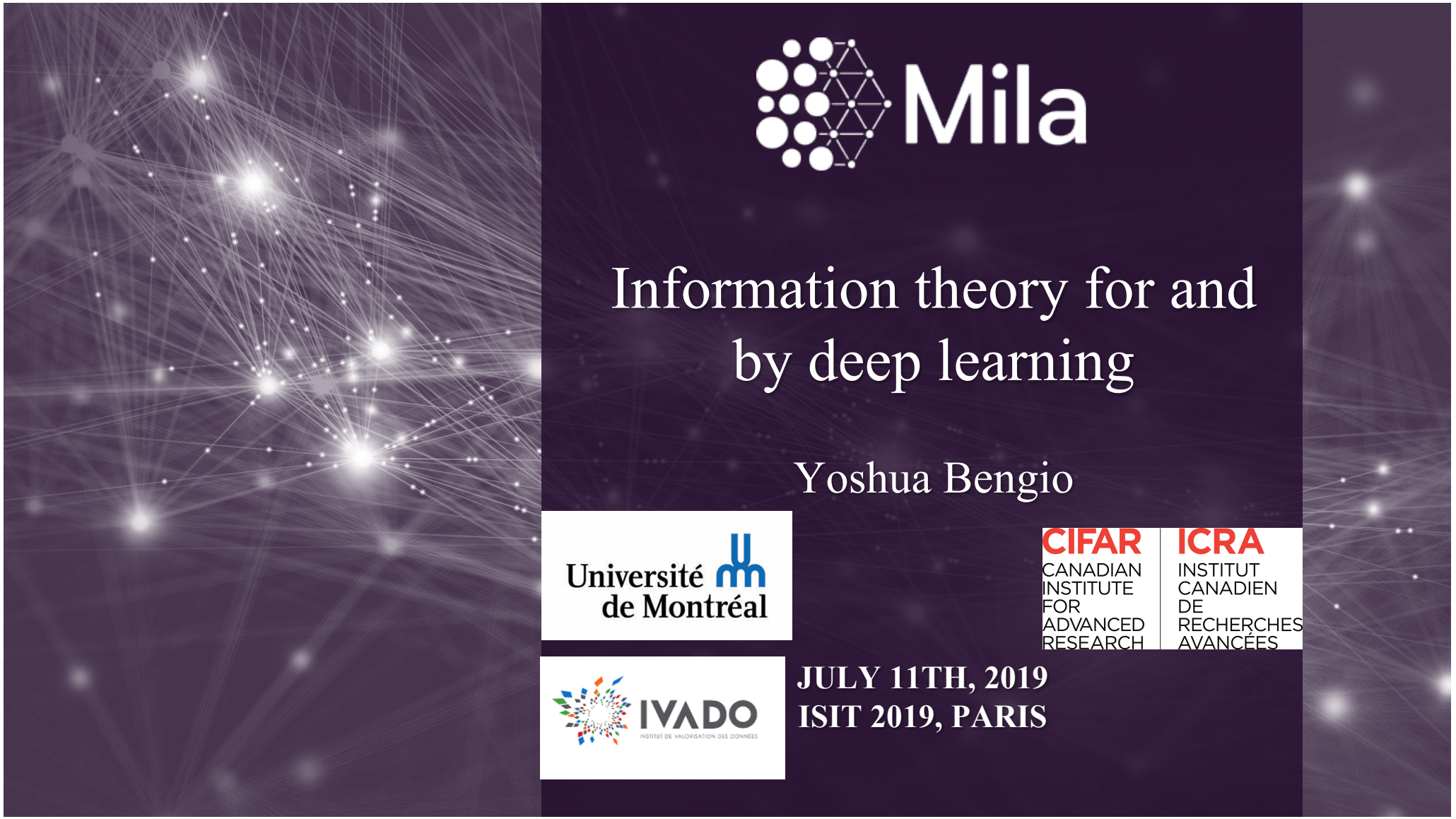
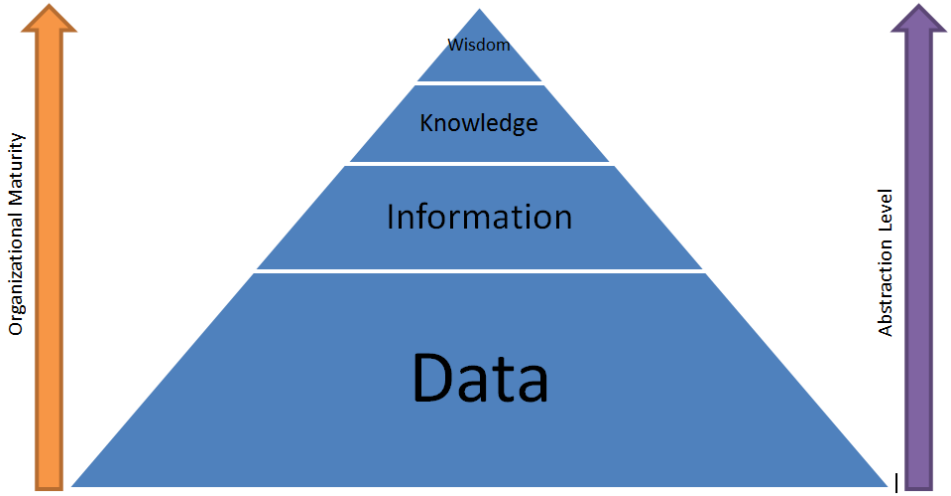

Mila
Information theory for and by deep learning
Yoshua Bengio
Université de Montréal
CIFAR
CANADIAN INSTITUTE FOR ADVANCED RESEARCH
ICRA
INSTITUT CANADIEN DE RECHERCHES AVANCÉES
IVADO
INSTITUT DE VALORISATION DES DONNÉES
JULY 11TH, 2019
ISIT 2019, PARIS

Learning Multiple Levels of Abstraction

(Bengio & LeCun 2007)

- The big payoff of deep learning is to allow learning higher levels of abstraction
- Higher-level abstractions **disentangle the factors of variation**, which allows much easier generalization and transfer



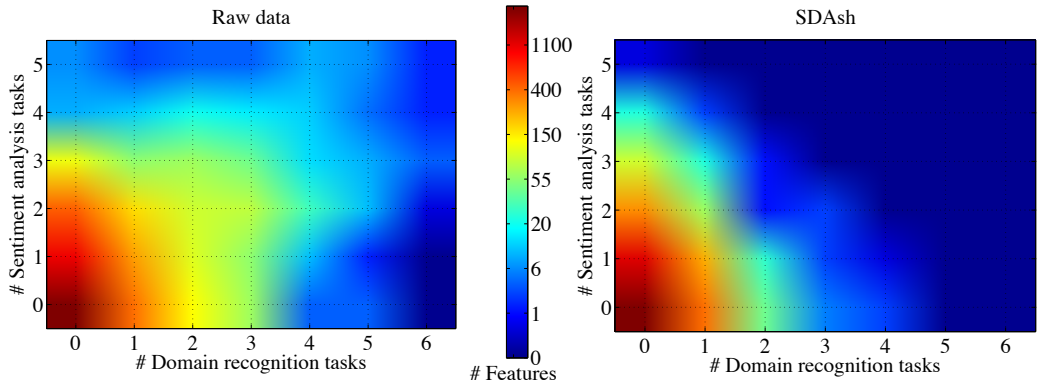
Invariance and Disentangling

- Invariant features
- Which invariances?
- Alternative: learning to disentangle factors
- Good disentangling →
  avoid the curse of dimensionality:
  Dependencies are "simple" when the data is projected in the right abstract space of high-level concepts

Disentangling from unsupervised objective

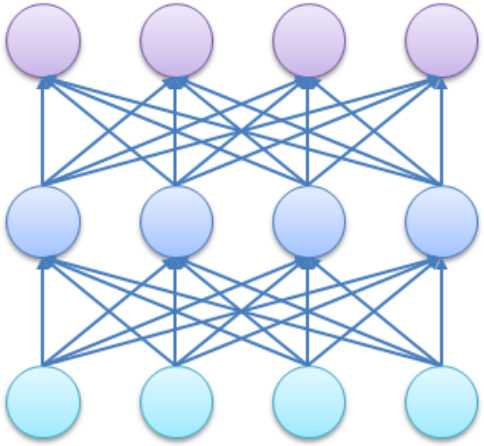
(Glorot, Bordes & Bengio ICML 2011)

- Early deep learning research already is looking for possible disentangling arising from unsupervised learning of representations
- Experiments on stacked denoising auto-encoders: features specialize to known underlying factors



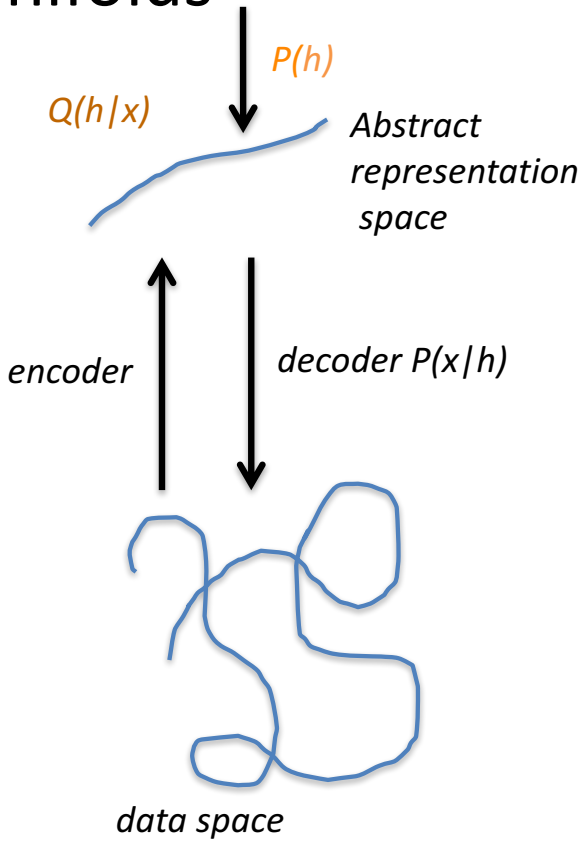
How to Discover Good Disentangled Representations

- How to discover abstractions?
- What is a good representation? ***(Bengio et al 2013)***
- Need clues (= priors) to help **disentangle** the underlying factors, such as
  - Spatial & temporal scales
  - Marginal independence
  - Simple & sparse dependencies between factors
    - *Consciousness prior*
  - Causal / mechanism independence
    - *Controllable factors*
  - Multiple spatial and temporal scales
    - *Coarse high-level factors explain lower-level details*



Latent Variables and Abstract Representations to Disentangle Manifolds

- Encoder/decoder view: maps between low & high-levels
- Encoder does inference: interpret the data at the abstract level
- Decoder can generate new configurations
- Encoder flattens and disentangles the data manifold

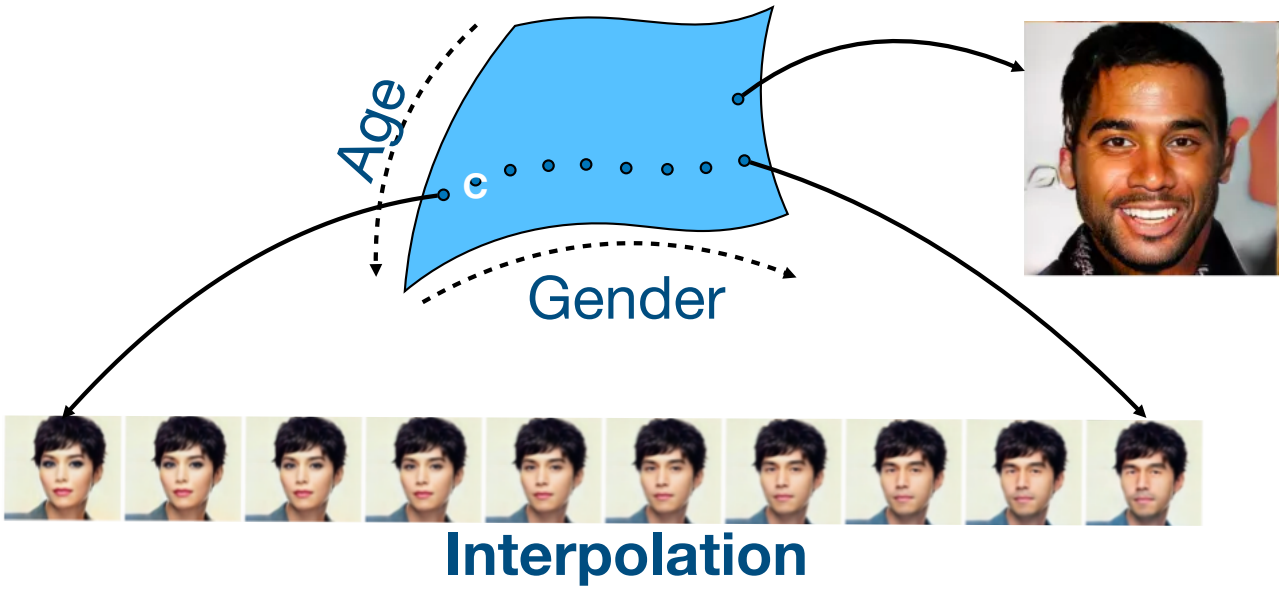


Why Latent Space Generative Models?

Discovery

- Learn relevant factors

"What I cannot create, I do not understand"
-Richard Feynman

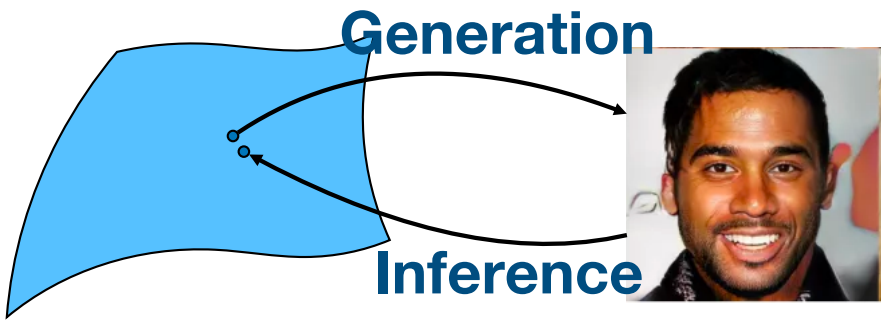


Why Latent Space Generative Models?

Discovery

- Learn relevant factors
- Inference

“What I cannot create, I do not understand”
-Richard Feynman

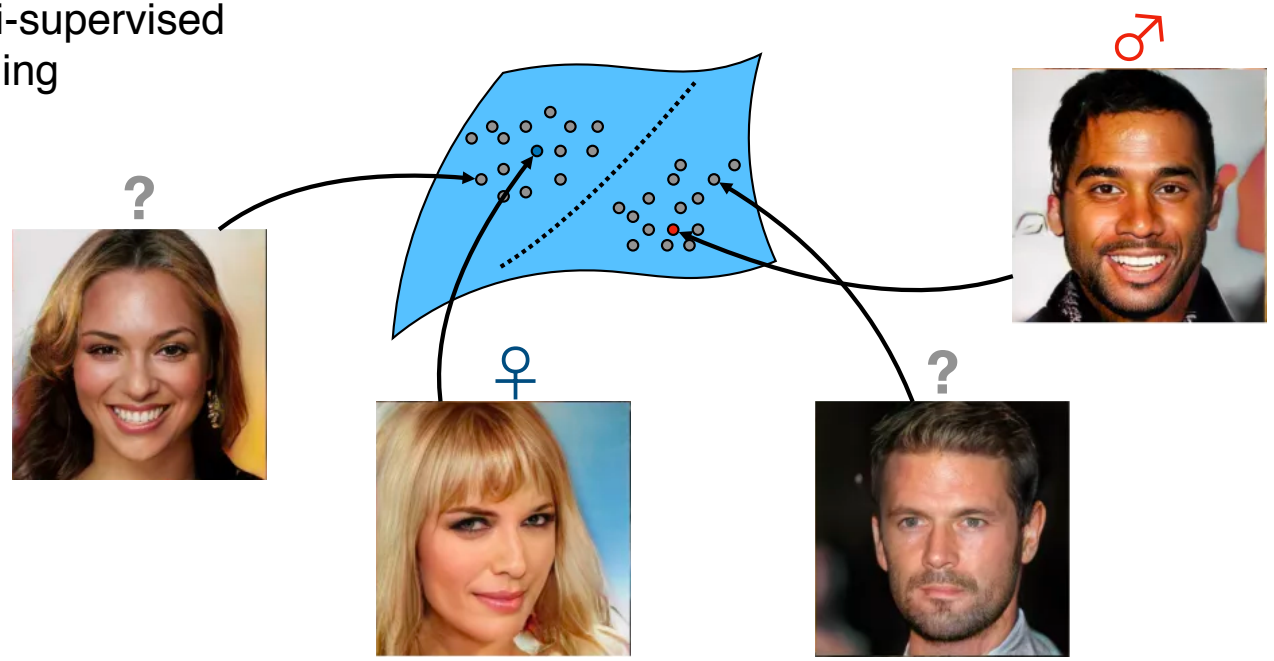


Why Latent Space Generative Models?

Discovery

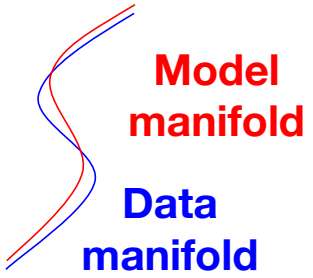
- Learn relevant factors
- Inference
- Semi-supervised learning

"What I cannot create, I do not understand"
-Richard Feynman



What's wrong with standard maximum likelihood?

- Pay a huge price for not putting probability mass at even a single training example, even if the data manifold and model manifold are very close.

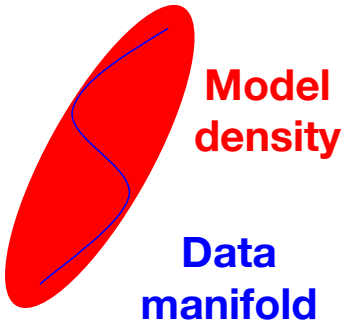


What's wrong with standard maximum likelihood?

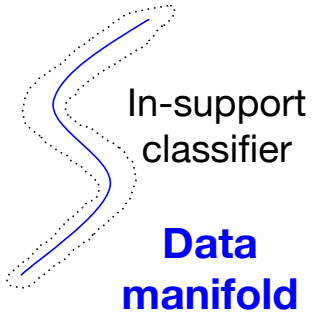
1. Pay a huge price for not putting probability mass at even a single training example, even if the data manifold and model manifold are very close.

   - So MLE makes the model distribution very fat and conservative

2. Another problem is that MLE measures error bits in pixel space whereas humans really care about errors in abstract space, so we would like loss measured in learned latent space



Classifiers for modeling distributions



- We were inspired by the work of Gutmann & Hyvarinen using probabilistic classifiers to estimate energy functions

  Gutmann & Hyvarinen 2012, Noise-Contrastive Estimation

- In high dimension, more relevant then density is whether you are in-support vs out-of-support

- A classifier of in-support vs out-of-support pays a *constant* price (rather than huge) for not putting support at a training example

Generative adversarial networks (GANs): a two player game with neural networks

Givens:

Samples from a **target distribution** $\mathbb{P}$

(Simple) prior $\mathbb{Q}_z$

$\mathbb{P}$

$\mathbb{Q}_z$

[Goodfellow et. al., 2014]

Generative adversarial networks (GANs): a two player game with neural networks

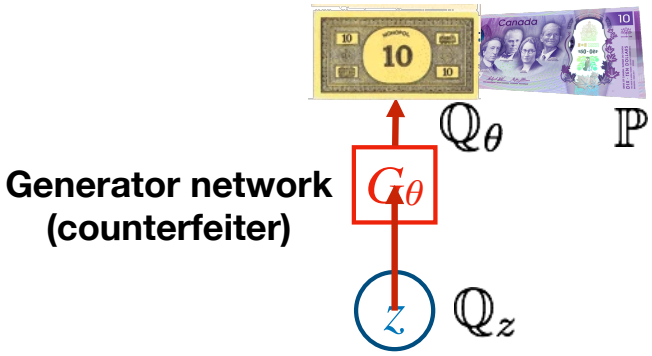
Givens:

Samples from a **target distribution** $\mathbb{P}$

(Simple) prior $\mathbb{Q}_z$

Player 1: Generator

A neural network with parameters, θ, whose samples **fool the discriminator**



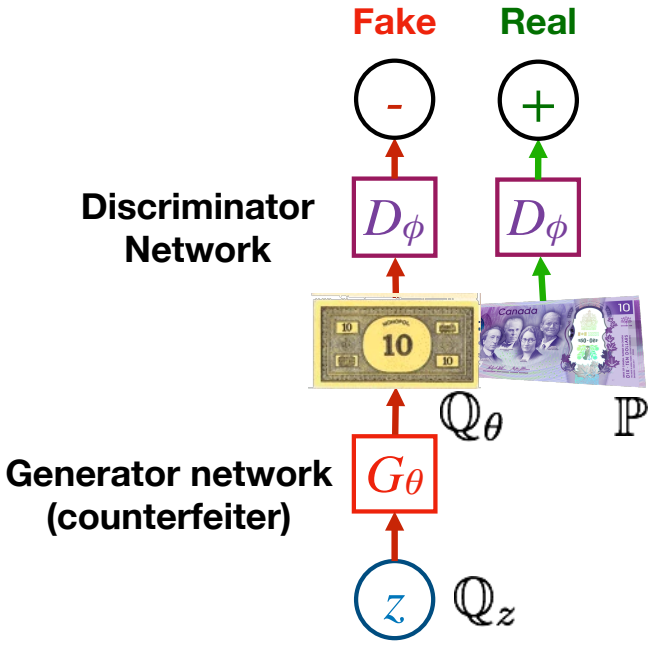
[Goodfellow et. al., 2014]

Generative adversarial networks (GANs): a two player game with neural networks

Givens:
Samples from a **target distribution** $\mathbb{P}$
(Simple) prior $\mathbb{Q}_z$

Player 1: Generator
A neural network with parameters, θ, whose samples **fool the discriminator**

Player 2: Discriminator
Distinguish (classify) real and fake correctly



[Goodfellow et. al., 2014]

Generative adversarial networks (GANs): a two player game with neural networks

Givens:
Samples from a **target distribution** $\mathbb{P}$
(Simple) prior $\mathbb{Q}_z$

Player 1: Generator
A neural network with parameters, θ, whose samples **fool the discriminator**

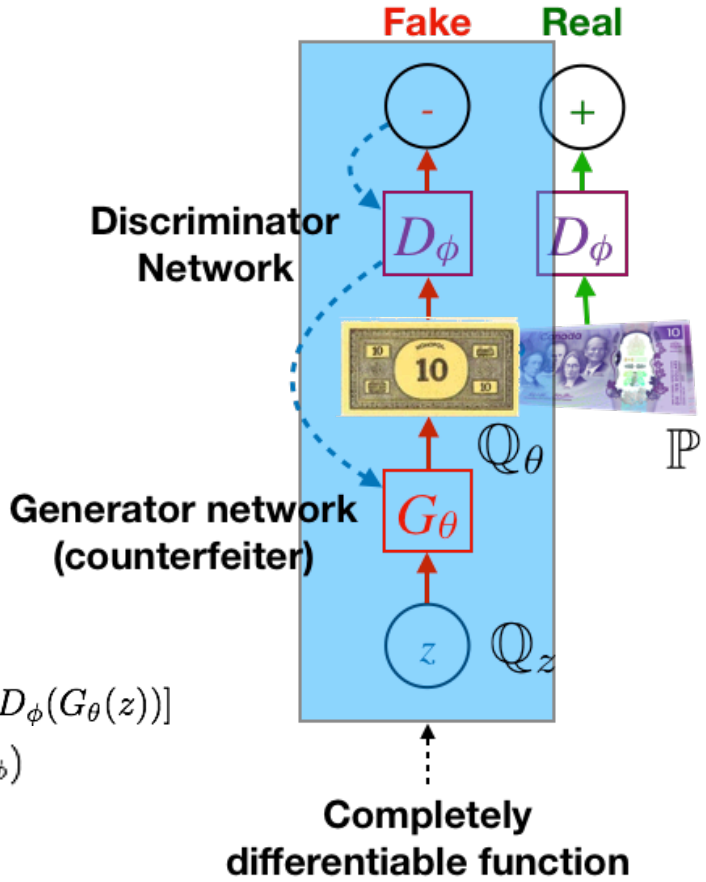
Player 2: Discriminator
Distinguish (classify) real and fake correctly

Minimax on *value function*

$$\mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta, D_\phi) = \mathbb{E}_{\mathbb{P}}\left[\log D_\phi(x)\right] + \mathbb{E}_{\mathbb{Q}_z}\left[\log(1 - D_\phi(G_\theta(z))\right]$$

$$(\hat{\theta}, \hat{\phi}) = \arg\min_\theta \arg\max_\phi \mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta, D_\phi)$$

Fine print: Continuous data only

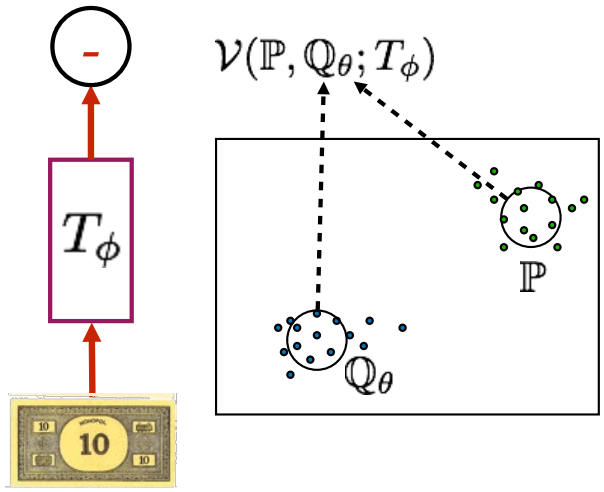


[Goodfellow et. al., 2014]

A closer look at the discriminator

- The discriminator defines a lower-bound

$$2 * \mathcal{D}_{JSD}(\mathbb{P}||\mathbb{Q}_\theta) - \log 4 \geq \mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta; T_\phi)$$



[*f-GAN*. Nowozin et. al., 2017]

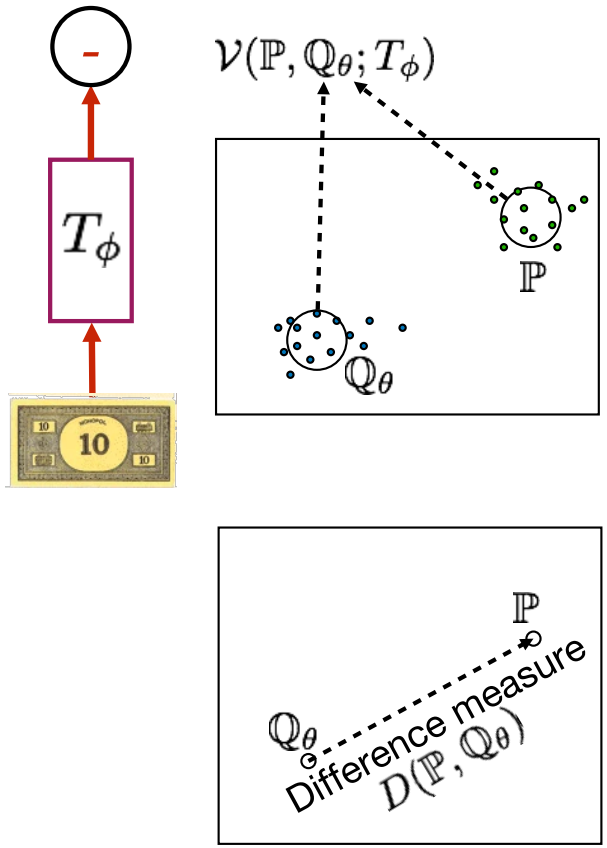
A closer look at the discriminator

- The discriminator defines a lower-bound

$$2 * \mathcal{D}_{JSD}(\mathbb{P}||\mathbb{Q}_\theta) - \log 4 \geq \mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta; T_\phi)$$

- f-divergence

$$\mathcal{D}_f(\mathbb{P}||\mathbb{Q}_\theta) = \mathbb{E}_{\mathbb{Q}_\theta}\left[f\left(\frac{p(x)}{q_\theta(x)}\right)\right]$$



Primal

[*f-GAN*. Nowozin et. al., 2017]

A closer look at the discriminator

- The discriminator defines a lower-bound

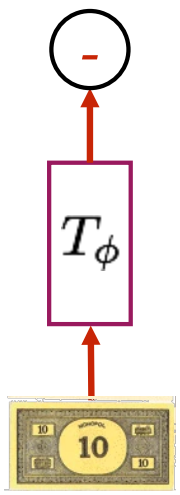
$$2 * \mathcal{D}_{JSD}(\mathbb{P}||\mathbb{Q}_\theta) - \log 4 \geq \mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta; T_\phi)$$

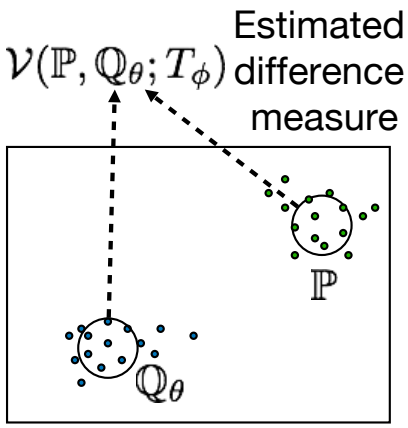
- f-divergence

$$\mathcal{D}_f(\mathbb{P}||\mathbb{Q}_\theta) = \mathbb{E}_{\mathbb{Q}_\theta}\left[f\left(\frac{p(x)}{q_\theta(x)}\right)\right]$$

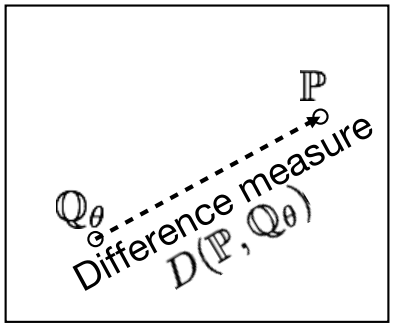
- Convex dual using neural networks

$$\begin{aligned}\mathcal{D}_f(\mathbb{P}||\mathbb{Q}_\theta) &\geq \mathbb{E}_{\mathbb{P}}[T_\phi(x)] - \mathbb{E}_{\mathbb{Q}_\theta}[f^\star(T_\phi(x))] \\ &= \mathcal{V}_f(\mathbb{P}, \mathbb{Q}_\theta; T_\phi)\end{aligned}$$





Dual using a classifier



Primal

[*f-GAN*. Nowozin et. al., 2017]

A closer look at the discriminator

- The discriminator defines a lower-bound

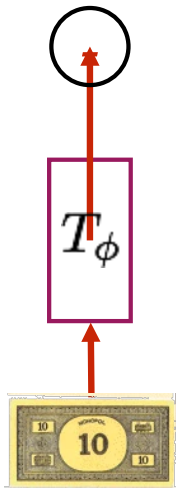
$$2 * \mathcal{D}_{JSD}(\mathbb{P}||\mathbb{Q}_\theta) - \log 4 \geq \mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta; T_\phi)$$

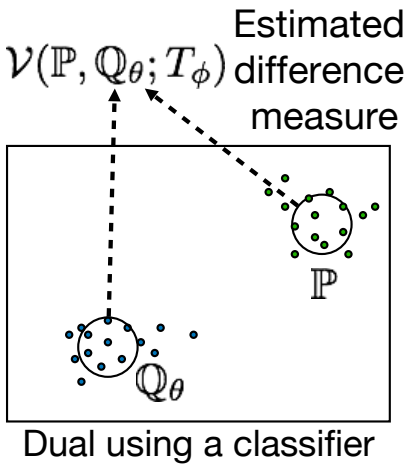
- f-divergence

$$\mathcal{D}_f(\mathbb{P}||\mathbb{Q}_\theta) = \mathbb{E}_{\mathbb{Q}_\theta}\left[f\left(\frac{p(x)}{q_\theta(x)}\right)\right]$$

- Convex dual using neural networks

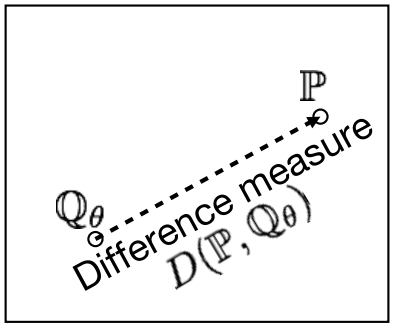
$$\begin{aligned}\mathcal{D}_f(\mathbb{P}||\mathbb{Q}_\theta) &\geq \mathbb{E}_{\mathbb{P}}[T_\phi(x)] - \mathbb{E}_{\mathbb{Q}_\theta}[f^\star(T_\phi(x))] \\ &= \mathcal{V}_f(\mathbb{P}, \mathbb{Q}_\theta; T_\phi)\end{aligned}$$





Dual using a classifier

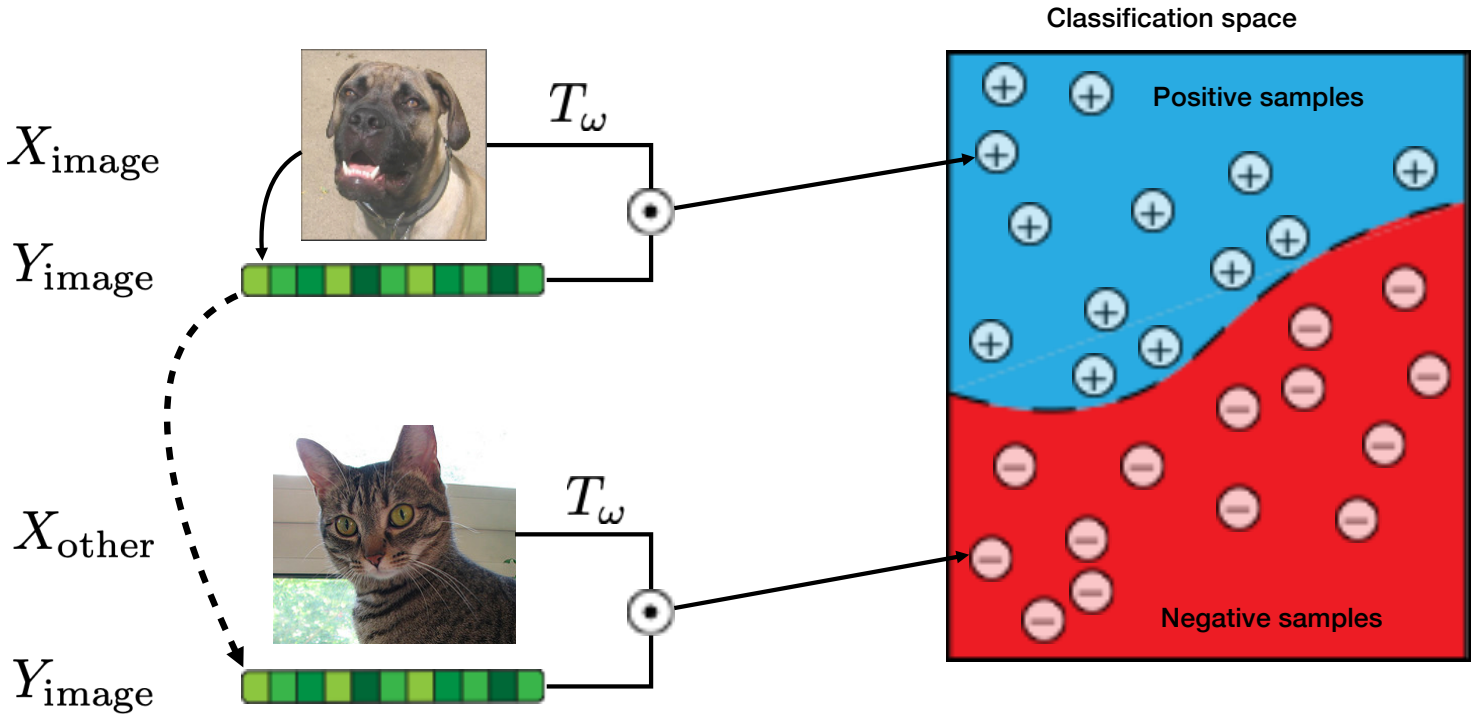
- Estimate using samples
- **Other Examples**
  KL, Jensen-Shannon, Squared Hellinger, Pearson χ^2
- **GANS are a convex dual optimization with a classifier**



Primal

[*f-GAN*. Nowozin et. al., 2017]

MI estimation is classification



Positive sample: sample from the joint distribution (e.g., $(X_{\text{image}}, Y_{\text{image}})$)

Negative sample: sample from the product of marginals (e.g., $(X_{\text{other}}, Y_{\text{image}})$)

Using a discriminator to optimize independence, mutual information or entropy

- The GAN discriminator is trained to estimate a similarity function between two distributions

- Two independent r-v A & B have the property that P(A,B)=P(A)P(B)

  - Given samples from P(A,B) you can obtain samples from P(A)P(B), e.g. by shuffling A values within a minibatch

Train a discriminator to separate between pairs (A,B) coming from P(A,B) and pairs coming from P(A) P(B)

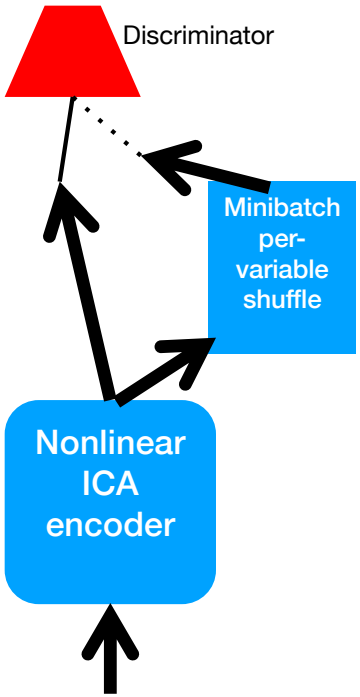
Brakel & Bengio ArXiv:1710.05050

Using a discriminator to optimize **independence**, mutual information or entropy

Brakel & Bengio ArXiv:1710.05050

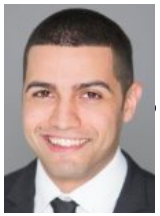
- Train a discriminator to separate between pairs (A,B) coming from P(A,B) and pairs coming from P(A) P(B)

- Generalize this to measuring **independence** of all the outputs of a representation function (encoder). Maximize independence by backprop independence score into encoder → NON-LINEAR ICA.



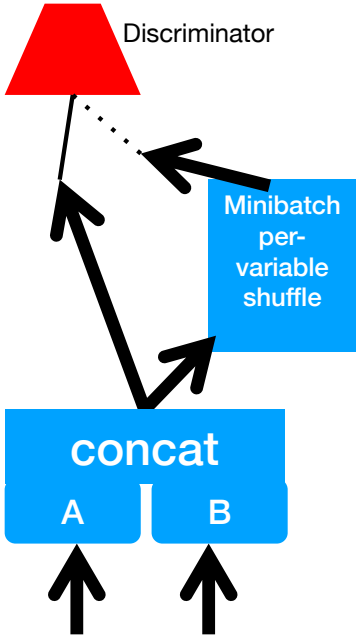
Using a discriminator to optimize independence, mutual information or entropy

MINE: Mutual Information Neural Estimator



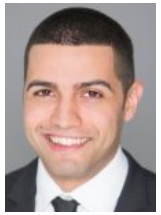
Belghazi et al ArXiv:1801.04062

Same architecture, but with a twist in the training objective which provides an asymptotically correct estimator of mutual independence



- Note that

$$MI(A,B) = H[A] - H[B|A]$$



Mutual information neural estimator (MINE)

Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, R Devon Hjelm

Mutual information: measure of dependence between two variables

$$I(X;Z) = \mathcal{D}_{KL}(\mathbb{P}_{X,Z}||\mathbb{P}_X \otimes \mathbb{P}_Z) = \mathbb{E}_{\mathbb{P}_{X,Z}}\left[\log\left(\frac{p(x,z)}{p(x)p(z)}\right)\right]$$

Fenchel convex dual (f-GAN): MINE-f

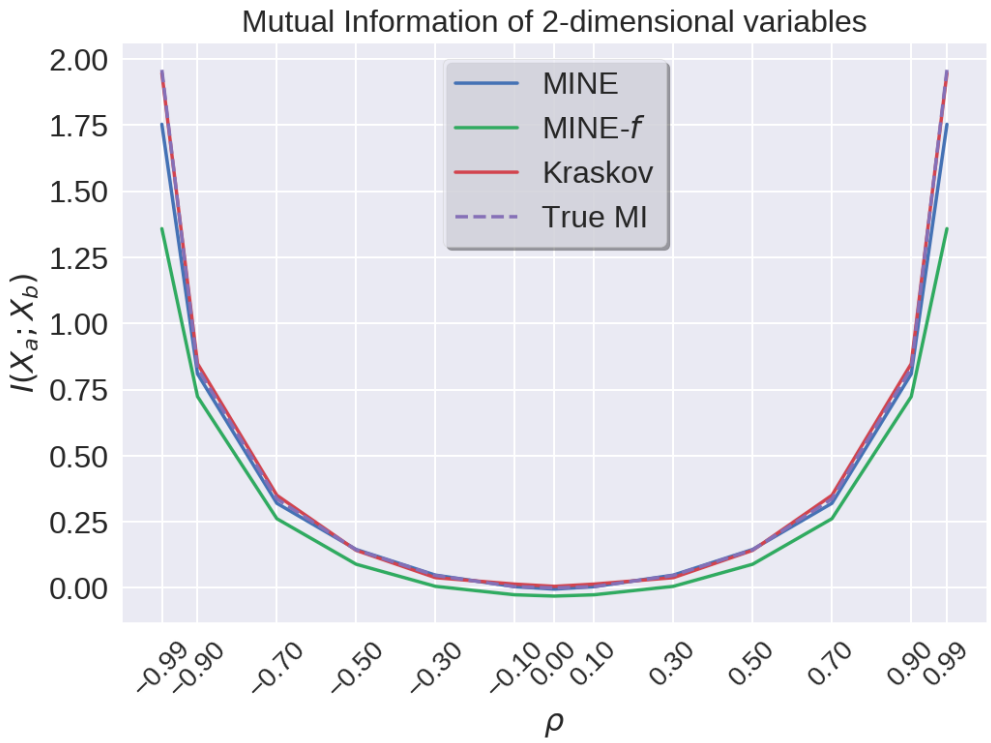
$$\mathcal{D}_{KL}(\mathbb{P}_{X,Z}||\mathbb{P}_X \otimes \mathbb{P}_Z) \geq \mathbb{E}_{\mathbb{P}_{X,Z}}[T_\phi(x)] - \mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Z}[e^{T_\phi(x)-1}]$$

Donsker-Varadhan (tighter): MINE

$$\mathcal{D}_{KL}(\mathbb{P}_{X,Z}||\mathbb{P}_X \otimes \mathbb{P}_Z) \geq \mathbb{E}_{\mathbb{P}_{X,Z}}[T_\phi(x)] - \log \mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Z}[e^{T_\phi(x)}]$$

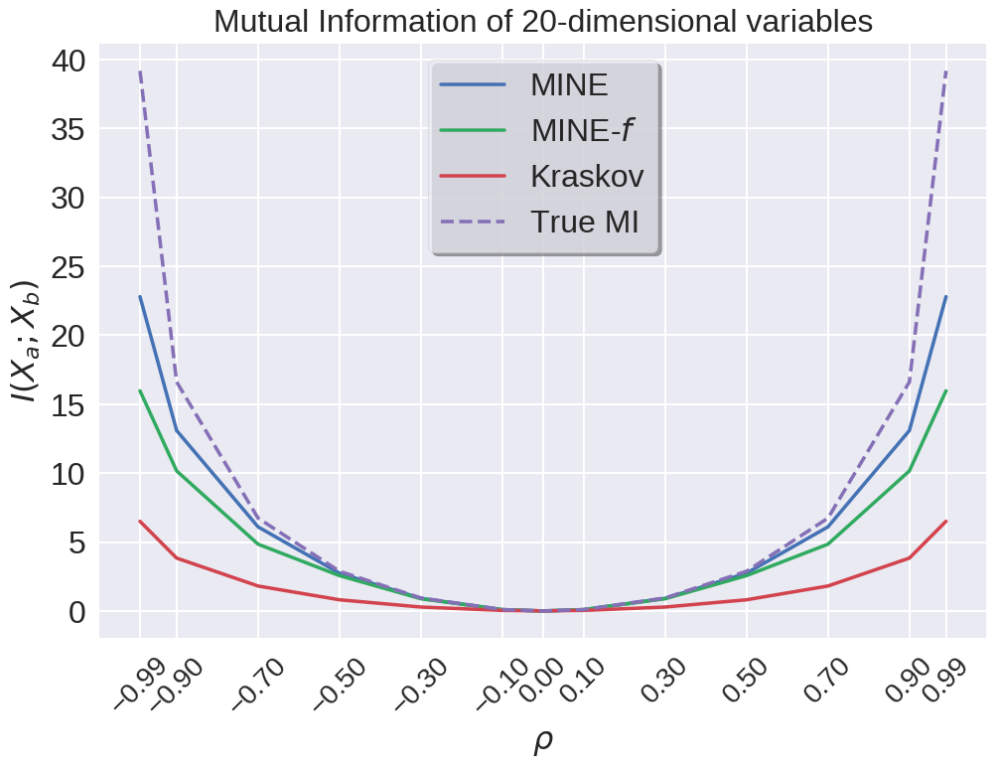
[Belghazi et. al., 2018]

Demonstration of estimation



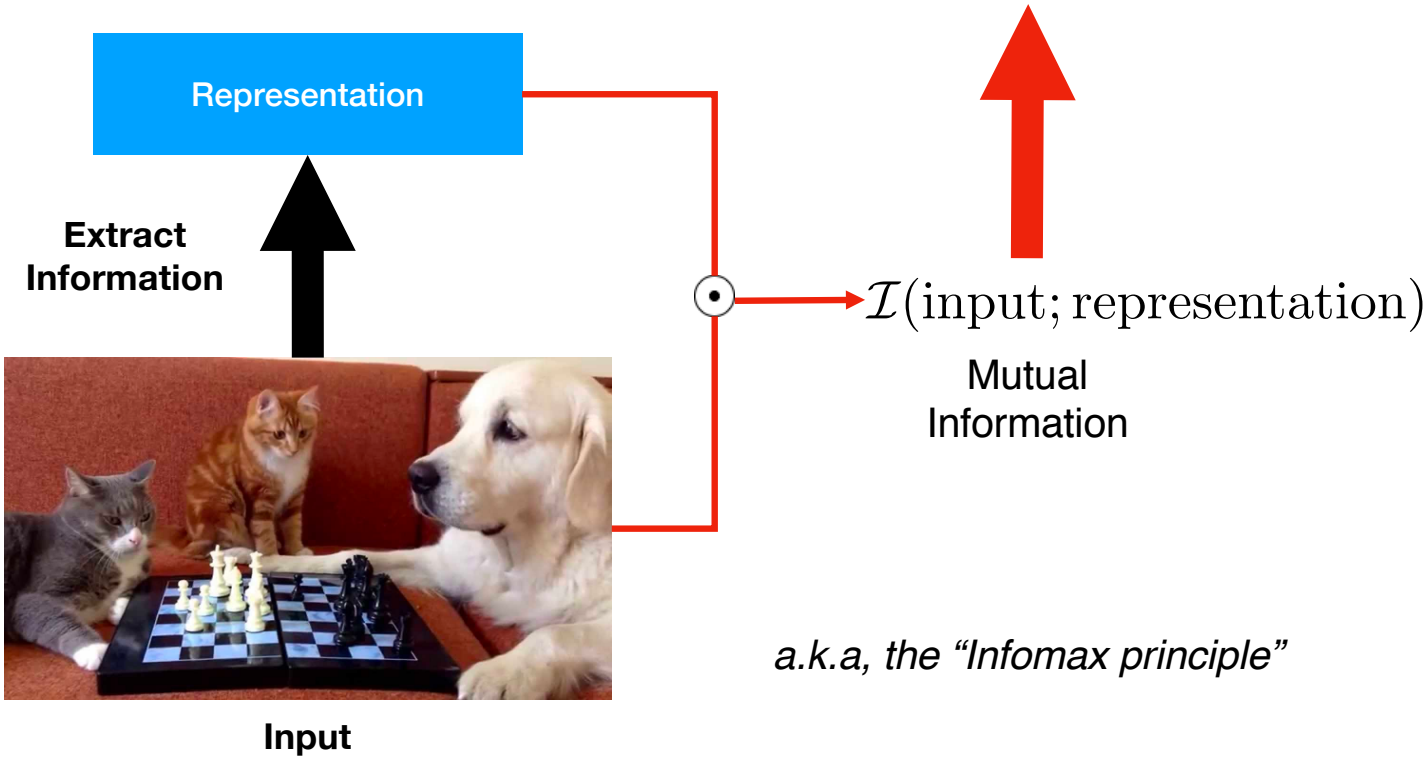
[Belghazi et. al., 2018]

Demonstration of estimation



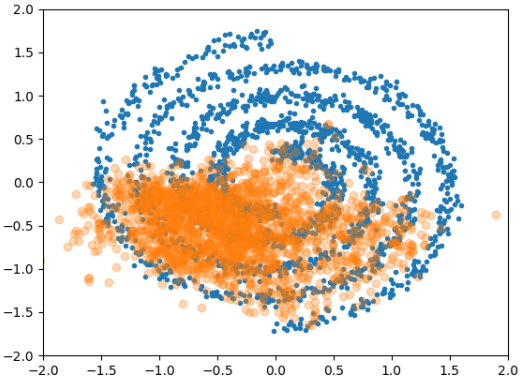
[Belghazi et. al., 2018]

Mutual Information for Representation Learning

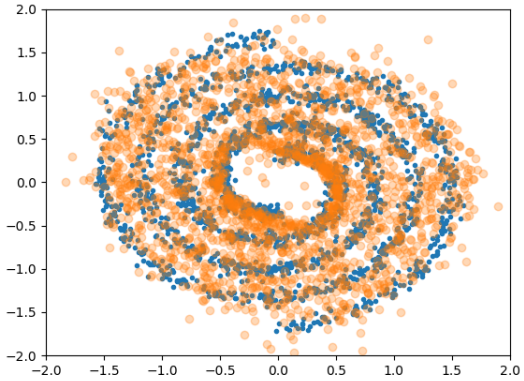


Maximizing mutual information: avoid GAN mode dropping by max MI(X,Z)

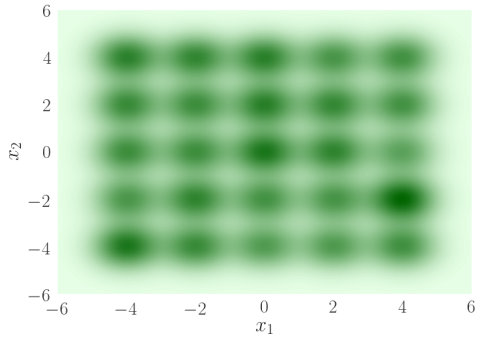
GAN



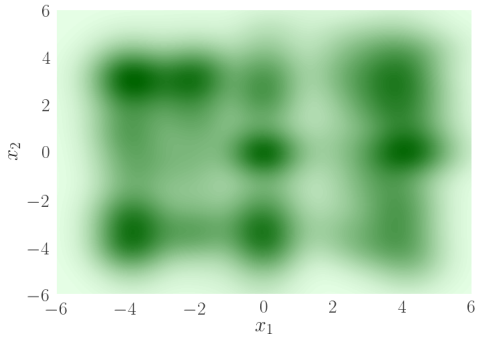
GAN+MINE



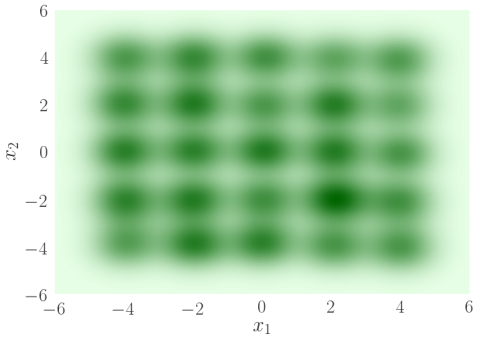
Ground Truth



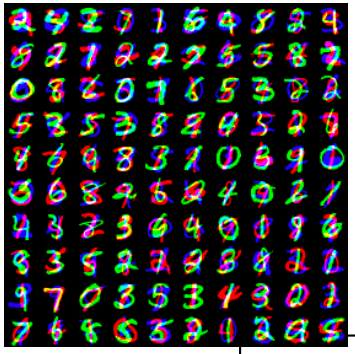
GAN



GAN+MINE



[Belghazi et. al., 2018]



Maximizing mutual information (stacked MNIST)

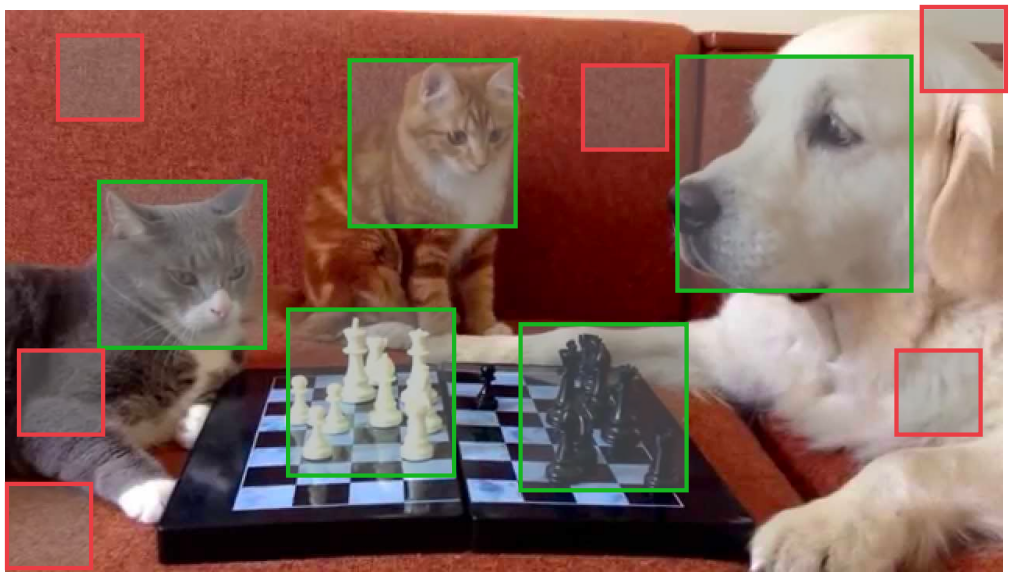
	Modes (max 1000)	$\mathcal{D}_{KL}(\mathbb{P}_Y \| \mathbb{Q}_Y)$
DCGAN	99	3,4
ALI	16	5,4
Unrolled GAN	48,7	4,32
VEEGAN	150	2,96
PacGAN	1000	0,6
DCGAN+MINE	1000	0,5

[Belghazi et. al., 2018]

We don’t necessarily need generation in pixel space

Interesting thing

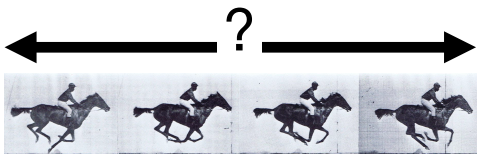
Not interesting thing



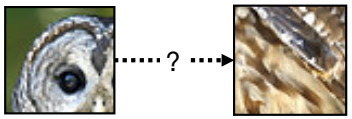
Generative models (in principle) care about all the pixels

Self-supervision in the wild

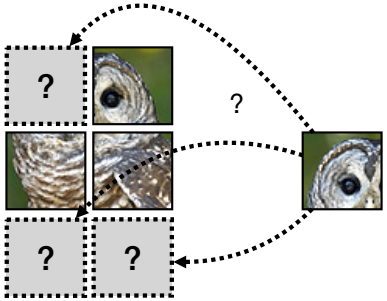
- **Question:** What direction is the video running?



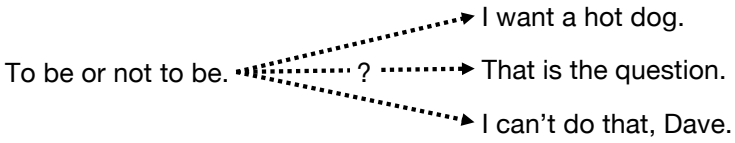
- **Question**: Do these image patches go together (context prediction)?



- **Question**: Where does this patch go (jigsaws)?

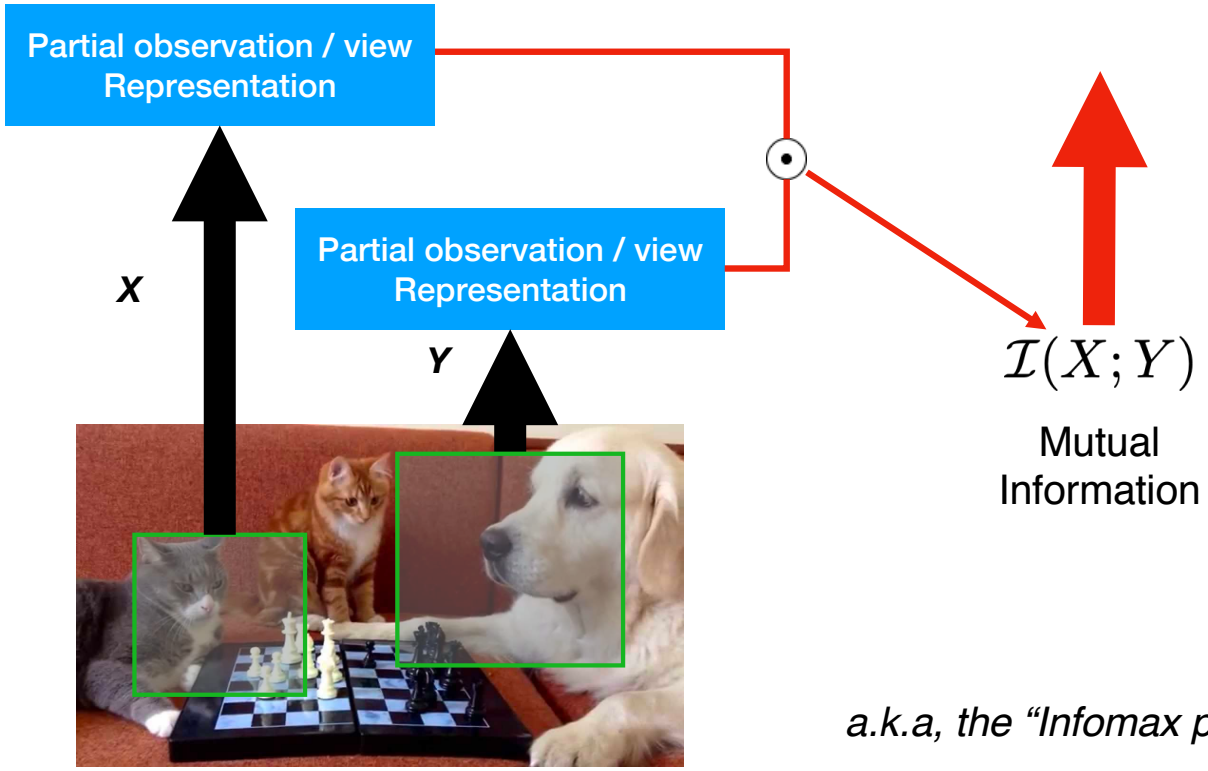


- **Question**: Which sentence follows this first one (Quick-thoughts)?



e.g., see Logeswaran et al. 2018, Doersch et al. 2015, 2017, Wei et al. 2018

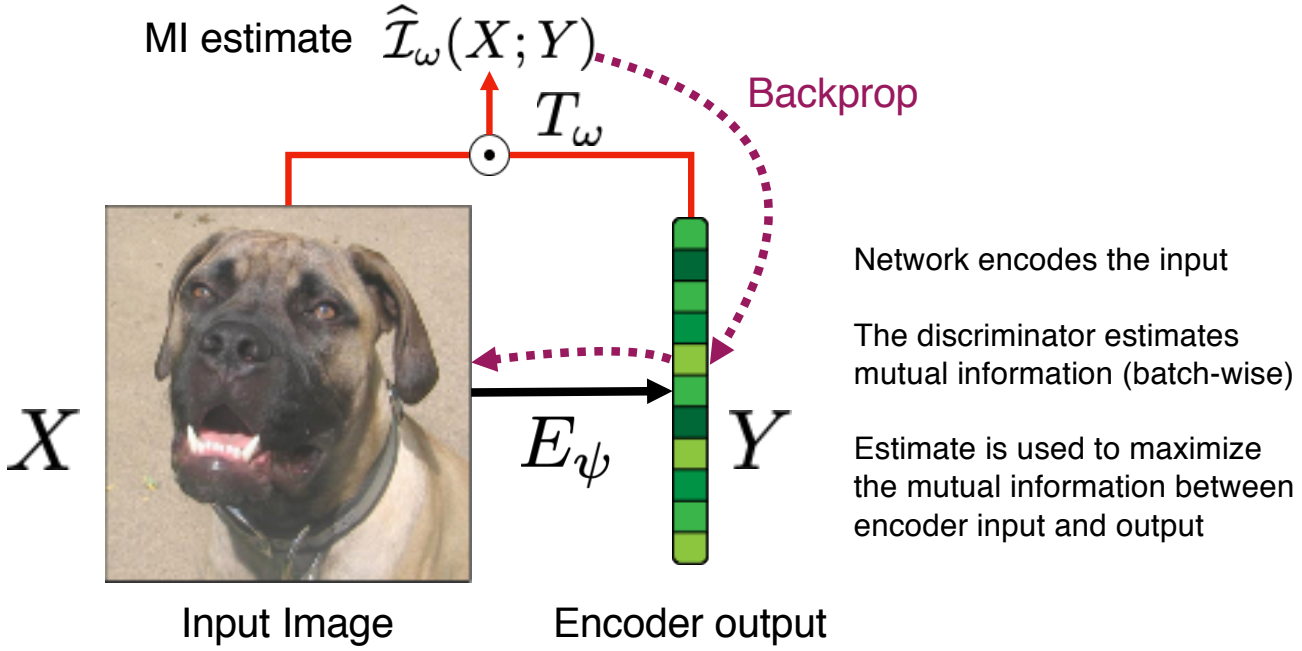
Mutual Information for Self-Supervised Representation Learning



a.k.a, the "Infomax principle"

Ask questions about the data at the representation / feature level

Maximizing mutual information in encoders



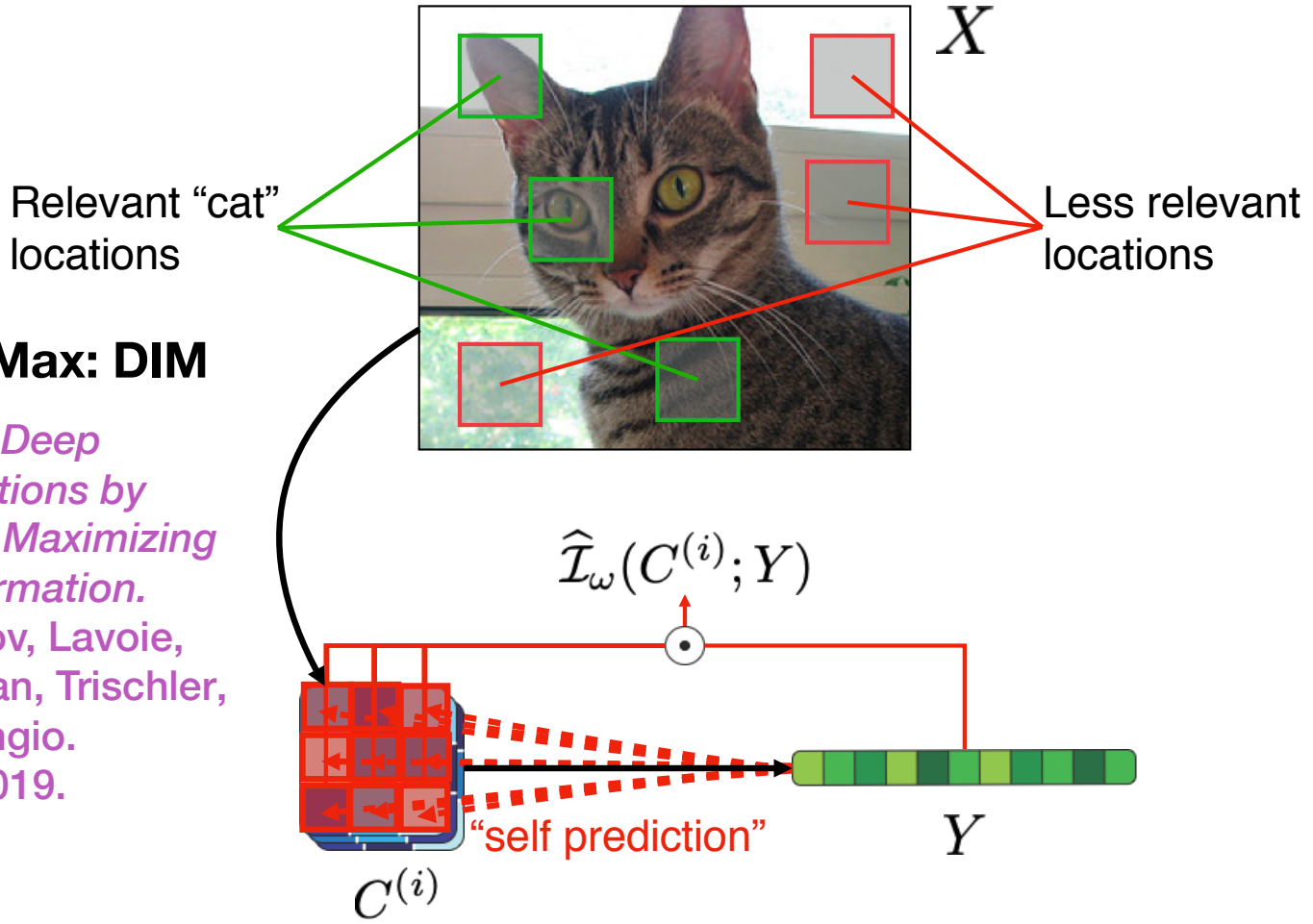
Network encodes the input

The discriminator estimates mutual information (batch-wise)

Estimate is used to maximize the mutual information between encoder input and output

Denoising auto-encoders reconstruction error: weaker lower bound on MI(input, representation)
(Vincent et al ICML 2008)

Using local structure is crucial



Deep Info Max: DIM

Learning Deep Representations by Estimating and Maximizing Mutual Information.
Hjelm, Foderov, Lavoie, Grewal, Bachman, Trischler, and Bengio.
ICLR 2019.

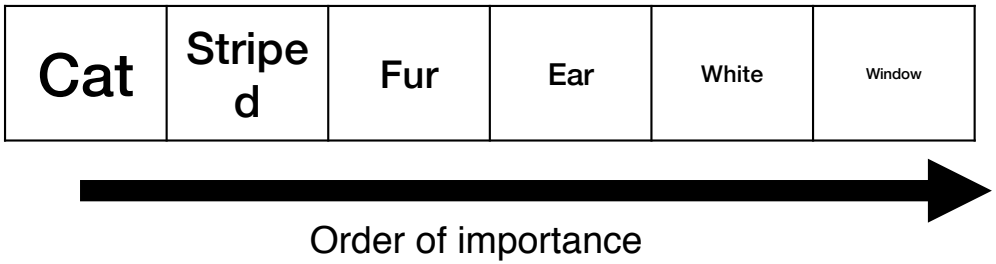
Using local structure is crucial

Local Feature Vectors

Cat, Ear, Sky	**Cat**, Ear, Sky	White, Sky, Bright	**Cat**, Ear, Pink	White, Sky, Bright	White, Sky, Bright
Cat, Ear, Wood	**Cat**, Ear, Pink	**Cat**, Fur, Striped	**Cat**, Ear, Striped	White, **Cat**, Window	White, Tan, Window
Window, Tan, Wood	**Cat**, eye, yellow	**Cat**, nose, striped	**Cat**, Eye, Yellow	**Cat**, Window, Wood	Window, Tan, Wood
Window, Tan, Wood	**Cat**, Window, Fur	**Cat** nose, pink	**Cat**, Whisker, Striped	**Cat**, Window, Fur	Window, Tan, Wood
Leaves, Green, Window	Leaves, Green, Window	**Cat**, Fur, Striped	**Cat**, Fur, Striped	**Cat**, Fur, Whisker	**Cat**, Fur, Striped
Leaves, Green, Tree	Leaves, Green, Tree	**Cat**, Fur, Striped	**Cat**, Fur, Striped	**Cat**, Fur, Striped	**Cat**, Fur, Striped

Maximizes the average mutual information across locations

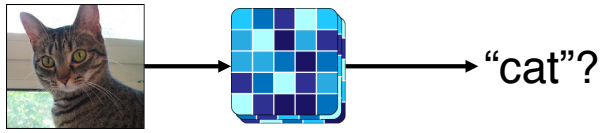
Global Feature Vector



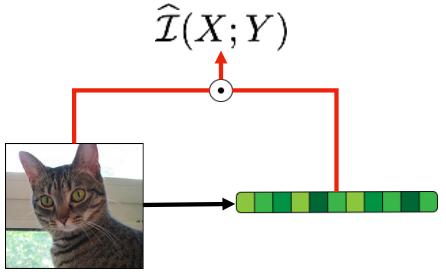
Evaluating the representations

- Linear / Nonlinear classifier on global features

- Linear / Nonlinear classifier on local features

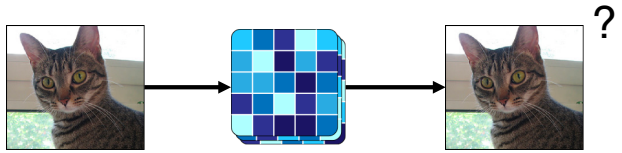


- Measuring mutual information (MINE)



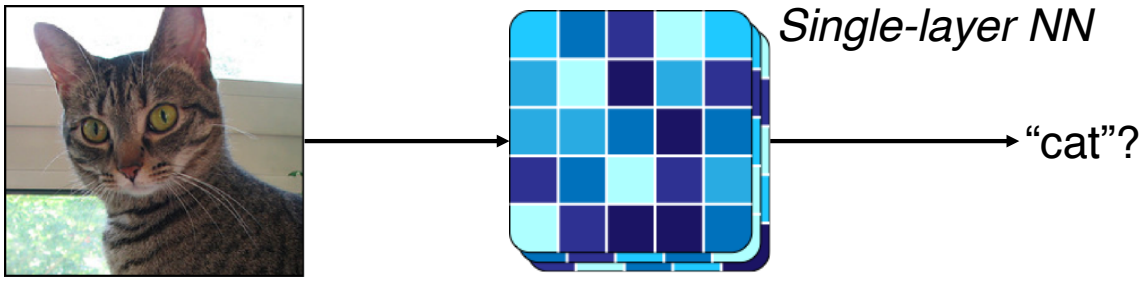
- Measuring dependence (NDM)

- Measuring reconstruction (MS-SSIM)



Hjelm et. al., ICLR 2019

Classification evaluation results



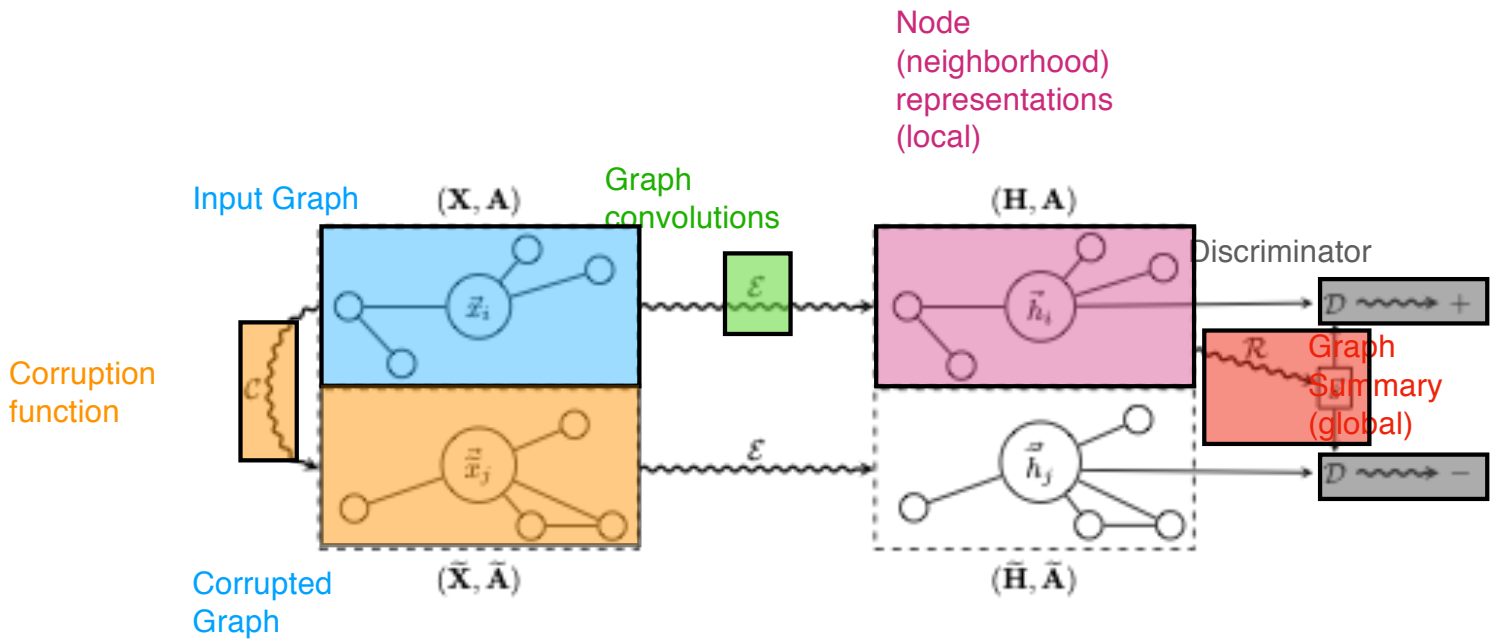
Model	CIFAR10	CIFAR100	Tiny Imagenet	STL10
Fully supervised	75,39 %	42,27	36,60	68,7
VAE	60,71	37,21	18,63	58,27
AAE	59,44	36,22	18,04	59,54
ALI/BiGAN	62,57	37,59	24,38	71,53
NAT	56,19	29,18	13,70	64,32
DIM(MINE)	**72,66**	**48,52**	30,35	69,15
DIM(JSD)	**73,25**	**48,13**	**33,54**	**72,86**
DIM(infoNCE)	**75,21**	**49,74**	**34,21**	**72,57**

Classification accuracy

Hjelm et. al., ICLR 2019

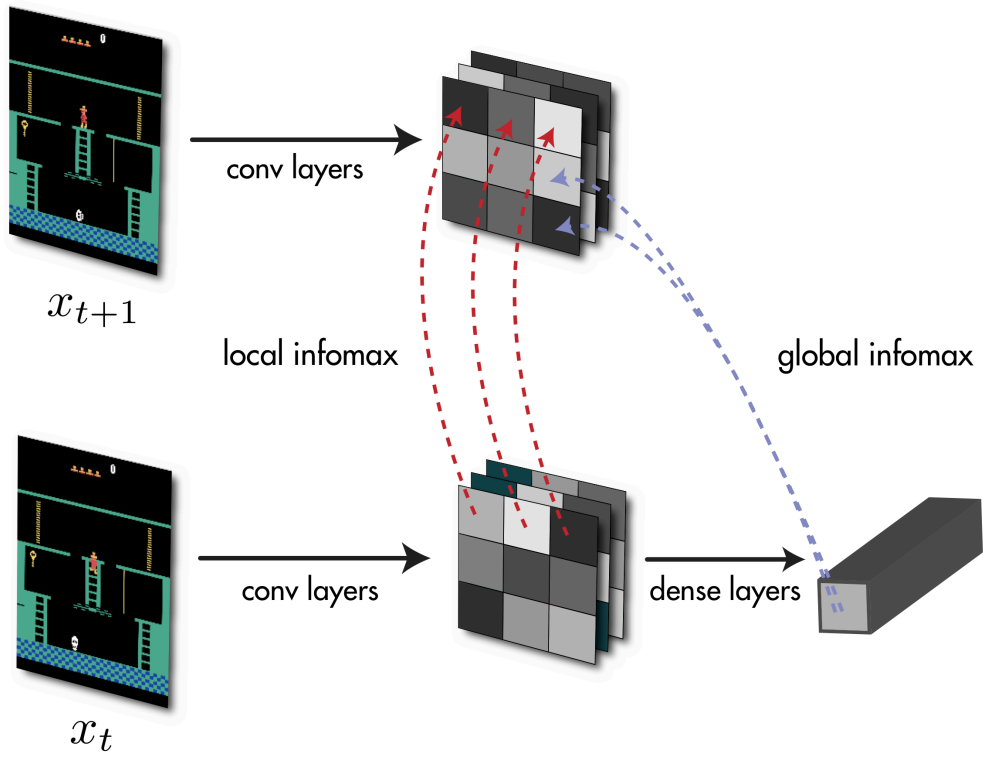
DIM on Graphs: Deep Graph Infomax (DGI)

Deep InfoMax extends easily to other types of data.



Velickovic et. al., ICLR 2019

Spatio-Temporal DIM (STDIM) for Atari



Anand, Racah, Ozair et. al., ICML workshop on self-supervised learning, 2019

Thanks!

Code for Deep InfoMax: https://github.com/rdevon/DIM

Code for Deep Graph InfoMax: https://github.com/PetarV-/DGI

Some Works covered:

- *Learning Independent Features with Adversarial Nets for Non-linear ICA.*
  Brakel, Philemon & Bengio, Y. (2017).
- *Mutual Information Neural Estimation.*
  Belghazi, Baratin, Ozair, Rajeswar, Bengio, Courville, and Hjelm.
  ICML 2018.
- *Learning Deep Representations by Estimating and Maximizing Mutual Information.*
  Hjelm, Foderov, Lavoie, Grewal, Bachman, Trischler, and Bengio.
  ICLR 2019.
- *Deep Graph Infomax (DGI).*
  Veličković, Fedus, Hamilton, and Hjelm.
  ICLR 2019.
- *Unsupervised State Representation Learning in Atari*.
  Anand*, Racah*, Ozair*, Bengio, Côté, and Hjelm.
  Workshop for Self-supervised Learning, ICML 2019.