Deep Learning Theory and Applications

Generative Adversarial Networks (GANs)

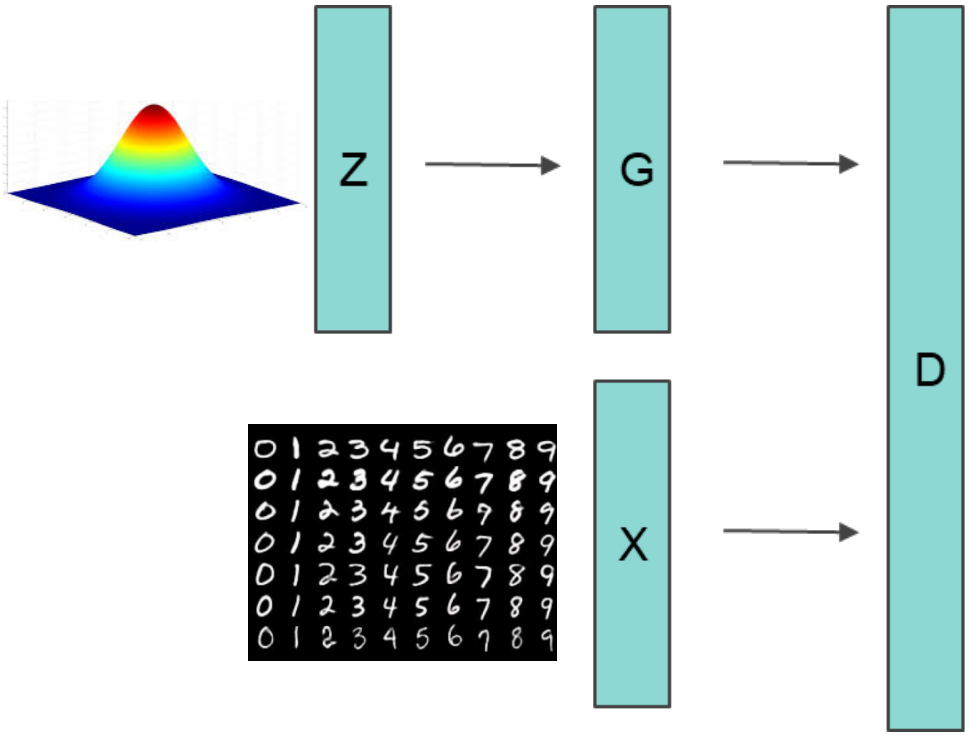
Yale

CPSC/AMTH 663

Outline

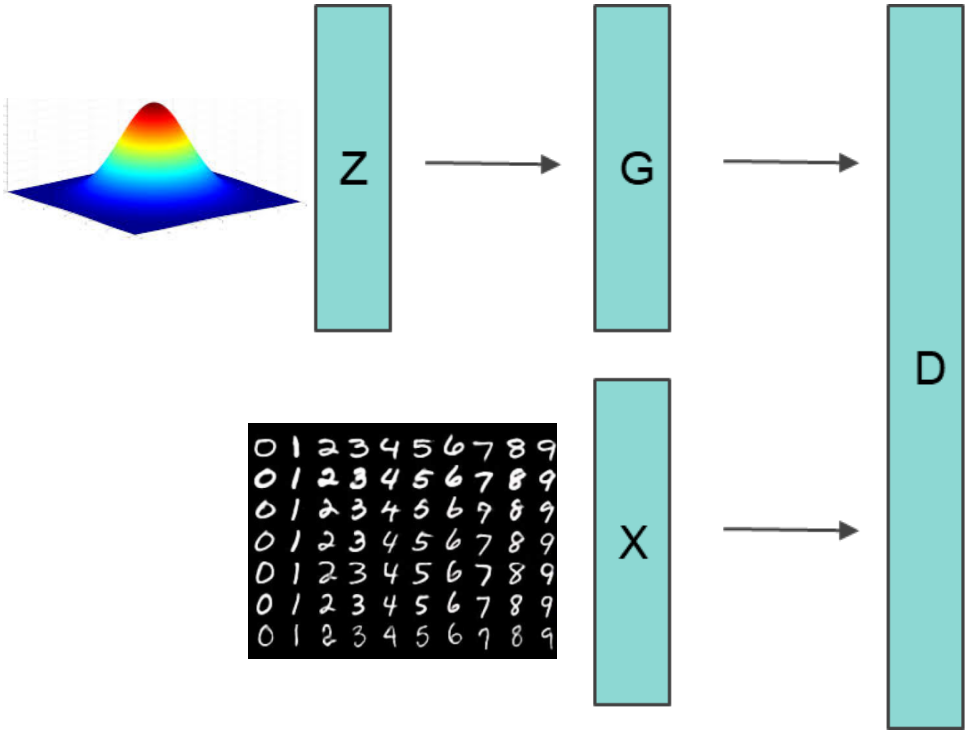
1. GANs
2. Conditional GANs
3. Autoencoder GANs
4. DiscoGANs
5. Cycle GANs
6. MAGAN
7. TraVeLGAN

GANs



- Sample from **Z**
  - Easy to do!
- One network learns a mapping G(**Z**)
- Second network tries to distinguish true examples of **X** from “fake” samples G(**Z**)

GANs



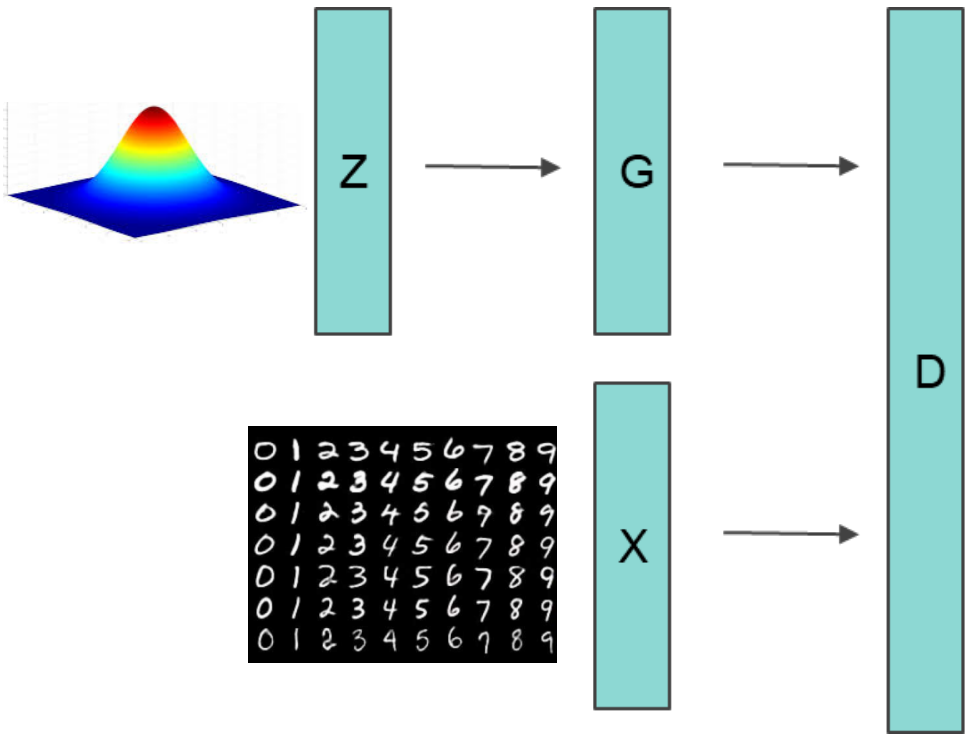
- After training, we throw out D
- Sample from **Z**
- Compute G(**Z**)
- New samples from an estimate of P(**X**) (not **X**)!
- Note that the GAN did not just memorize the finite data set, as the generated samples do not appear exactly in **X**

G(**Z**)

Closest **x** to G(**Z**)

Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.

GANs



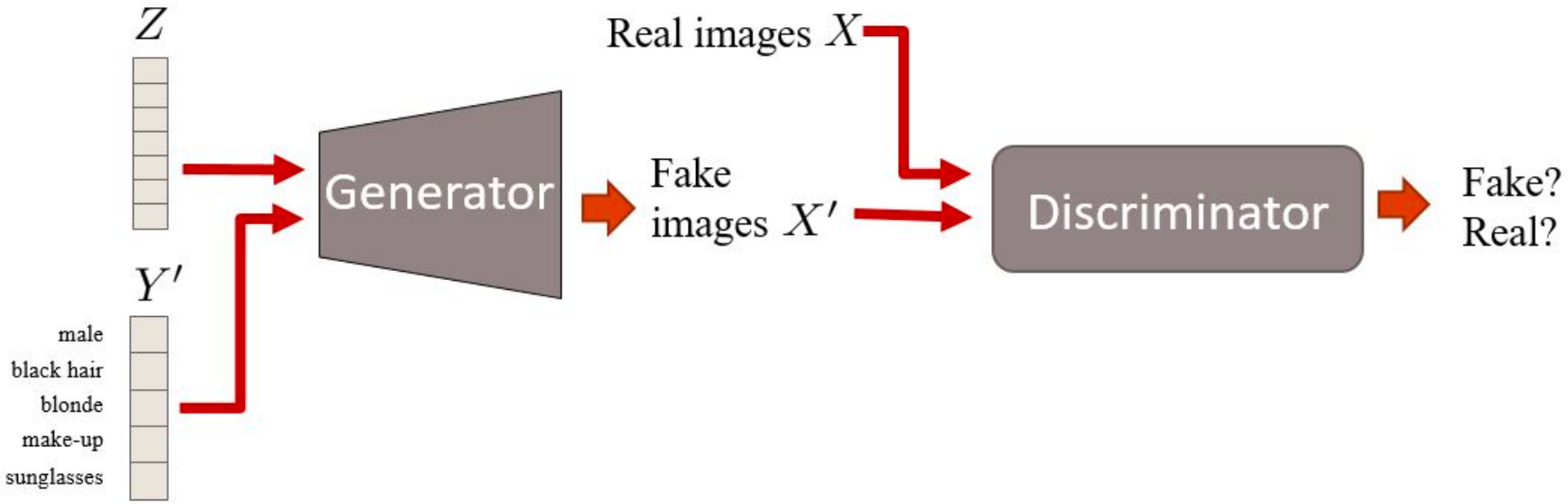
- Training alternates between D and G
- Discriminator tries to classify true samples as 1 and fake samples as 0
- Generator tries to fool the discriminator
- Gradients flow in between the networks

Conditional GANs

- GANs take noise z and turn it into our data x
- Great! But what if we want to control it?
  - “It generated ***another*** image of a 3? I wanted a 7 this time!”
- Conditional GANs:
  - If we have labels, we want to teach it to generate not just any x, but an x of a particular label when given some noise z

Conditional GANs

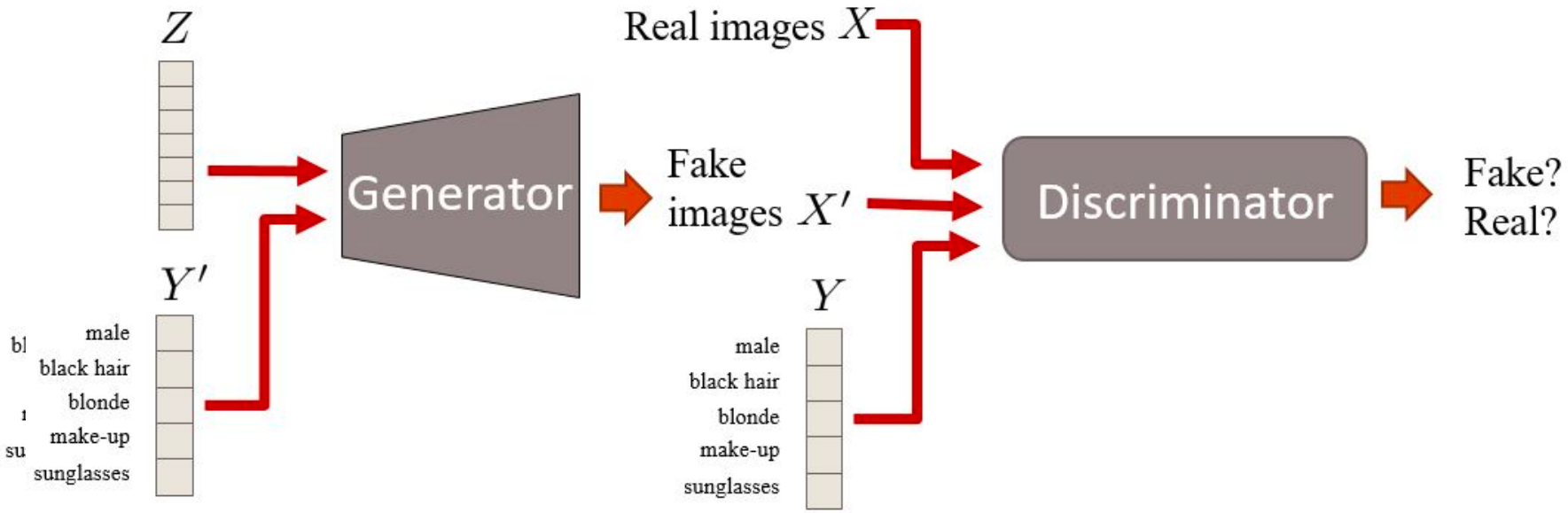
- Feed the generator both a sample from z and the condition
- Discriminator tries to tell real/fake apart as before
  - The generator could generate really realistic images that don't match the condition and the discriminator would never know



http://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them

Conditional GANs

- Have to also give the discriminator the labels for each point
  - The discriminator not only looks to see if it is realistic, but also if it is an image/condition pair that doesn't appear in real data

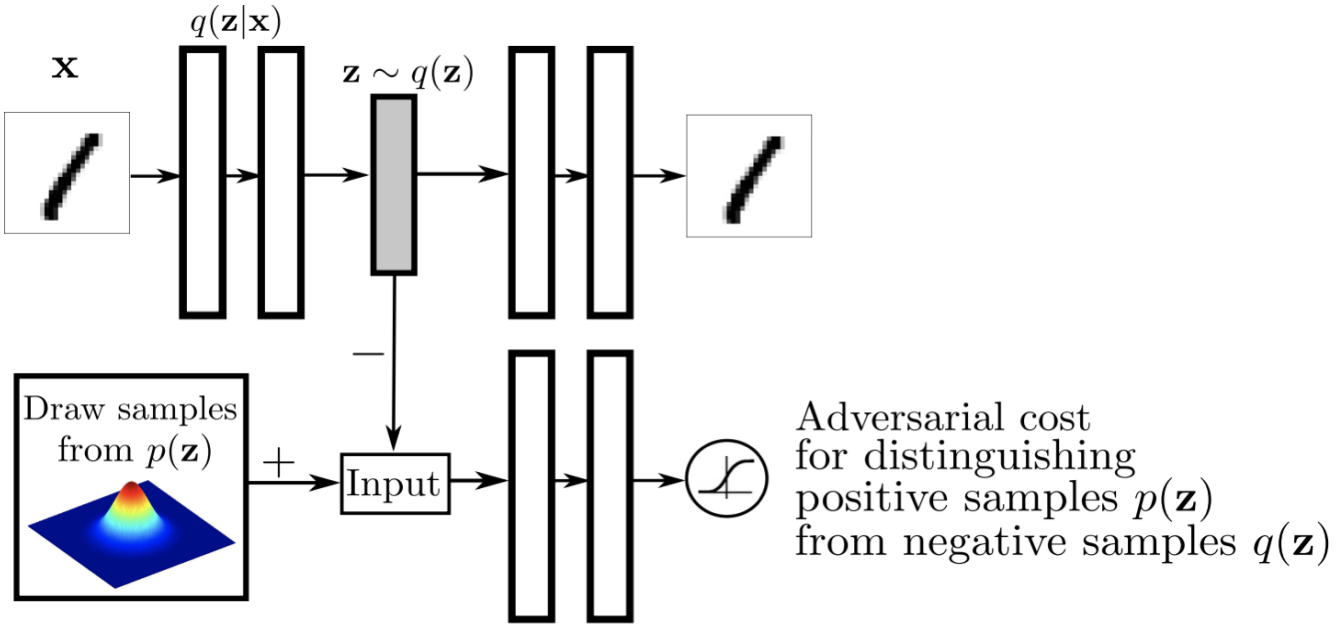


http://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them

Can adversarial training be used on Autoencoders ?

- Can we use it to make an improved version of a VAE?
- Can the role of the KL-divergence penalty be replaced by an advesary?

Adversarial Autoencoders

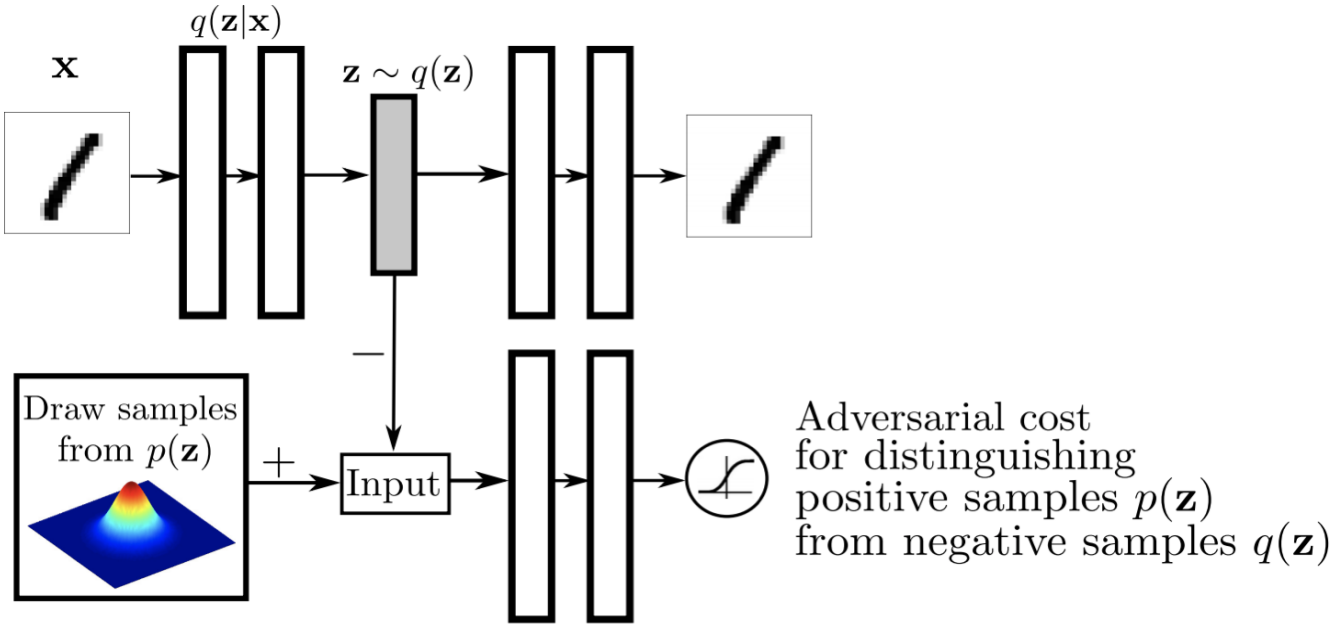


Differences

1. Discriminator classifies samples from **z** against the AE hidden layer
   a) Low-dimensional and easy distributions
2. Generator has both encoder and decoder
3. Generator receives gradient signal from discriminator **and** reconstruction loss

Makhzani, Alireza, et al. "Adversarial autoencoders." *arXiv preprint arXiv:1511.05644* (2015).

Adversarial Autoencoders



- Generate new sample as follows:
  1. Sample from $p(\boldsymbol{z})$
  2. Decode $\boldsymbol{z}$
- Encoder and discriminator only needed for training, not sampling

Makhzani, Alireza, et al. "Adversarial autoencoders." *arXiv preprint arXiv:1511.05644* (2015).

Other advantages of AANs

- The discriminator has a simpler job: just needs to distinguish between latent layer distribution and gaussian distribution

- Has denoising and other advantages of an autoencoder

DiscoGAN: Motivation

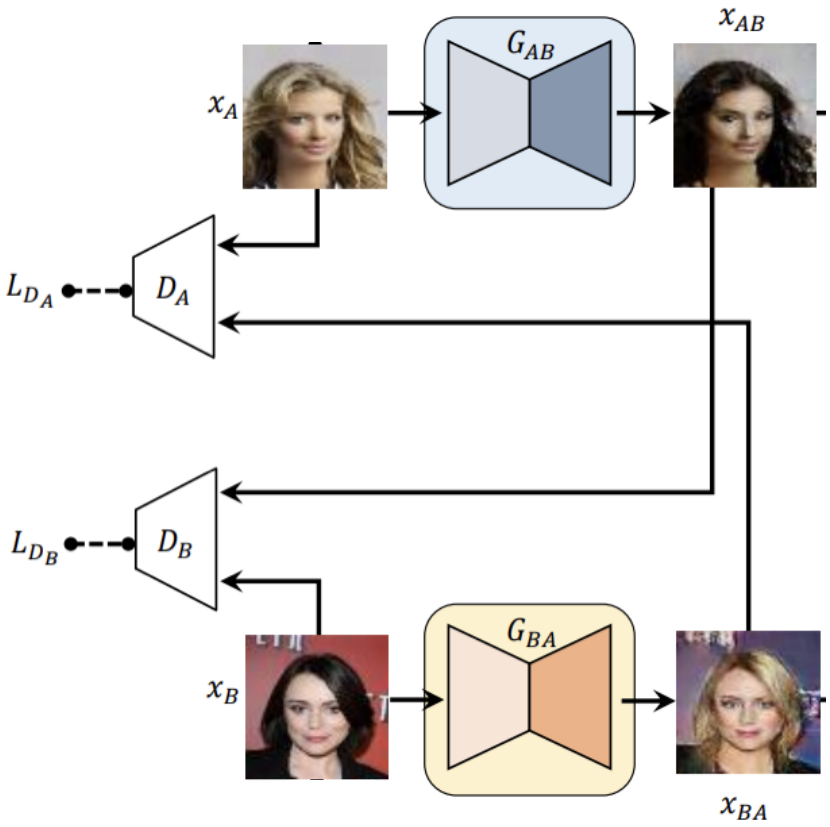
- GANs and adversarial AEs sample from one domain
- What if we have two domains, A and B?
- Can we learn to turn a point in A into a point that looks like its from B?
- Other attempts required paired input (a_i, b_i)
  - Uses are limited since labels can be expensive or difficult to obtain
- Can we do this with an unsupervised training setup?

DiscoGAN: Motivation

- Domain A: blond hair

- Domain B: black hair

Previous approach

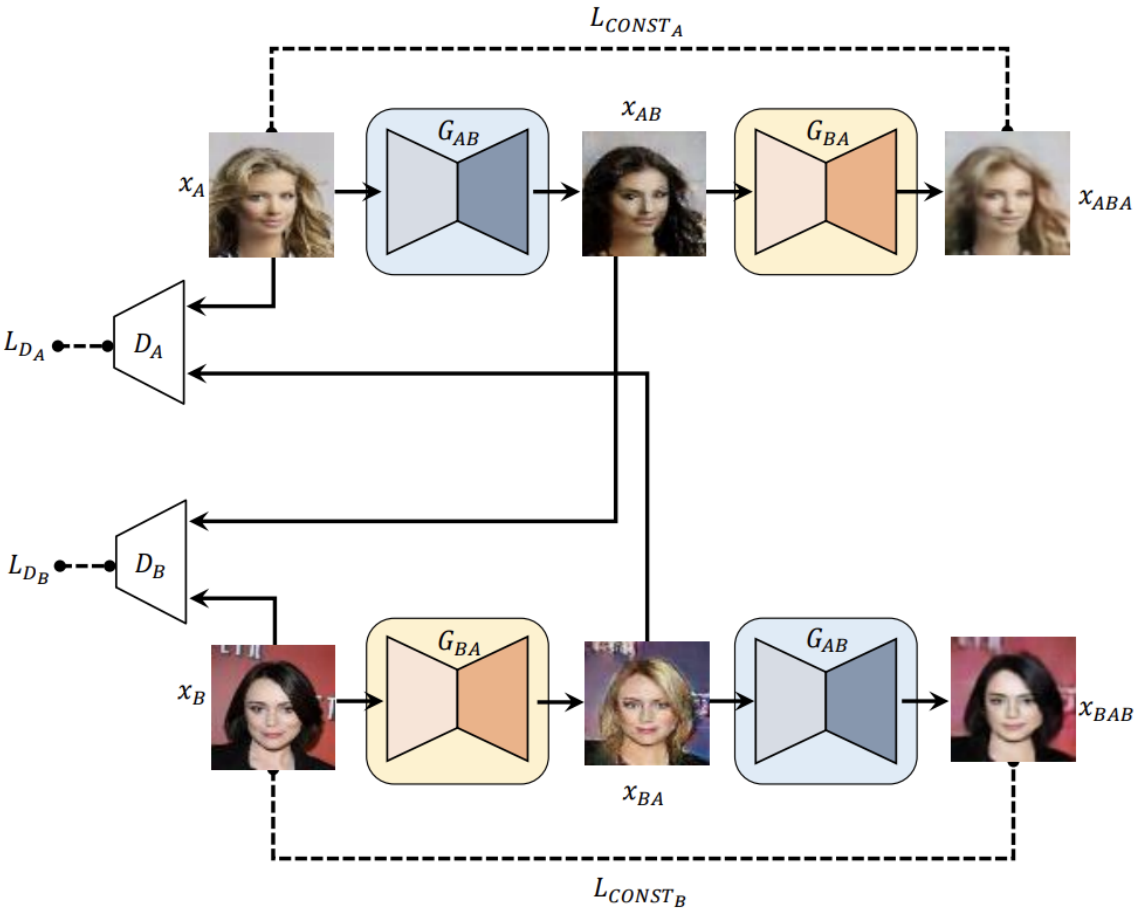


Kim, Taeksoo, et al. "Learning to discover cross-domain relations with generative adversarial networks." *arXiv preprint arXiv:1703.05192* (2017).

DiscoGAN

- Domain A: blond hair

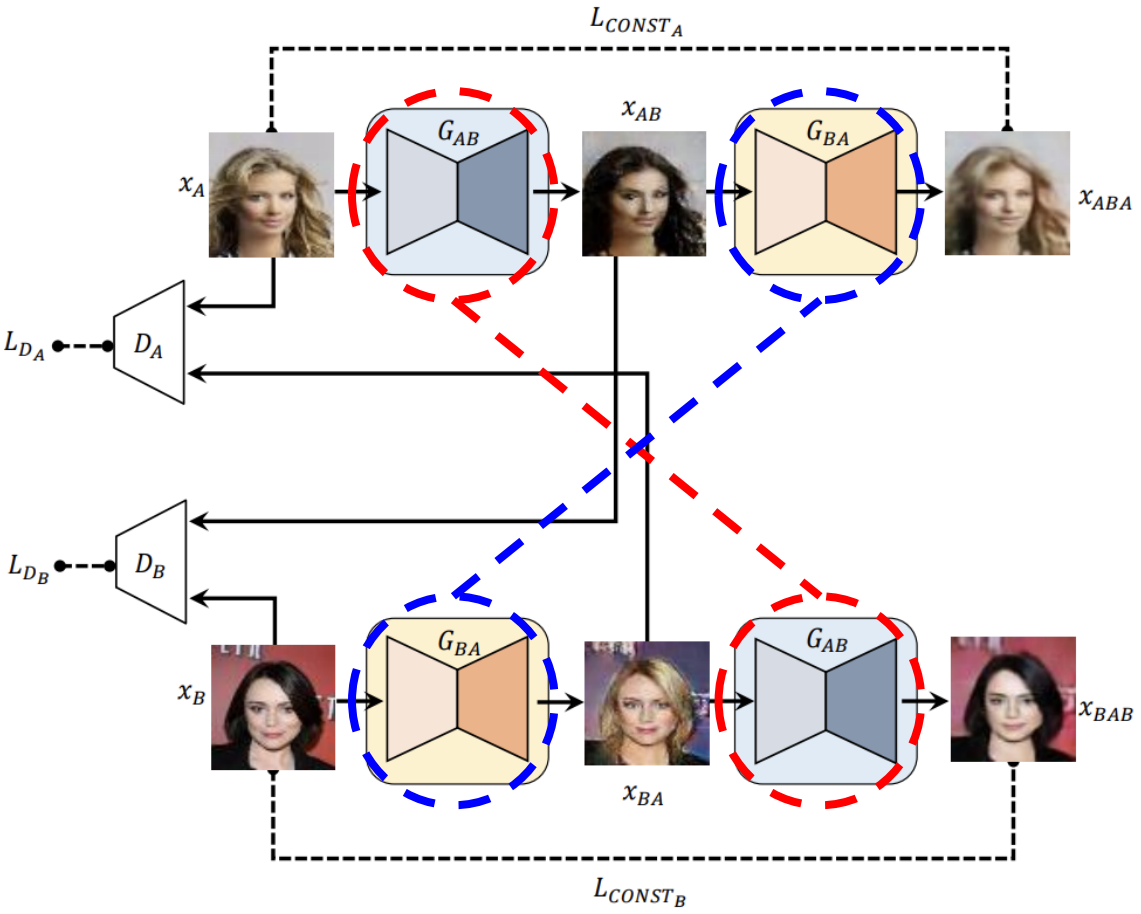
- Domain B: black hair



Kim, Taeksoo, et al. "Learning to discover cross-domain relations with generative adversarial networks." *arXiv preprint arXiv:1703.05192* (2017).

DiscoGAN

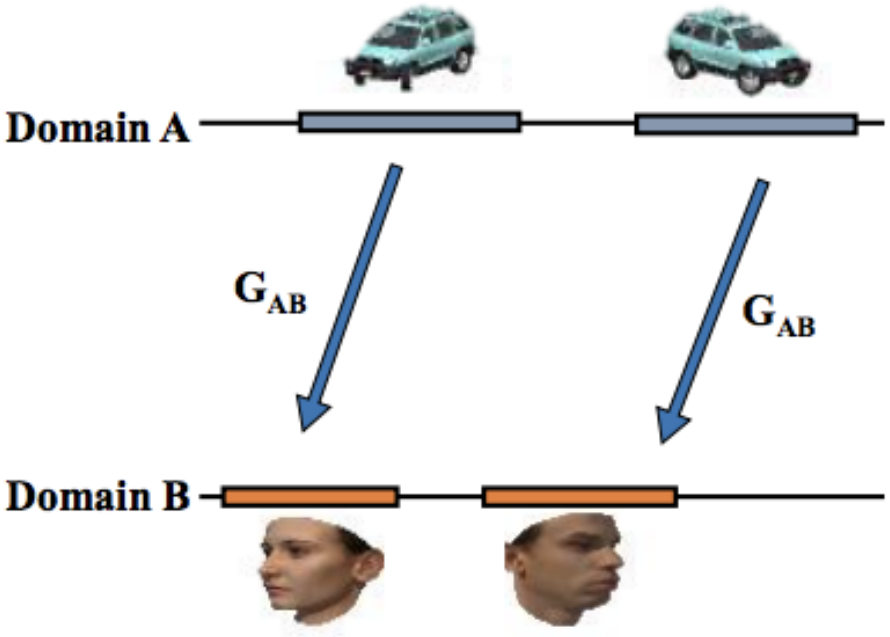
- Weight sharing/reuse is important
  - Same reason CNNs work
- Same network weights/activations must perform multiple tasks
  - Indirectly increases number of training samples and generalizability
- Reduces number of weights to train
  - Optimization is easier



Kim, Taeksoo, et al. "Learning to discover cross-domain relations with generative adversarial networks." *arXiv preprint arXiv:1703.05192* (2017).

DiscoGAN

- Weight sharing/reuse is important
  - Same reason CNNs work
- Same network weights/activations must perform multiple tasks
  - Indirectly increases number of training samples and generalizability
- Reduces number of weights to train
  - Optimization is easier



Kim, Taeksoo, et al. "Learning to discover cross-domain relations with generative adversarial networks." *arXiv preprint arXiv:1703.05192* (2017).

DiscoGAN

DiscoGAN

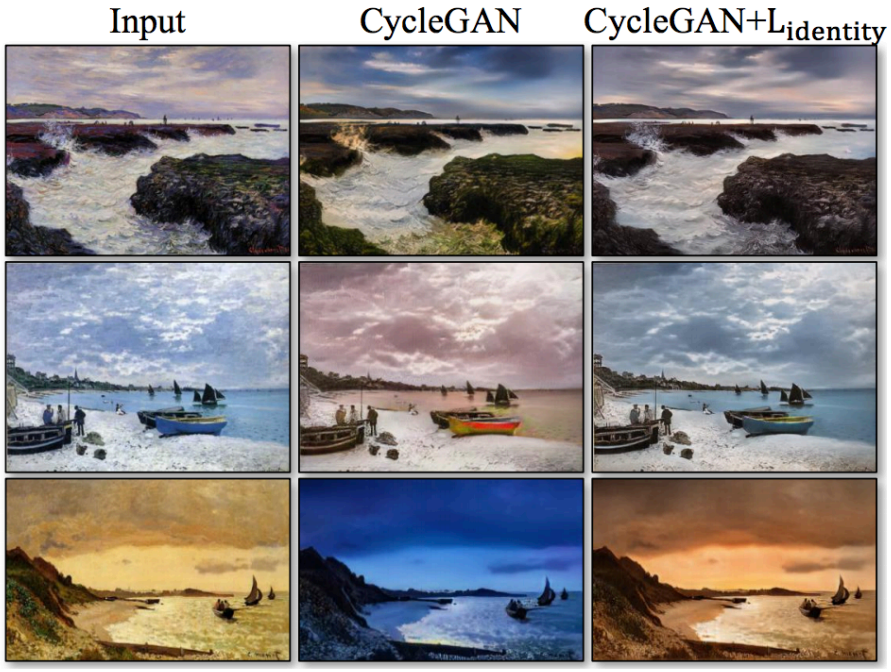
Issues

- Is this the correct shoe for this purse?
- Is there a better match?
- Original paper doesn't talk about it

- Sometimes the mapping isn't correct

CycleGAN

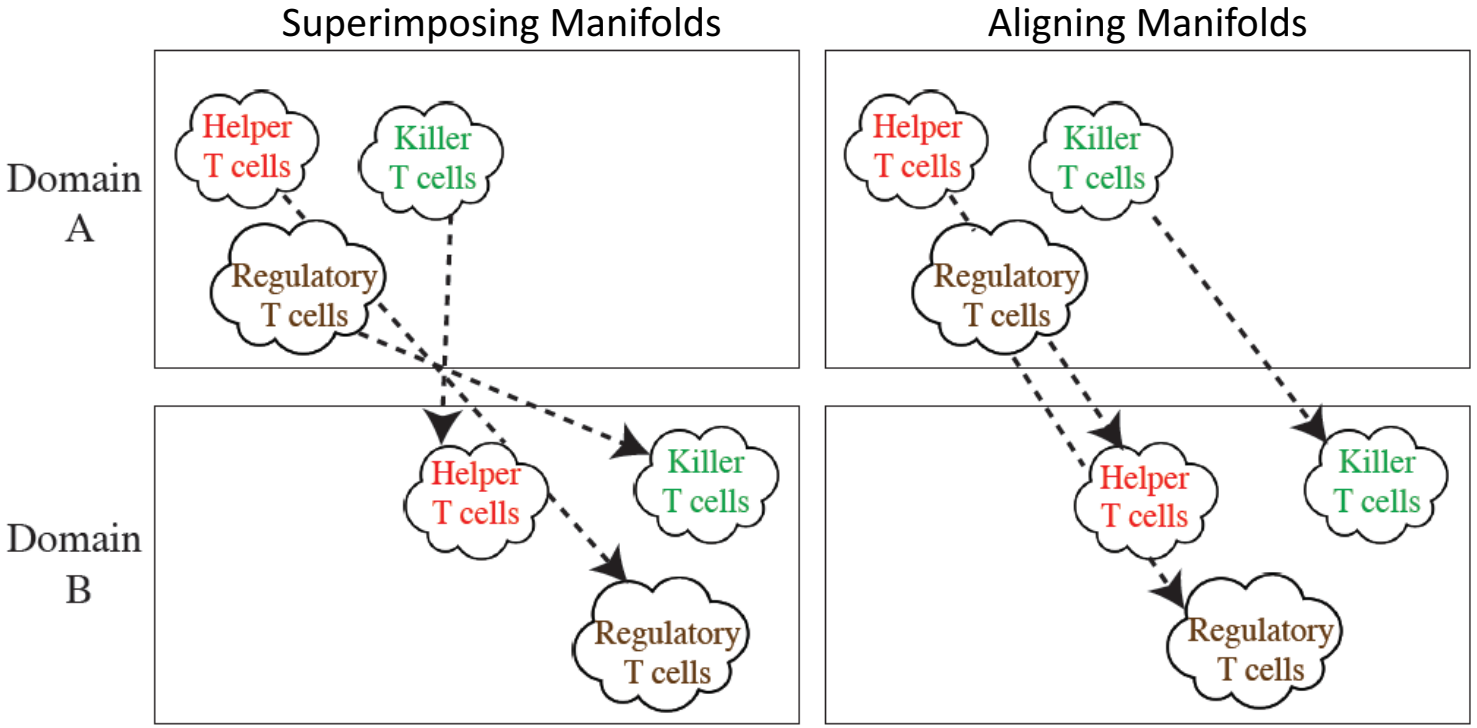
- Came out shortly after DiscoGAN, popularized the idea
- Added another component to the loss:
  - Normally G_{AB} takes x_A to domain B
  - This means $G_{AB}(x_A)$ will be different from x_A
  - Add additional constraint that $G_{AB}(x_B) = x_B$
    - Intuitively, G should take something not in A to B, but if it's already in B, it shouldn't change it



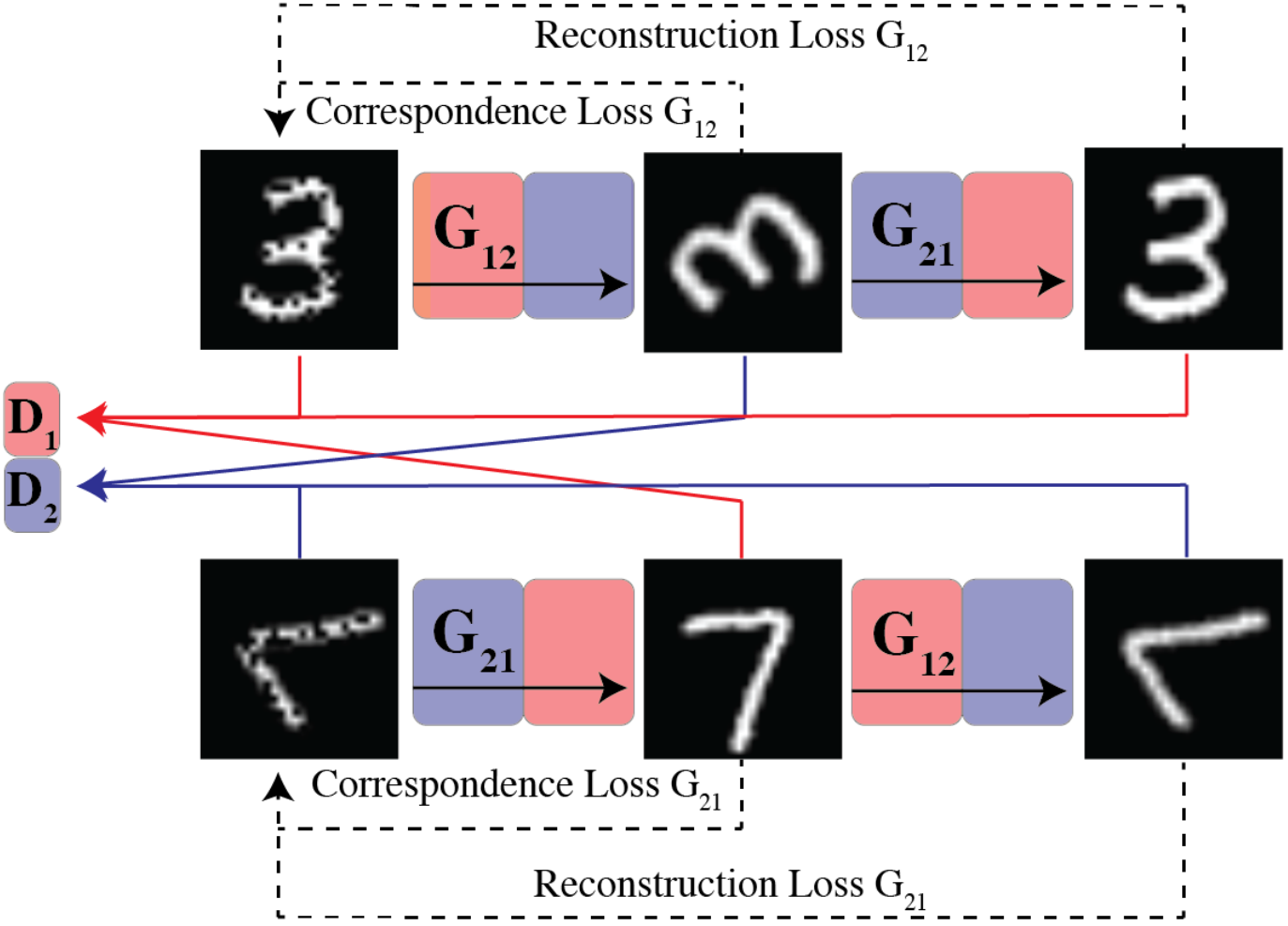
Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

Manifold Alignment GAN (MAGAN)

- Amodio & Krishnaswamy (2018)
- Consider manifolds A and B to be two CyTOF panels
  - Different markers, samples from same tissue
- Panel A measures 40 markers
- Panel B measures 40 markers
- 10 shared markers and 30 unique markers, each
- Using 10 shared markers, find corresponding cells across panels

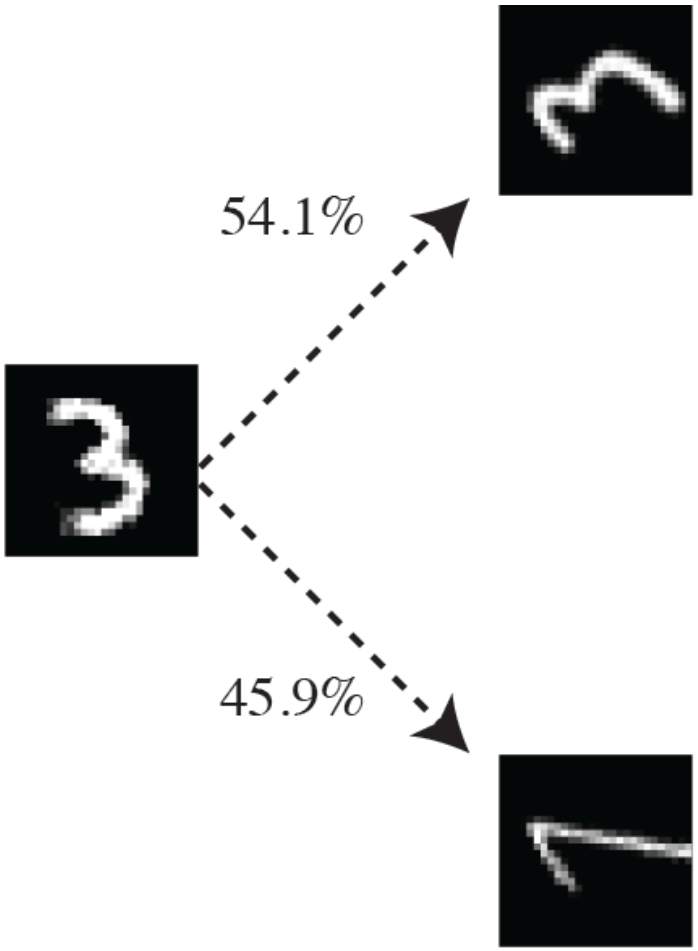


MAGAN Architecture

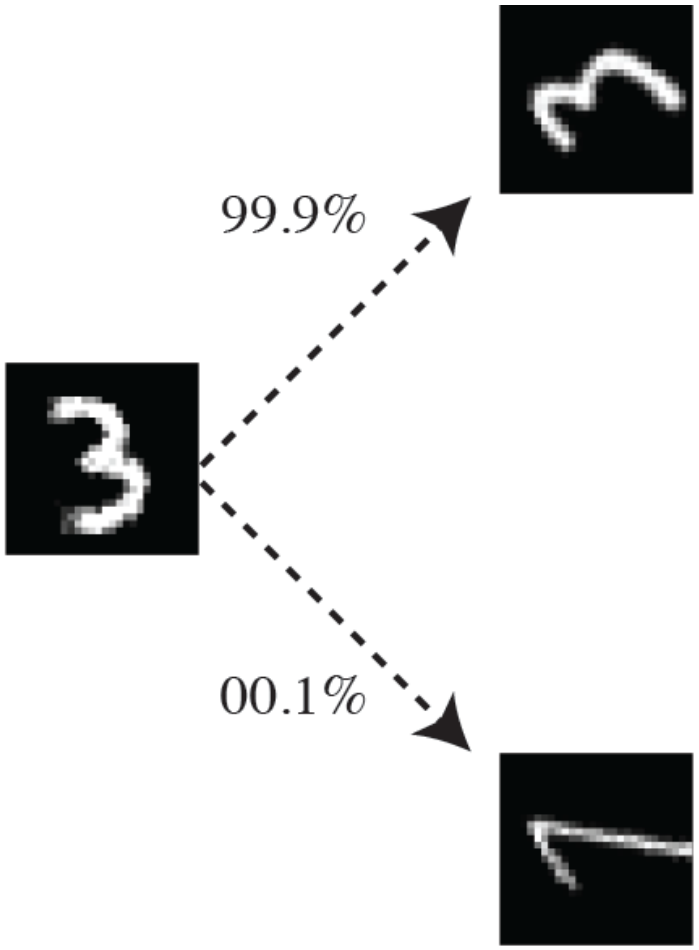


Matching rotated digits

- 100 simulations of each model



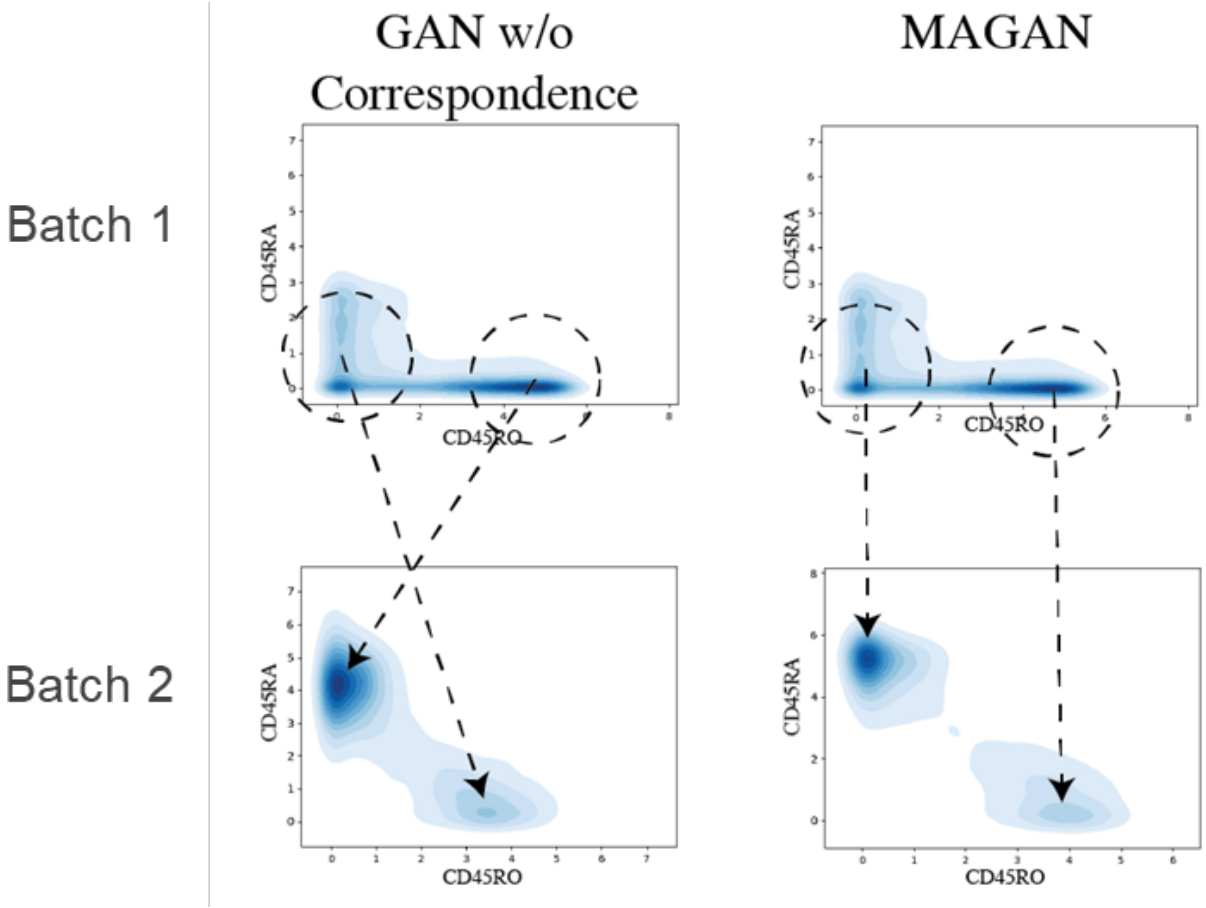
GAN w/o Correspondence



MAGAN

Aligning CyTOF Batches w/ MAGAN

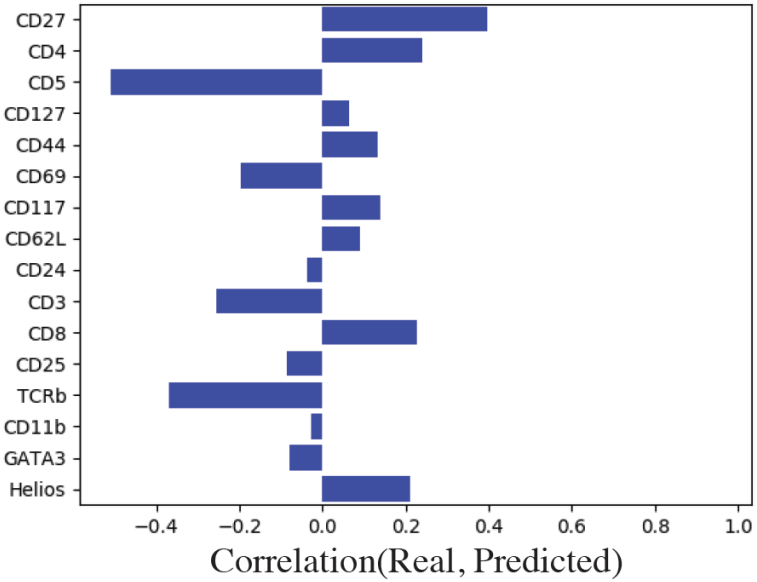
- Two underlying cell populations measured in each batch (naïve T cells and central memory T cells)
  - Dropout in CD45RA in batch 1
- MAGAN correctly matches the cell types in the two batches



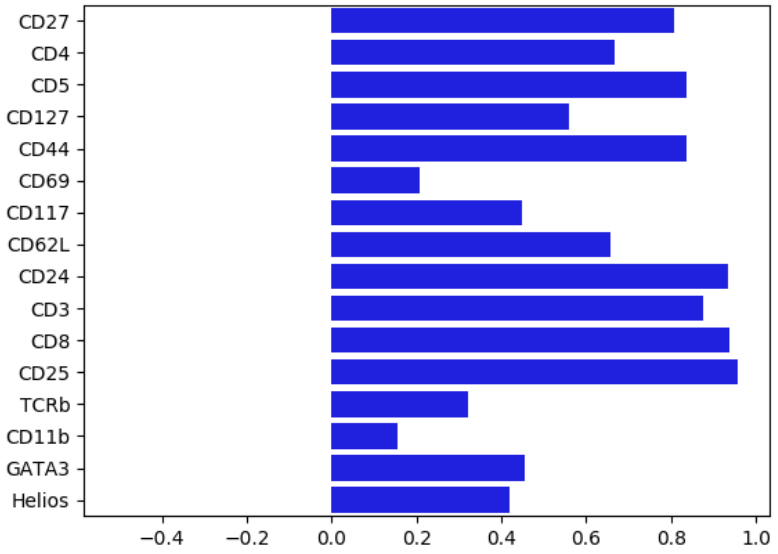
Aligning CyTOF Panels w/ MAGAN

- Validation on publicly available CyTOF data from developing mouse brain*
- Hold out a shared marker in one panel
- Apply MAGAN to reconstruct held-out marker
  - Compare correlation between real value and prediction

GAN w/o Correspondence



MAGAN



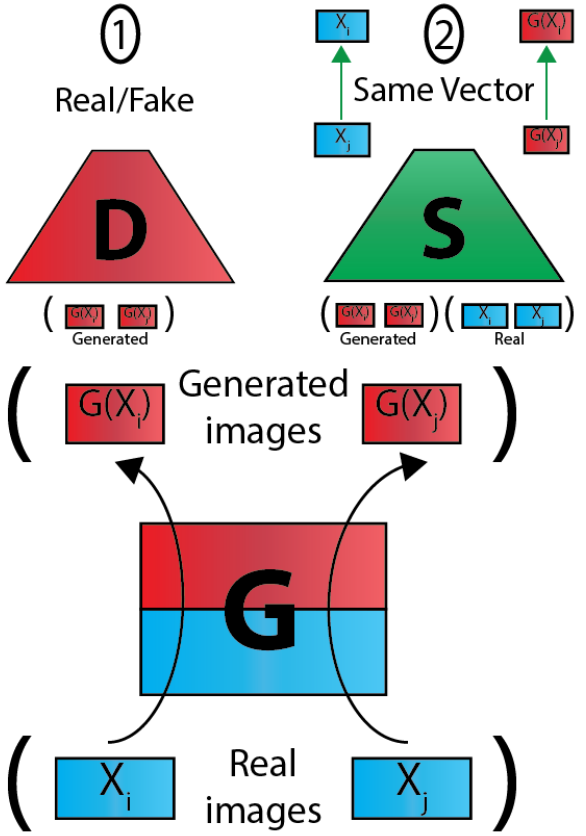
* Setty, Manu et al. "Wishbone identifies bifurcating developmental trajectories from single-cell data". *Nature Biotechnology.*

Limitations of Cycle-consistency

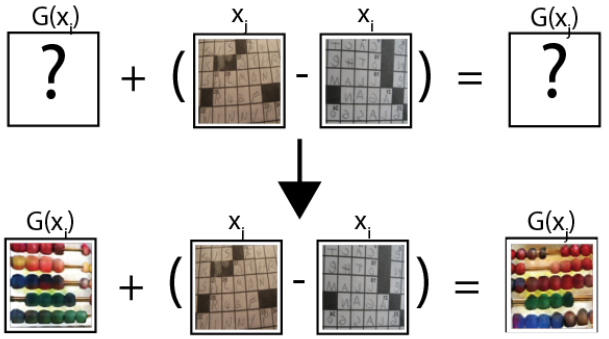
- Can only learn simple **style-transfer** transformations
  - e.g. paint a horse to look like a zebra
- Can only learn ***one*** transformation
  - e.g. all horses become zebras, and nothing else is transformed
  - Thus domains must be homogenous
  - Papers do a lot of cropping and centering…
- Transform based on the **pixel space**
  - Is MSE between two 256x256x3-dimensional images meaningful?
- Assumes **any invertible** transformation is meaningful
  - What does invertibility have to do with semantic meaning?
- Not **interpretable**
  - Why did individual *i* in domain 1 get paired with this particular individual in domain 2? CycleGAN offers no insight into its decision

TraVeLGAN

- Instead of enforcing cycle-consistency, enforce that the vector between two real points is the same as the vector between their corresponding two generated points
  - Key: do this in a latent space, not pixel space!
  - Where do we get this latent space?
- Learn another network (**S**) to go along with G and D to find a latent space where these transformations are preserved
  - Train S alongside the other networks, gradually developing an appropriate semantics
- Since we are no longer enforcing pixel-wise MSE with cycle-consistency, we have a lot more freedom to change the image for domain transfer now

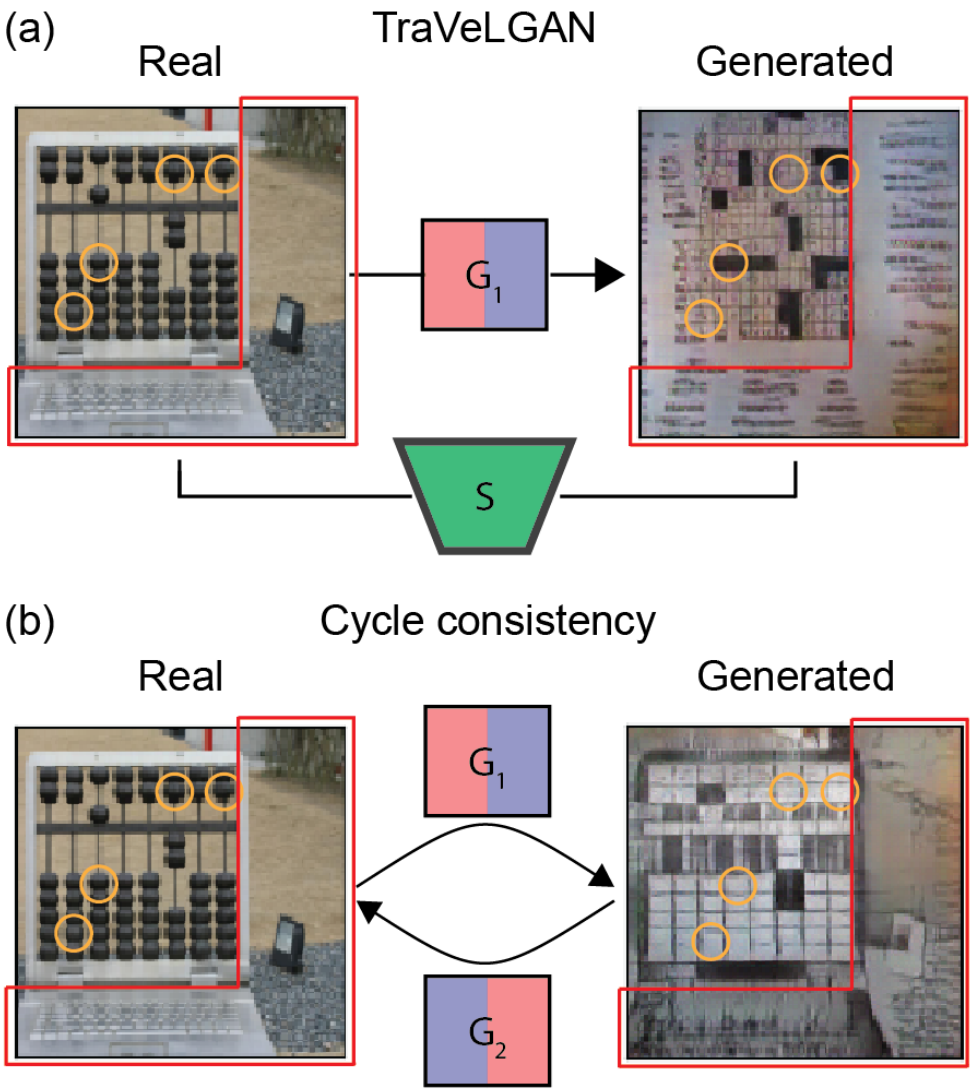


Transformation **Ve**ctor **L**earning (**TraVeL**ing)

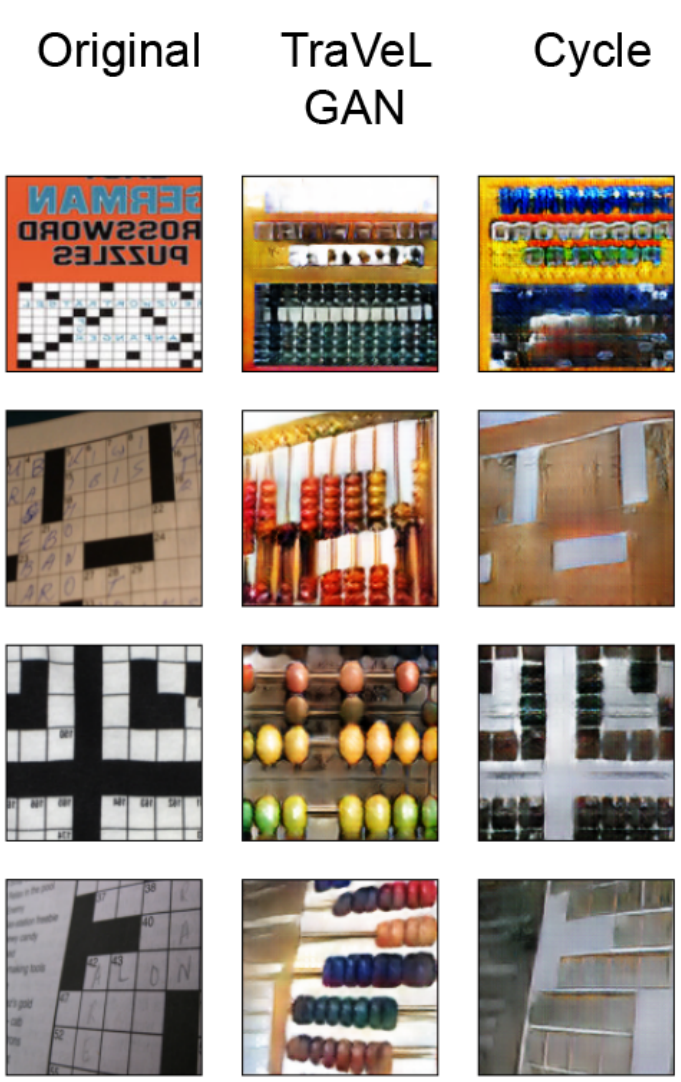


TraVeLGAN: examples

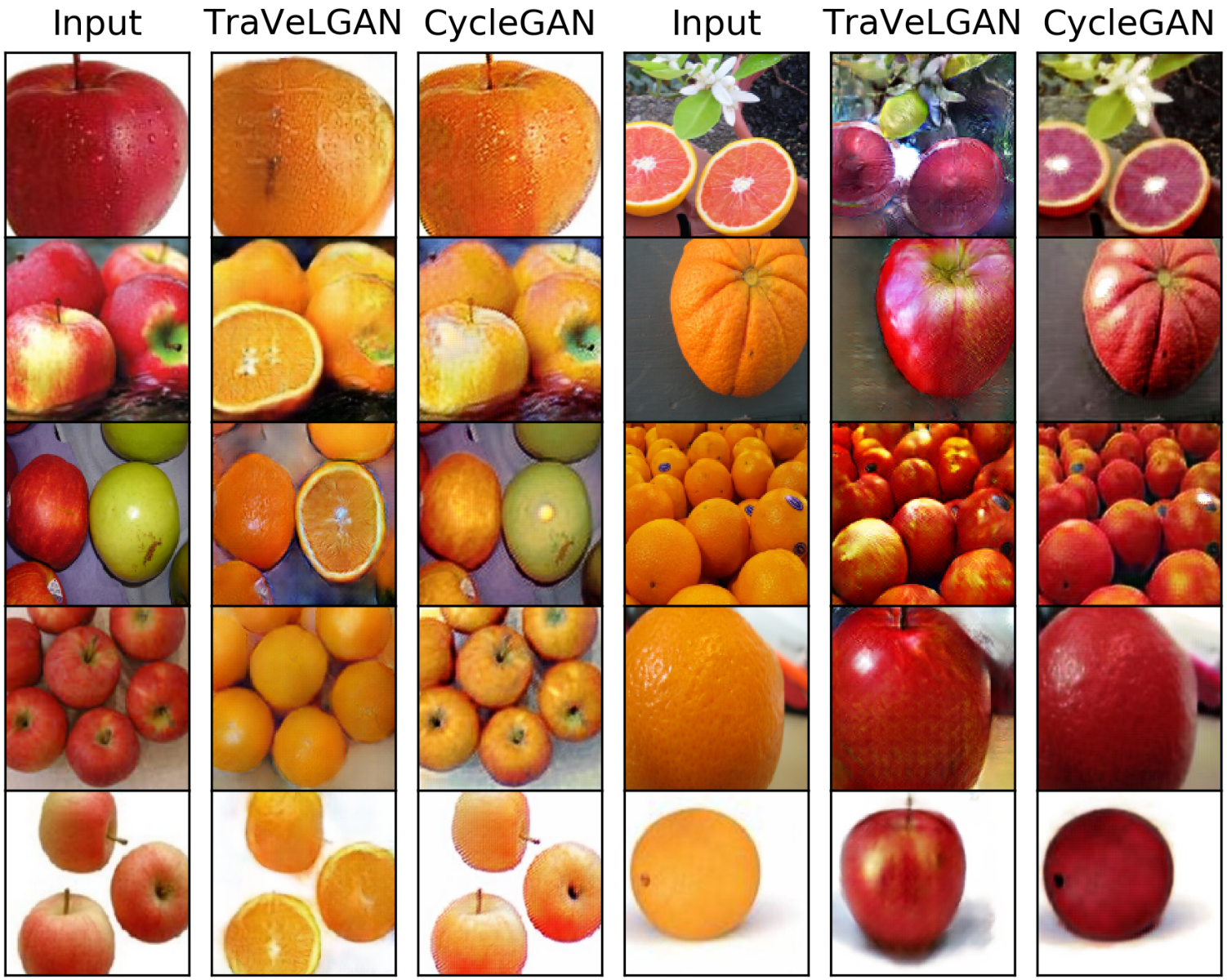
- Transferring abaci to crosswords
- The desired function is not invertible
- Any crossword configuration is a valid abacus configuration, but the reverse is not true
- Changes to the background of an image usually have to be lossy, which we can't have if we require pixel-wise cycle-consistency



TraVeLGAN: examples



TraVeLGAN: examples



Further reading

- Goodfellow et al., Section 20.10.4
- Lectures on GANs CS 11-785 at CMU
- DiscoGAN: https://arxiv.org/abs/1703.05192
- CycleGAN: https://arxiv.org/abs/1703.10593
- MAGAN: http://proceedings.mlr.press/v80/amodio18a.html
- TraVeLGAN: https://arxiv.org/abs/1902.09631