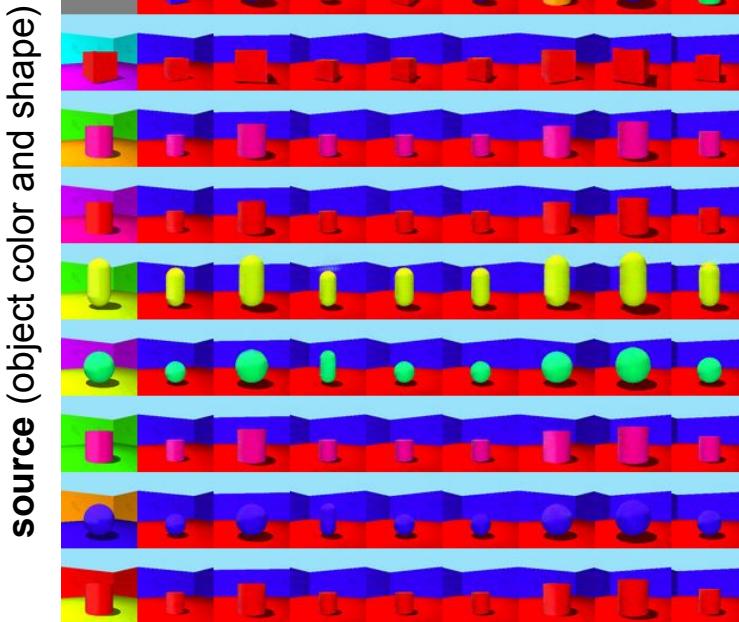
Results

Shapes-3D-A

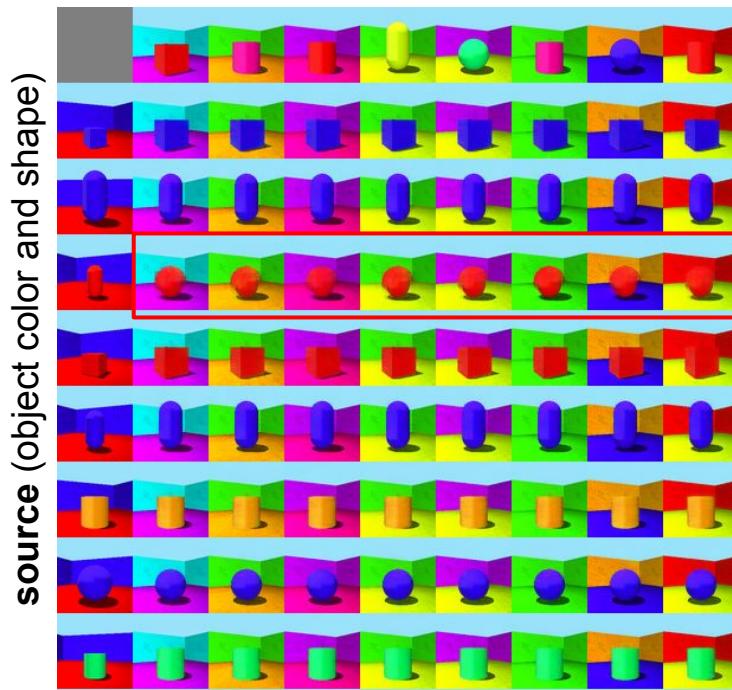


shared: object color, shape
source: floor, wall color
target: size, orientation

guide (rotation and size)



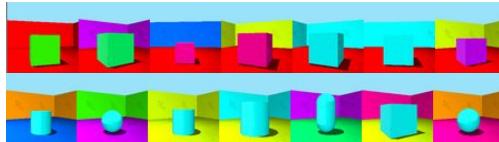
guide (floor and wall color)



source (object color and shape)

Results

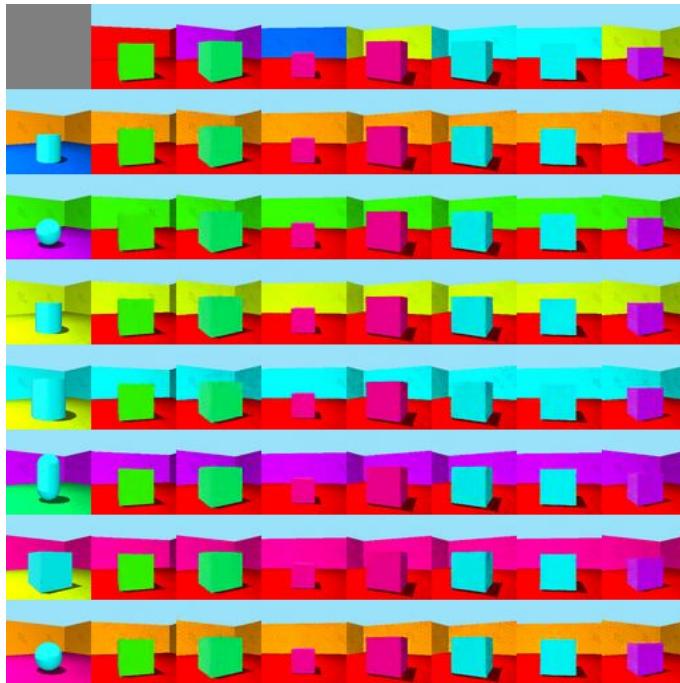
Shapes-3D-B



shared: wall color, size
source: object color, orient.
target: shape, floor color

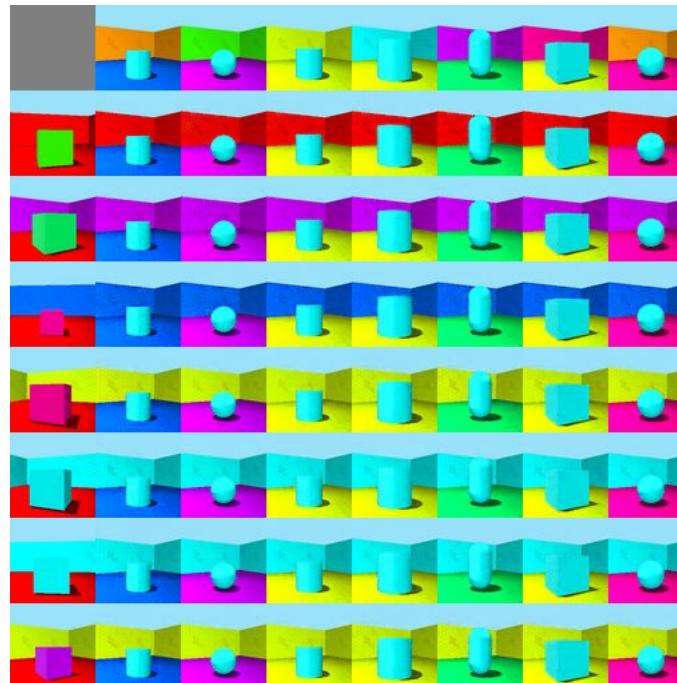
guide (object color and orientation)

source (wall color and size)



guide (floor color and shape)

source (wall color and size)



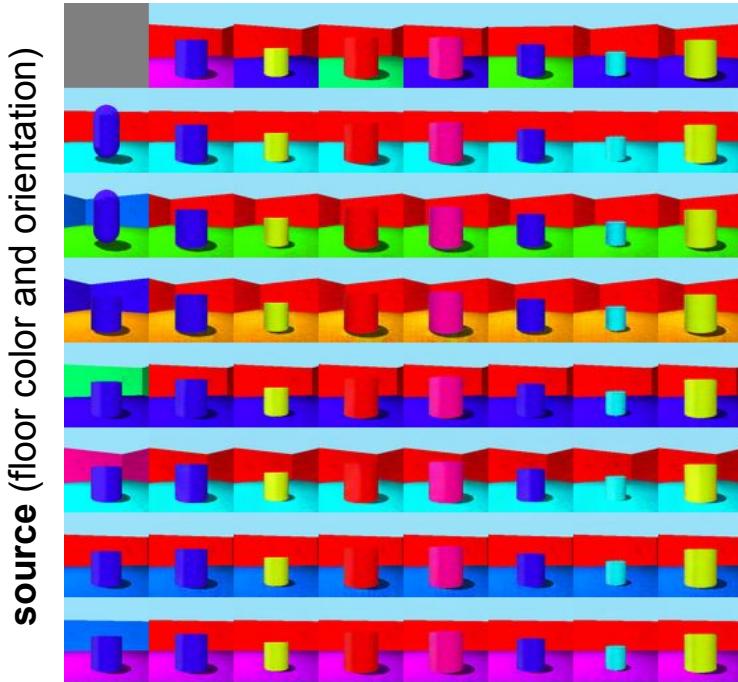
Results

Shapes-3D-C

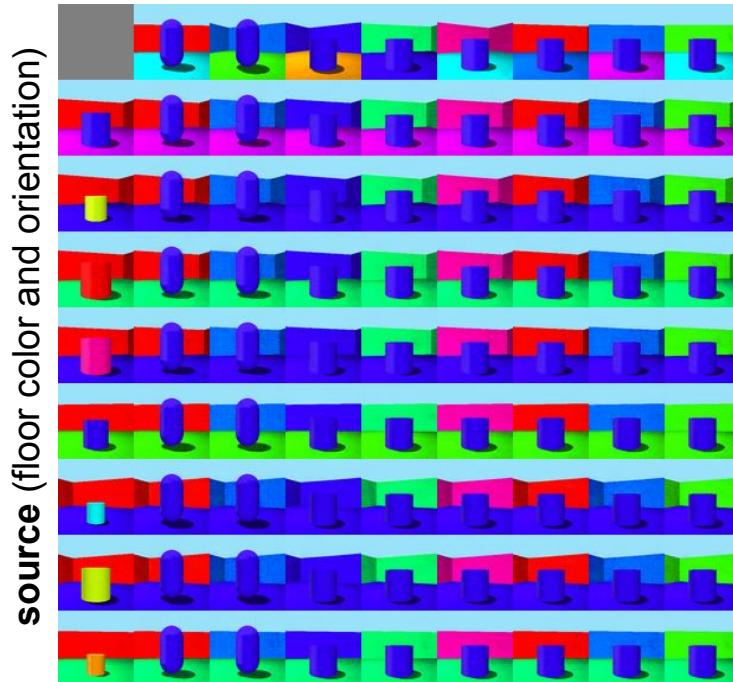


shared: floor color, orient.
source: wall color, shape
target: size, object color

guide (size and object color)



guide (shape and wall color)

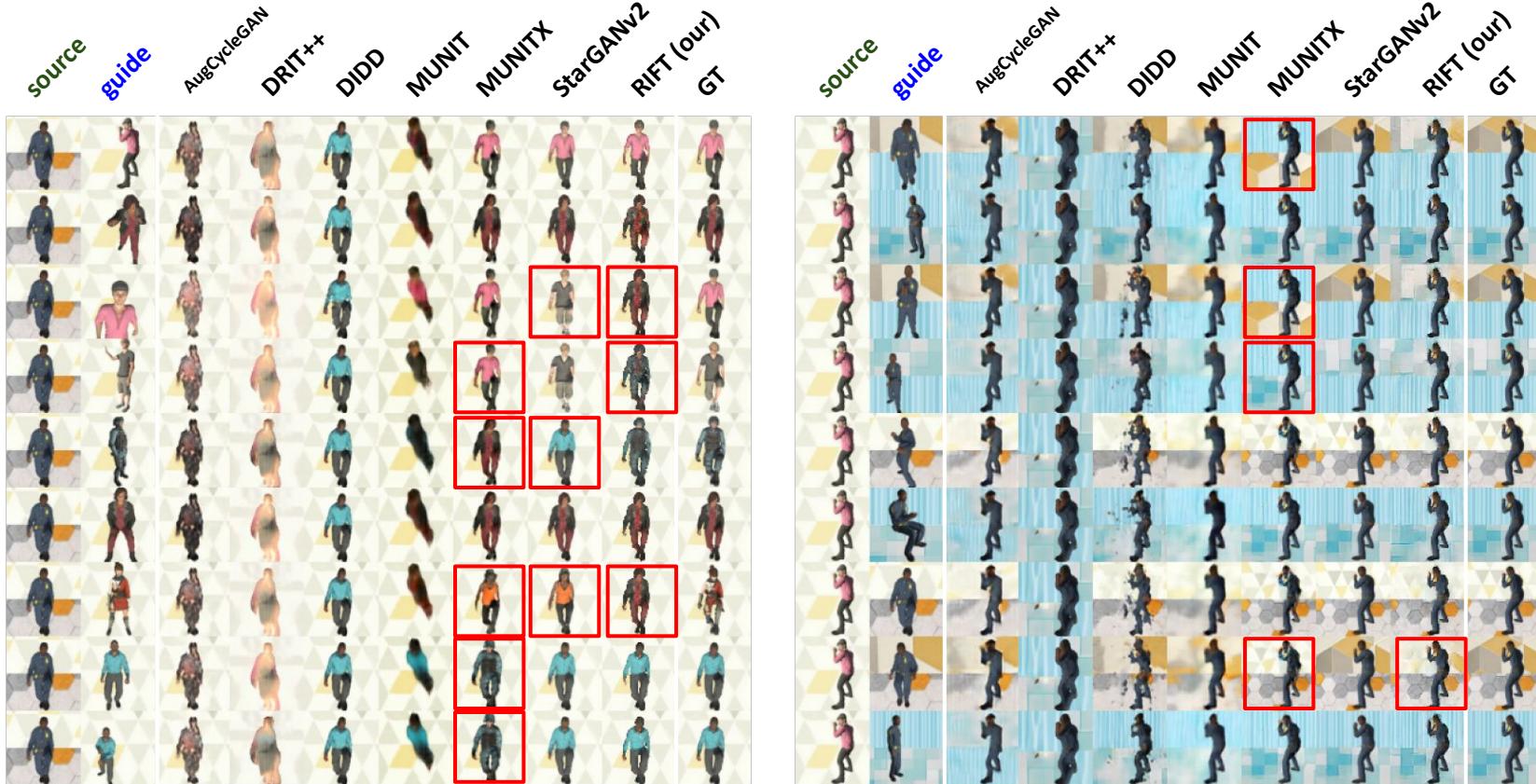


Results

SynAction



shared: pose
source: background
target: identity/clothing

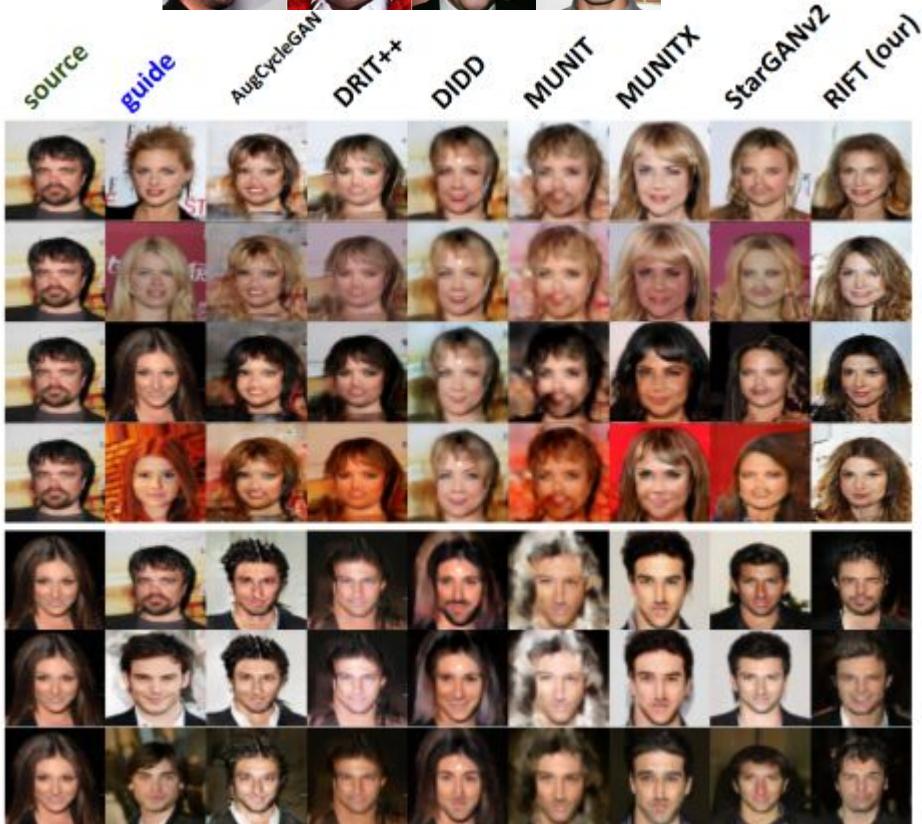


Results

CelebA



shared: pose, background
source: hair color
target: facial hair



Metrics and Qualitative Results

Manipulation Accuracy (for categorical):

$$\text{ACC}_k^A = p(f_k(F_{A2B}(a, b)) = y_k^* \mid f_k(a) \neq f_k(b))$$

where the “correct” attribute value equals $y_k^* = f_k(a)$ for shared attributes, and $y_k^* = f_k(b)$ otherwise. For real-

Manipulation Accuracy (for real-valued):

$$\text{ACC}_k^A = p(\|f_k(F_{A2B}(a, b)) - y_k^*\| \leq \|f_k(F_{A2B}(a, b)) - y'_k\|)$$

where $y_k^* = f_k(a)$ and $y'_k = f_k(b)$ for shared attributes, and vice-versa otherwise.

Relative Discrepancy (for shapes only):

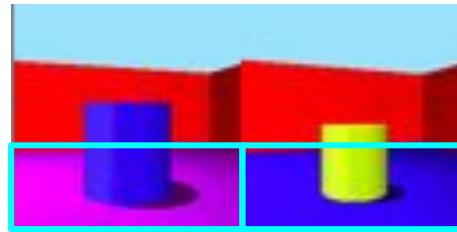
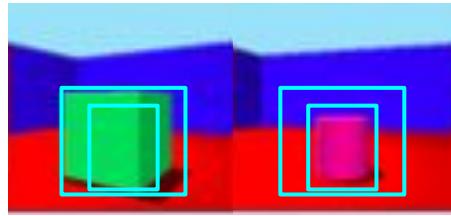
$$\text{RD} = 100 \cdot \frac{\sum_k |\text{ACC}_k^S - \text{ACC}_k^C|}{\sum_k (\text{ACC}_k^S + \text{ACC}_k^C)}.$$

Method	3DS	SA	CA	AVG	RD
StarGANv2	45	82	51	59	97
MUNIT	58	37	53	49	56
MUNITX	33	52	55	47	74
DRIT++	18	24	55	32	20
AugCycleGAN	12	37	40	29	20
DIDD	44	67	64	58	35
RIFT (ours)	88	<u>78</u>	<u>60</u>	75	6
RAND	12	24	49	27	9

Table 1: Average (AVG↑) manipulation accuracy (ACC) and relative discrepancy (RD↓) across 3D-Shapes-ABC (3DS), SynAction (SA), and CelebA-FM (CA). Notation: **best**, 2nd best.

Remaining Challenges

1. How to deal with attributes that “occupy” very different number of pixels in reconstruction losses (e.g. size vs color)?

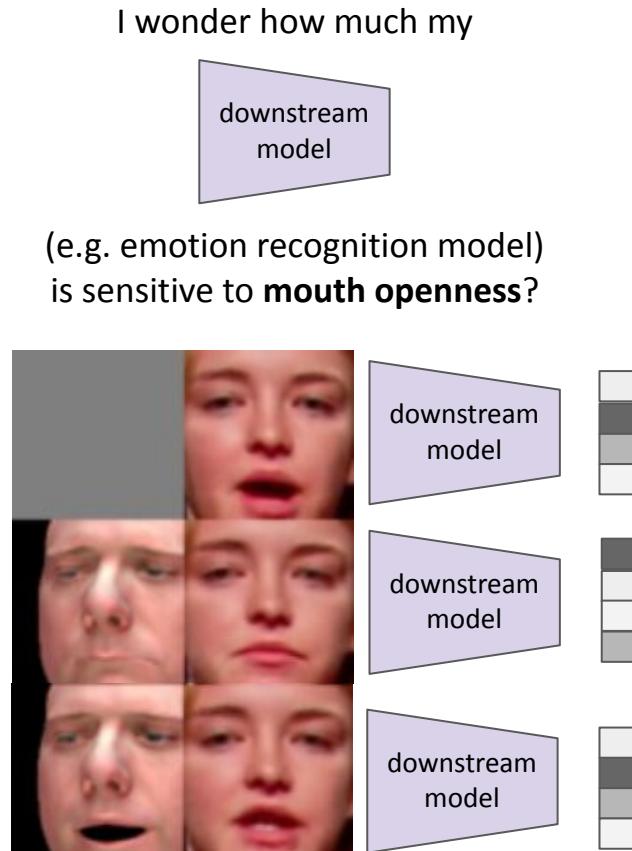
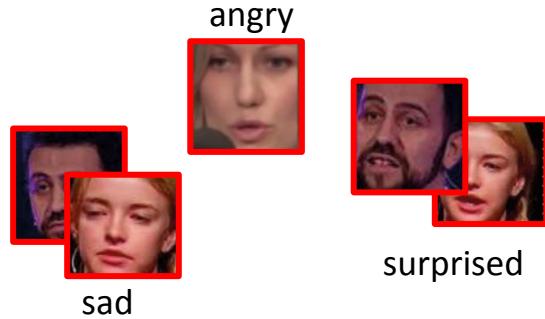
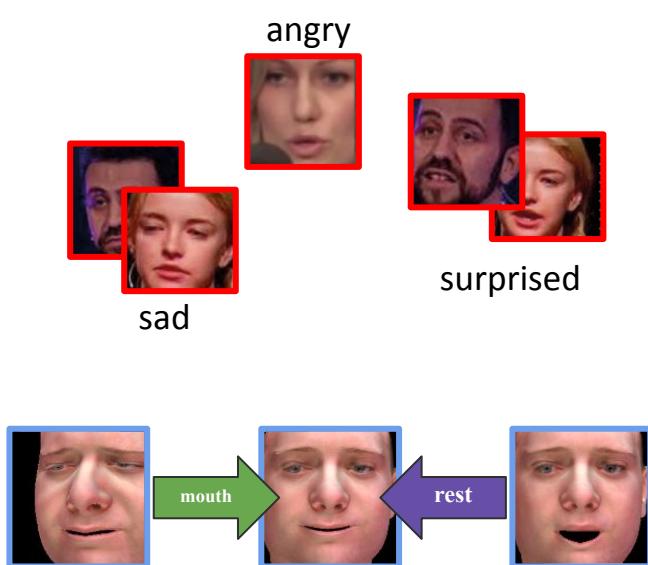


2. What if attributes are varied in both but have different distributions?
(e.g. 3% females are blonde, but 50% of males are blonde)

Takeaway

We can use unsupervised alignment to
discover domain-specific factors of
variability without any supervision!

Applications: Interpretability and Control



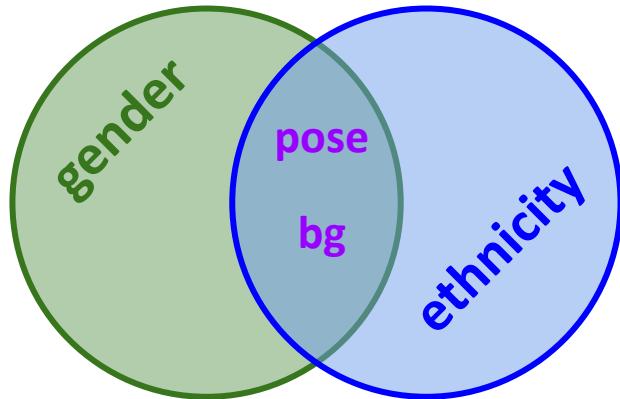
Applications: Interpretability and Control



Train

factors of variability

Test



*shared
from
source*

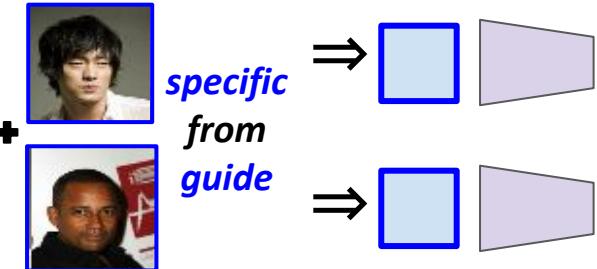


downstream
model

I wonder whether my

is sensitive to factors of variability
absent in train, but **present** in test?

Train RIFT!



Applications: Interpretability and Control

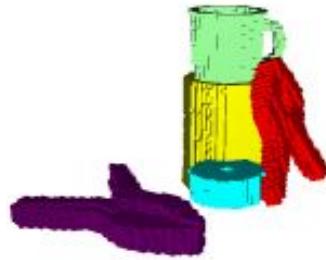


texture

RIFT

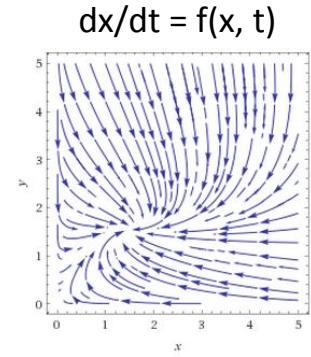


LRMF + ?



size
location
shape

PuppetGAN



Learning better one-to-one mappings

We can get **stable** alignment dy
dualizing the logistic discriminator!
(ICLR-W'18)

We can get **stable** alignment wrt
powerful discriminator families using
normalizing flows! (NeurIPS20)

Defending models against
performing adversarial attacks **on**
themselves improves semantic
consistency! (NeurIPS19)

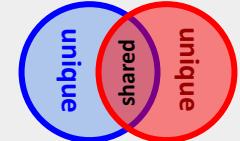
Manipulating factors with cross-domain supervision

We can alter a **single specific attribute** of
real images using **only synthetic**
supervision! (ICCV19 Oral)



We can manipulate attributes **unique** to
each domain independently from those
shared across domains!

(in submission)



Thank you for your attention!

Questions?