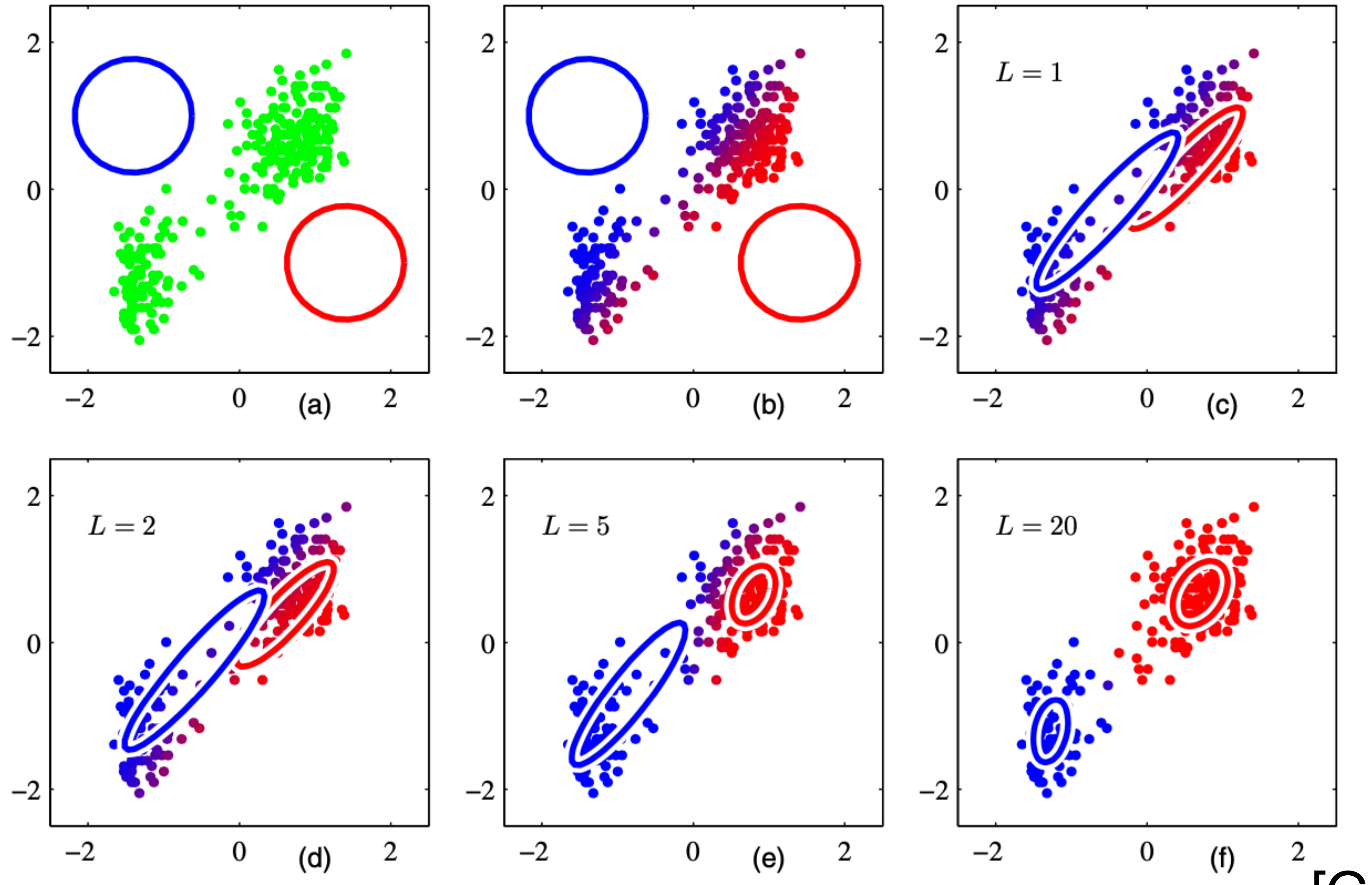
SELF-SUPERVISED REPRESENTATION LEARNING

Jonathan Gordon and Javier Antorán

Self Supervised Learning (SSL)

Unsupervised Learning:

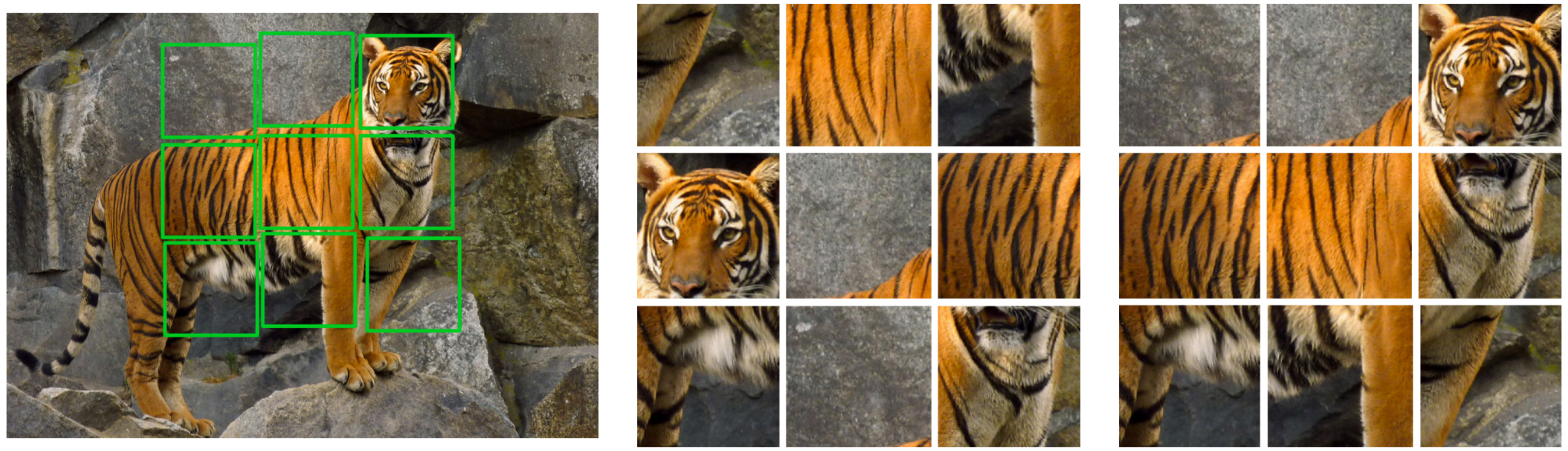
Learning to Model the Observed Data



[C. Bishop]

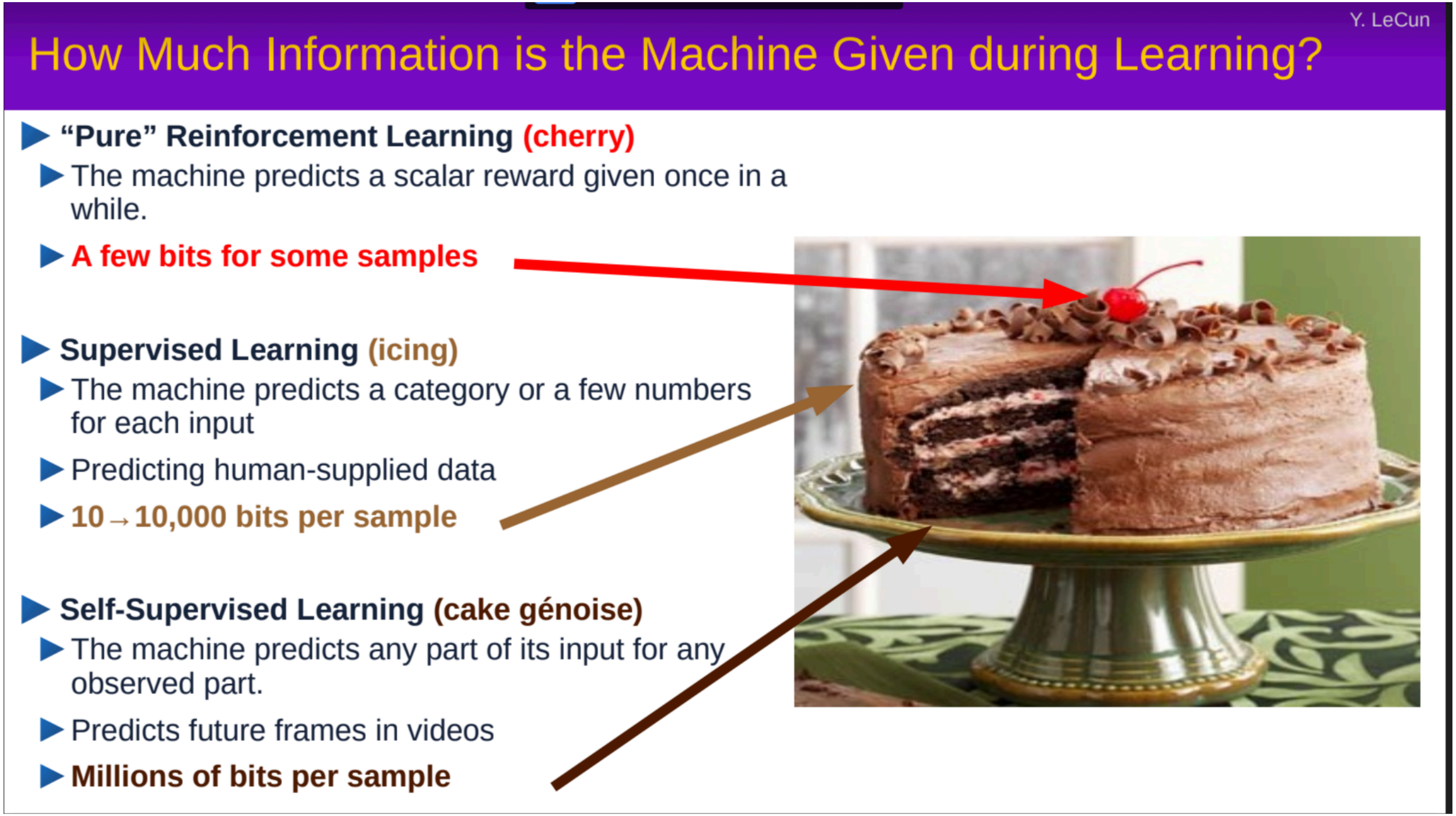
Self Supervised Learning:

Learning by Predicting Proxy Targets
Extracted From Unlabelled Data



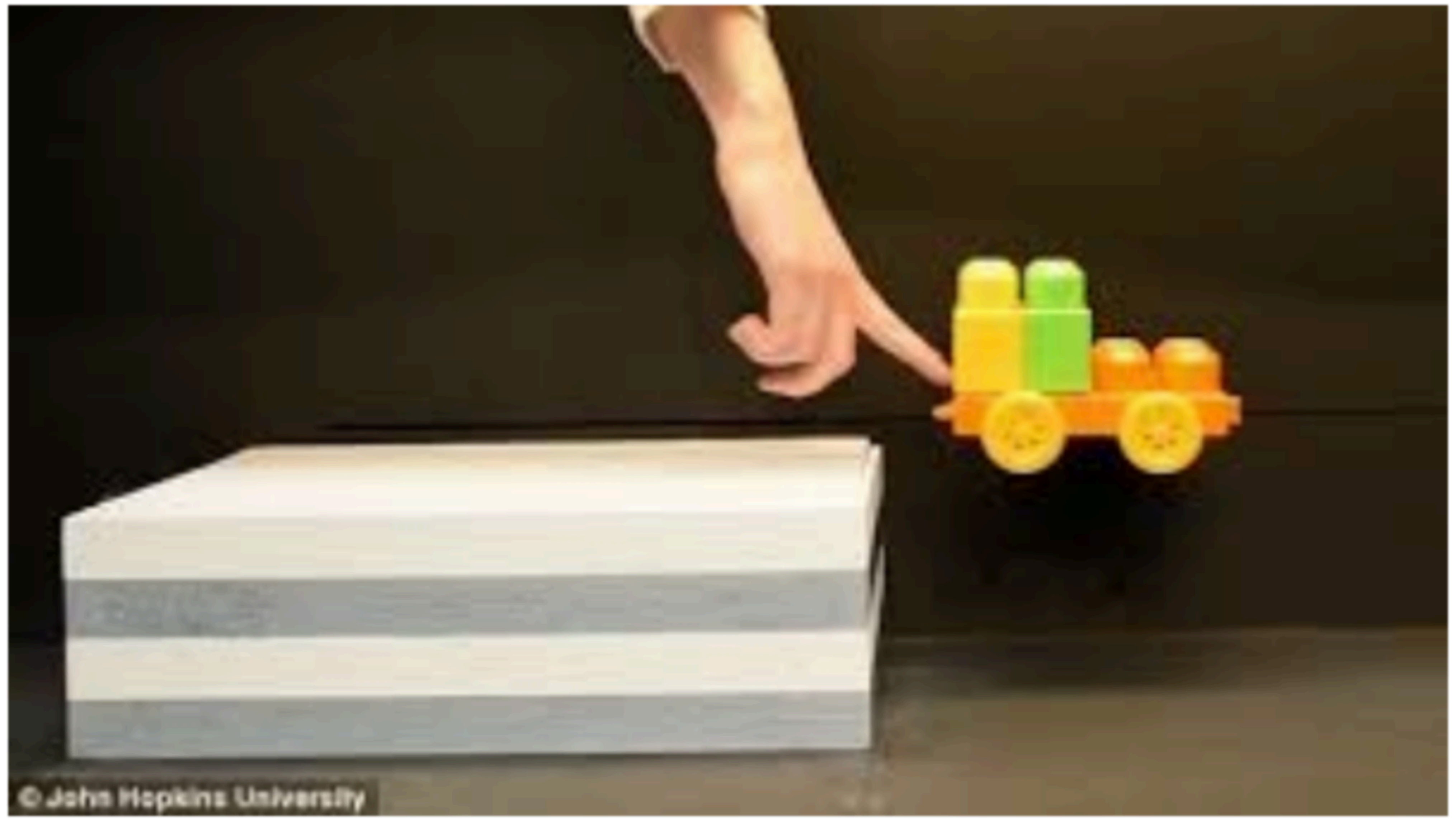
[Larsson et. al.]

But Why Self Supervised Learning?

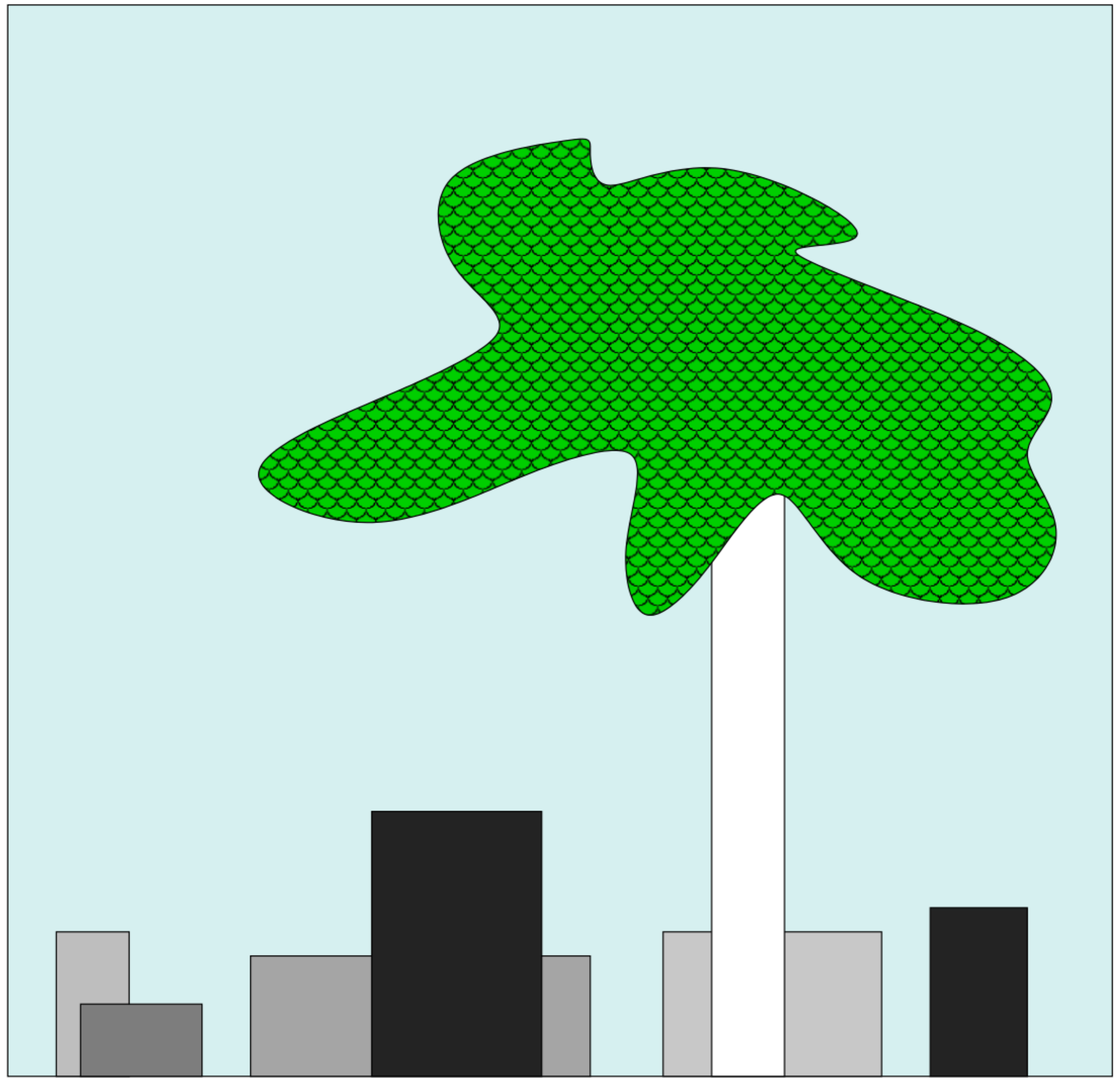


[Y. LeCun]

But Why Self Supervised Learning?



[Y. LeCun]



[D. MacKay]

In Other Words:

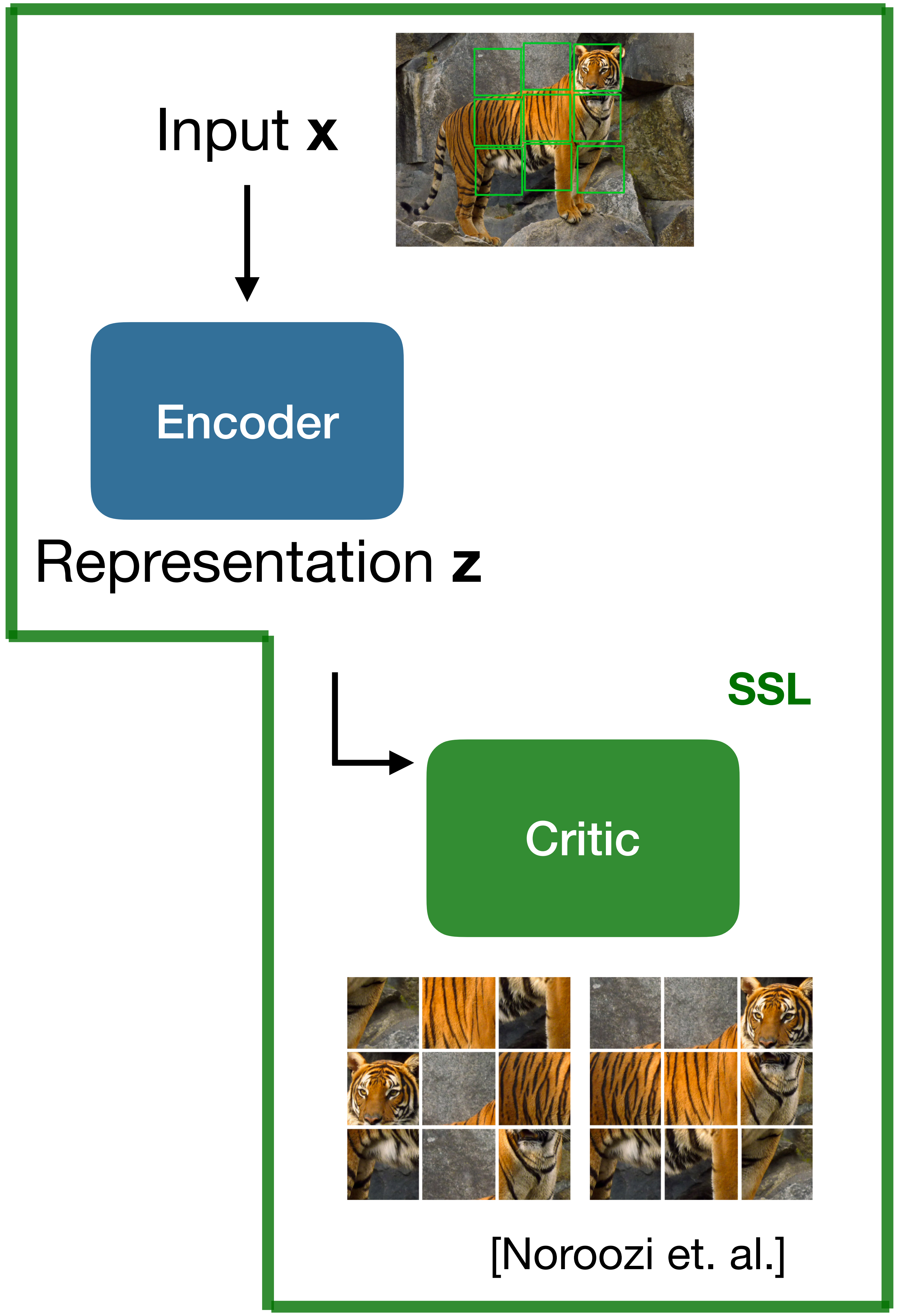
[Y. LeCun]

Talk Layout

1. Examples of Self Supervised Learning (SSL) Approaches
    - Information Theoretic Interpretation
2. Mutual Information (MI) Motivated SSL approaches
    - Deep InfoMax
    - Contrastive Predictive Coding
3. Is MI the real reason behind the success of SSL?
4. Generative Models for Representation Learning
5. Self Supervised Learning for Identifiability in Non-Linear ICA

Typical SSL Approaches:

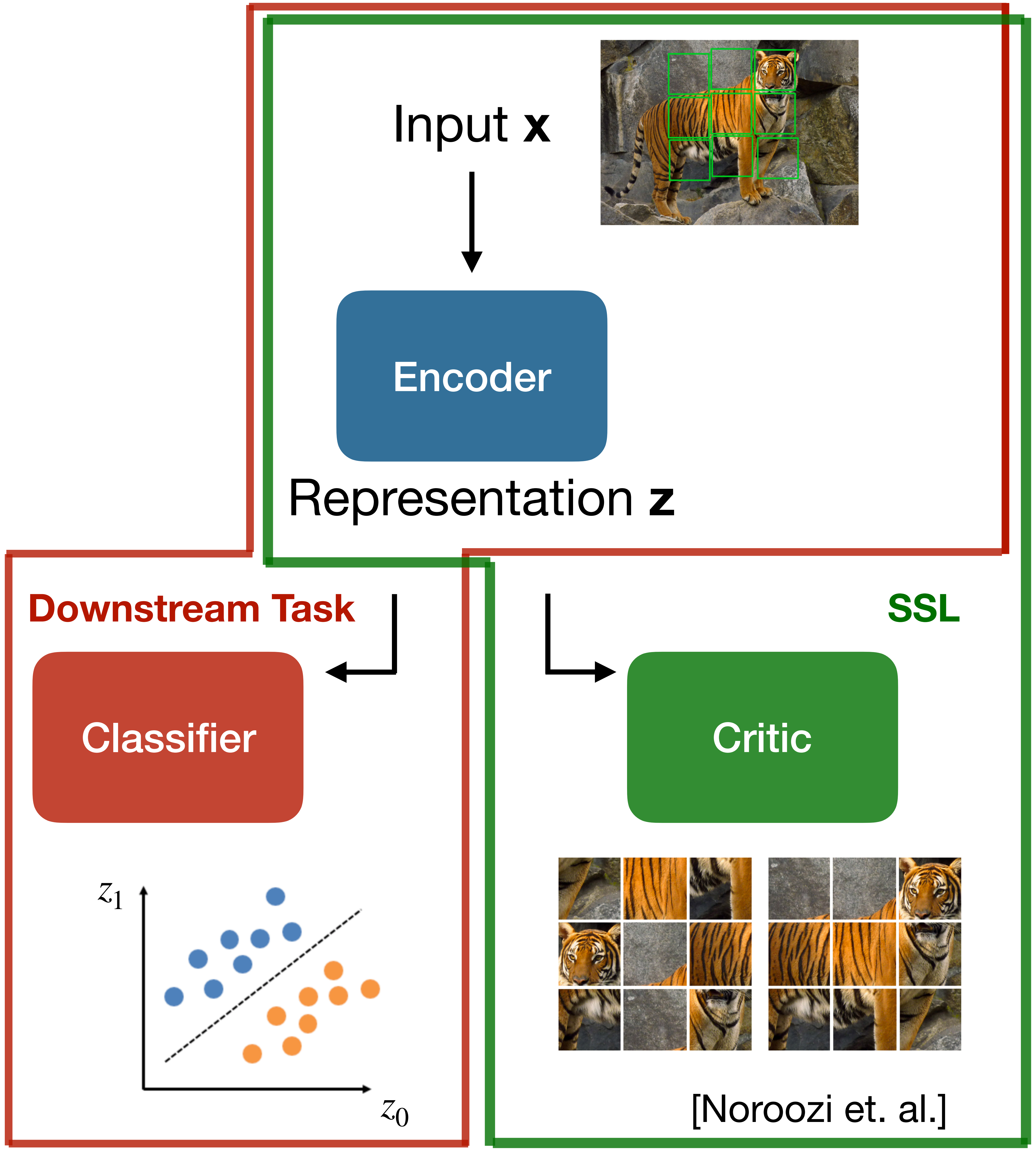
- Predict Future From Past
- Predict Adjacent Sections in Structured Data
- Predict Occluded Area from Non-Occluded One
- Undo Data-Augmentation Transformations



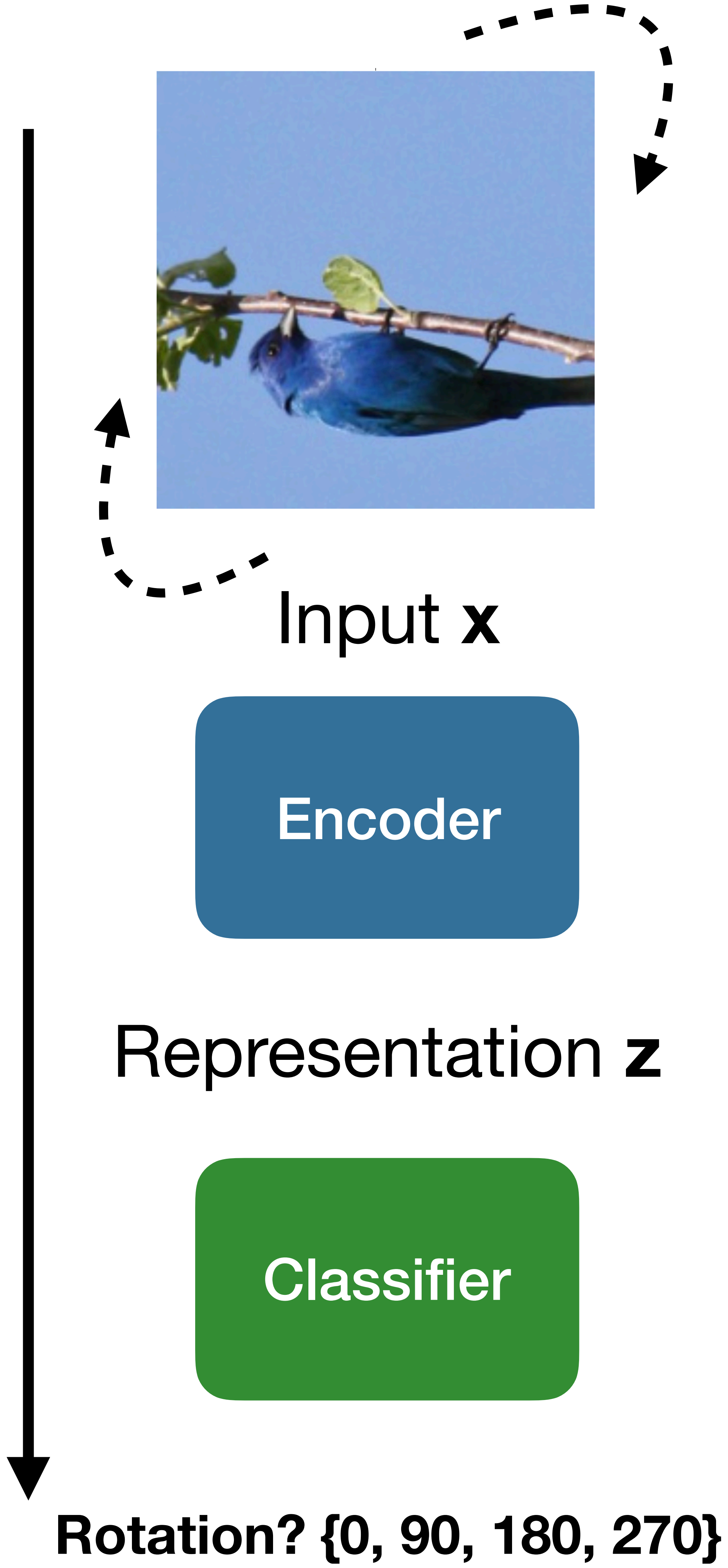
Typical SSL Approaches:

- Predict Future From Past
- Predict Adjacent Sections in Structured Data
- Predict Occluded Area from Non-Occluded One
- Undo Data-Augmentation Transformations

Obtain gains in downstream tasks (usually classification)



Predicting Rotations



CNNs are not rotation invariant.
Will need to learn part-whole relationships.

Supervised Attention

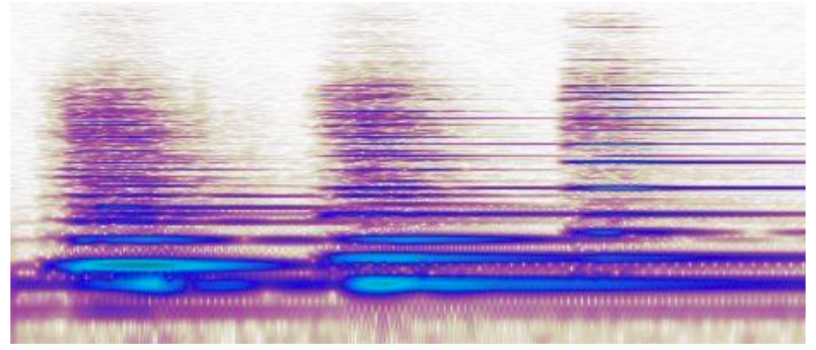
Self-Supervised Attention

[Gidaris et. al.]

Multimodal SSL: Audio + Video

Video **x_v**

Audio **x_a**



Encoder v

Encoder v

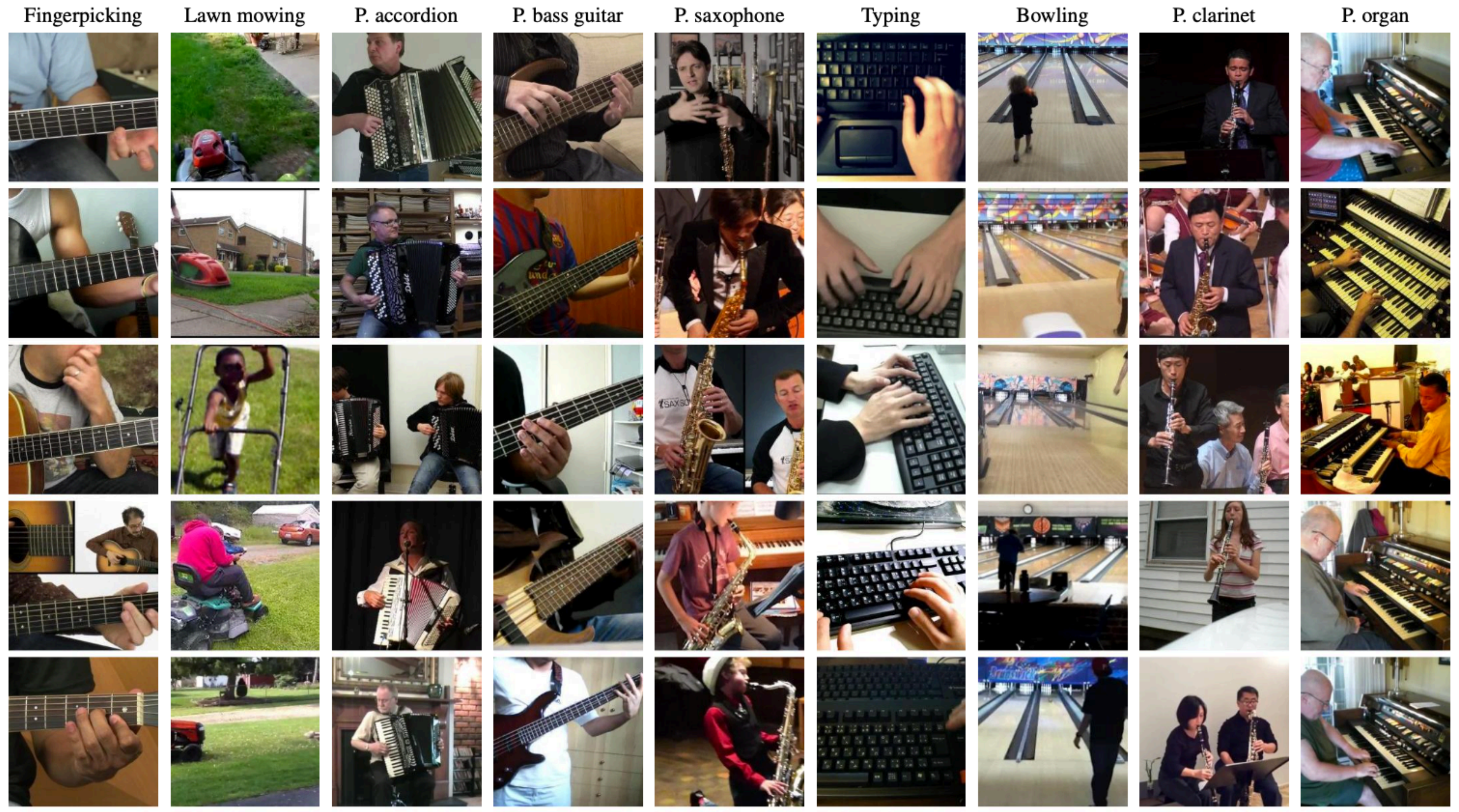
z_v

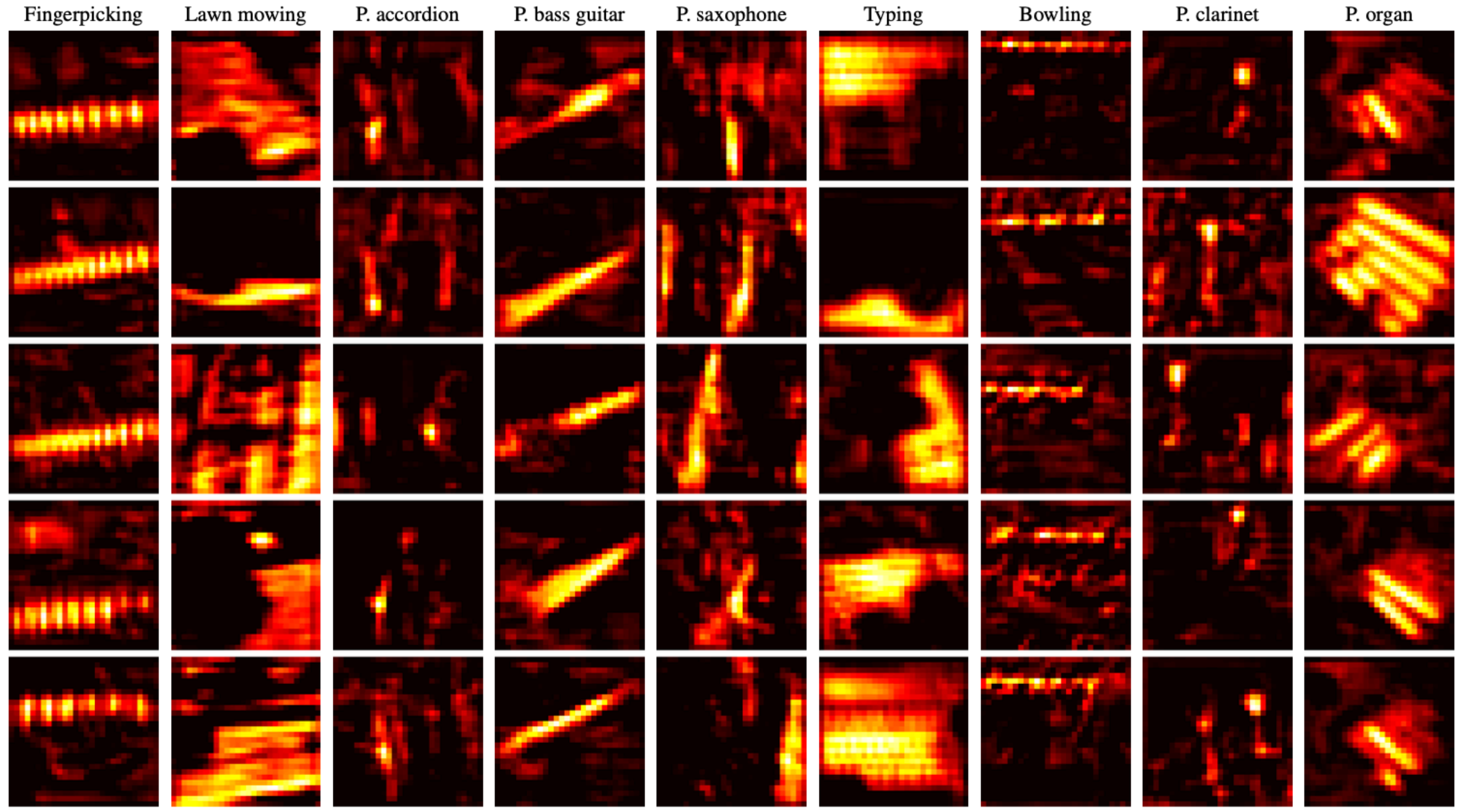
z_a

Classifier

a-v Correspondence?: {Y, N}

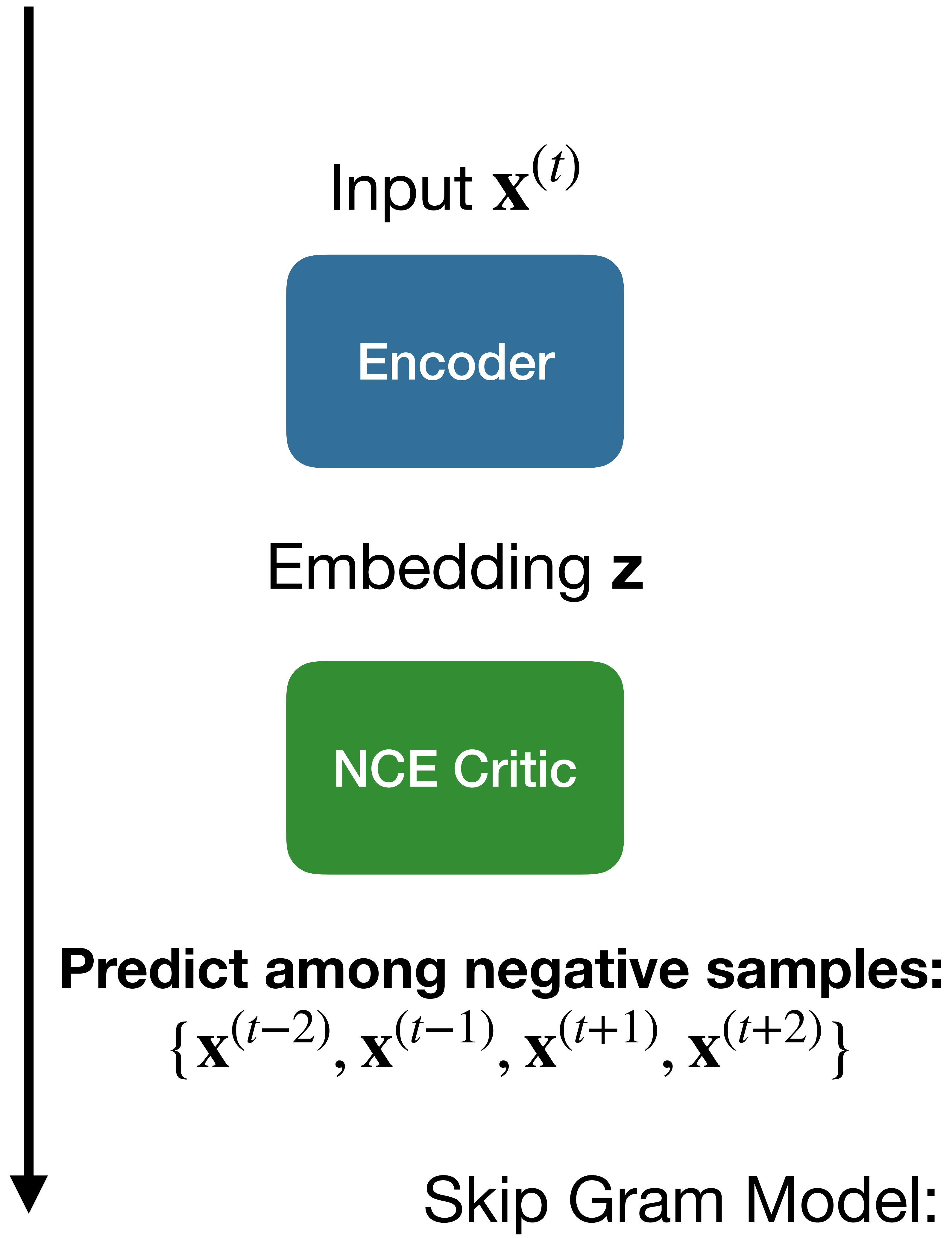
Separate $p(\mathbf{a}, \mathbf{v})$ from $p(\mathbf{a})p(\mathbf{v})$





[Arandjelovic et. al.]

Word Embeddings



Unsupervised Analogical Reasoning

Newspapers			
New York San Jose	New York Times San Jose Mercury News	Baltimore Cincinnati	Baltimore Sun Cincinnati Enquirer
NHL Teams			
Boston Phoenix	Boston Bruins Phoenix Coyotes	Montreal Nashville	Montreal Canadiens Nashville Predators
NBA Teams			
Detroit Oakland	Detroit Pistons Golden State Warriors	Toronto Memphis	Toronto Raptors Memphis Grizzlies
Airlines			
Austria Belgium	Austrian Airlines Brussels Airlines	Spain Greece	Spainair Aegean Airlines
Company executives			
Steve Ballmer Samuel J. Palmisano	Microsoft IBM	Larry Page Werner Vogels	Google Amazon

Skip Gram Model: Separating $p(\mathbf{x}^{(t)}, \mathbf{x}^{(t+k)})$ from $p(\mathbf{x}^{(t)})p(\mathbf{x}^{(t+k)})$

[Mikolov et. al.]

Typical SSL Approaches:

- ~~Predict Future From Past~~
- ~~Predict Adjacent Sections in Structured Data~~
- ~~Predict Occluded Area from Non-Occluded One~~
- ~~Undo Data-Augmentation Transformations~~

Maximise Mutual Information Between Inputs and Representations?

$$I(\mathbf{x}, \mathbf{z}) = E_{p(\mathbf{x},\mathbf{z})}[\log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}]$$

But What is Mutual Information?

But What is Mutual Information?

$$I(\mathbf{x}, \mathbf{z}) = KL(p(\mathbf{x}, \mathbf{z}) || p(\mathbf{x})p(\mathbf{z})) = E_{p(\mathbf{x},\mathbf{z})}[\log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}]$$

But What is Mutual Information?

$$I(\mathbf{x}, \mathbf{z}) = KL(p(\mathbf{x}, \mathbf{z}) || p(\mathbf{x})p(\mathbf{z})) = E_{p(\mathbf{x},\mathbf{z})}[\log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}]$$

$$I(\mathbf{x}, \mathbf{z}) = H(\mathbf{x}) - H(\mathbf{x} | \mathbf{z})$$

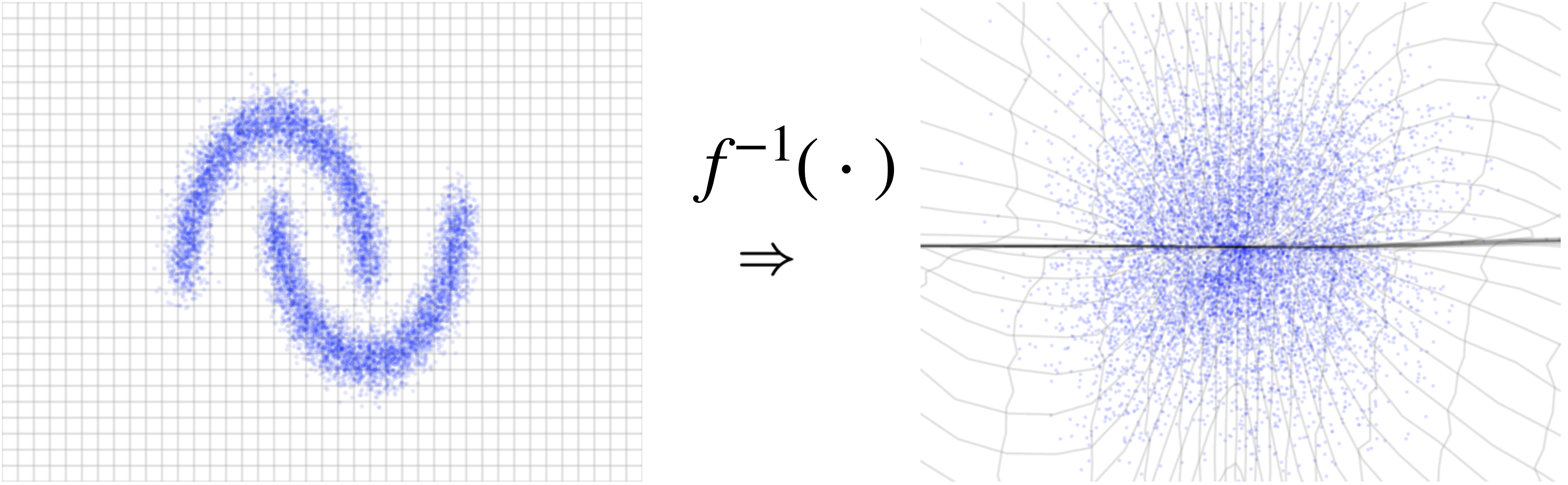
But What is Mutual Information?

$$I(\mathbf{x}, \mathbf{z}) = KL(p(\mathbf{x}, \mathbf{z}) || p(\mathbf{x})p(\mathbf{z})) = E_{p(\mathbf{x},\mathbf{z})}[\log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}]$$

$$I(\mathbf{x}, \mathbf{z}) = H(\mathbf{x}) - H(\mathbf{x} | \mathbf{z})$$

- Symmetric, Bounded: $0 \leq I(\mathbf{x}, \mathbf{y}) = I(\mathbf{y}, \mathbf{x}) \leq \min(H(\mathbf{x}), H(\mathbf{y}))$
- Invariant to Reparametrisations!

Here: $I(\mathbf{x}, \mathbf{z}) = H(\mathbf{x}) = H(\mathbf{z})$ $\longrightarrow$



z **x**

[Dinh et. al.]

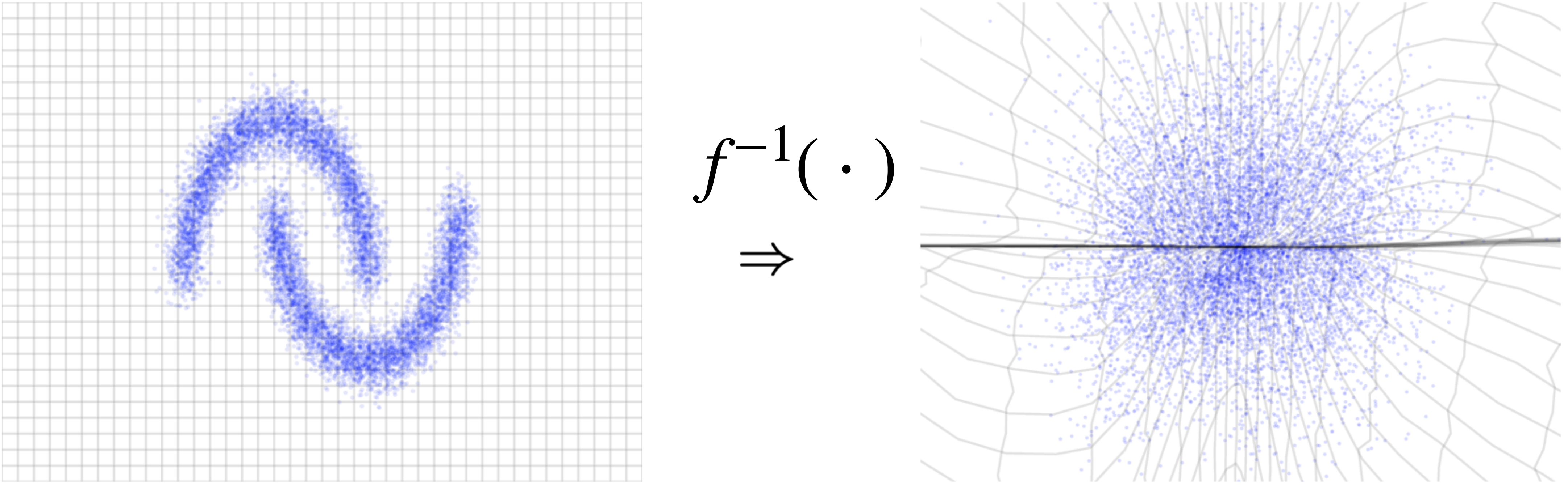
But What is Mutual Information?

$$I(\mathbf{x}, \mathbf{z}) = KL(p(\mathbf{x}, \mathbf{z}) \,||\, p(\mathbf{x})p(\mathbf{z})) = E_{p(\mathbf{x},\mathbf{z})}[\log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}]$$

$$I(\mathbf{x}, \mathbf{z}) = H(\mathbf{x}) - H(\mathbf{x} \,|\, \mathbf{z})$$

- Symmetric, Bounded: $0 \leq I(\mathbf{x}, \mathbf{y}) = I(\mathbf{y}, \mathbf{x}) \leq \min(H(\mathbf{x}), H(\mathbf{y}))$
- Invariant to Reparametrisations!

Here: $I(\mathbf{x}, \mathbf{z}) = H(\mathbf{x}) = H(\mathbf{z})$ $\longrightarrow$



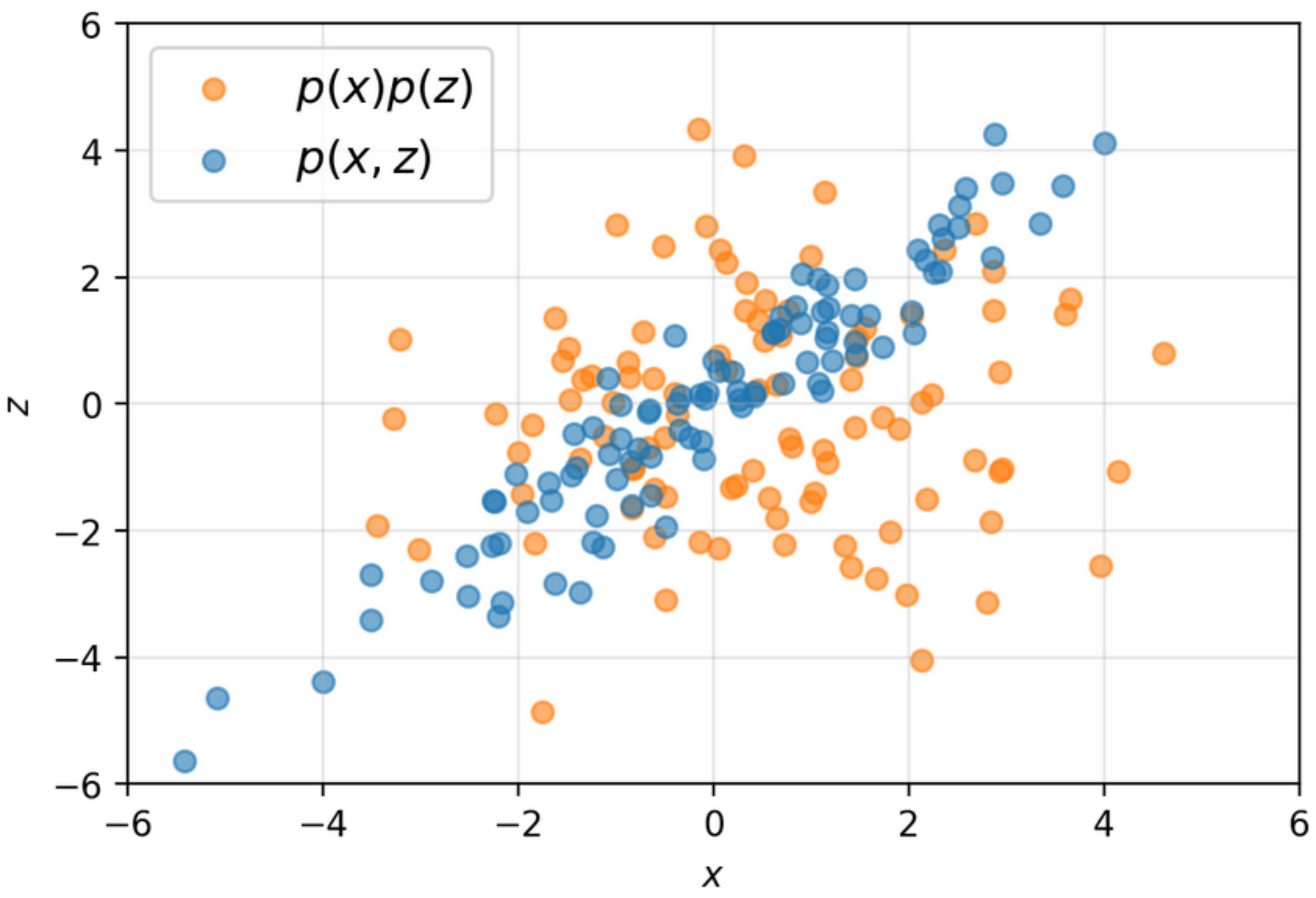
z **x**

[Dinh et. al.]

ISSUE: No closed form for distributions. Only have samples!

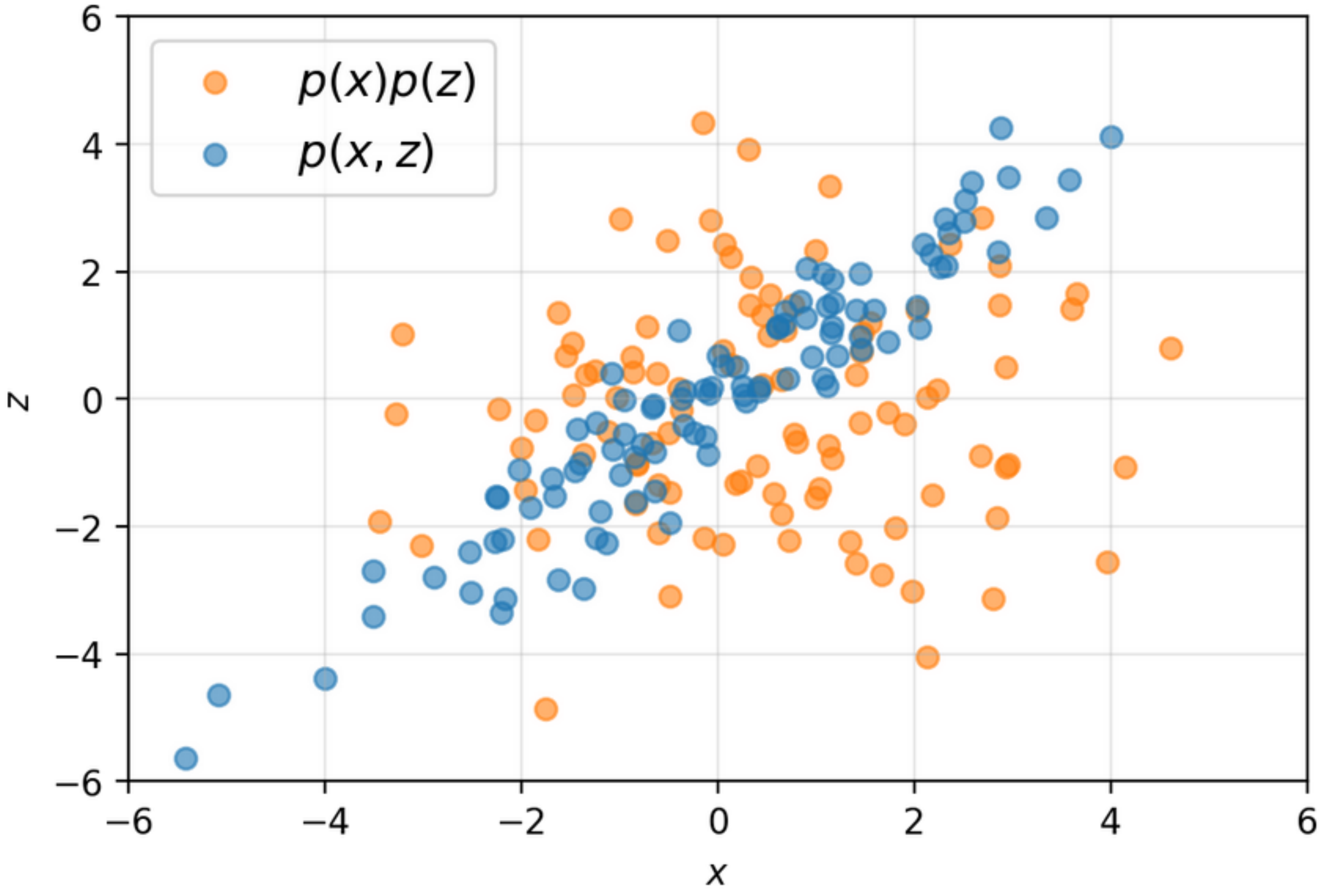
Density Ratio Estimation

$$I(\mathbf{x}, \mathbf{z}) = KL(p(\mathbf{x}, \mathbf{z}) || p(\mathbf{x})p(\mathbf{z})) = E_{p(\mathbf{x},\mathbf{z})}[\log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}]$$



Density Ratio Estimation

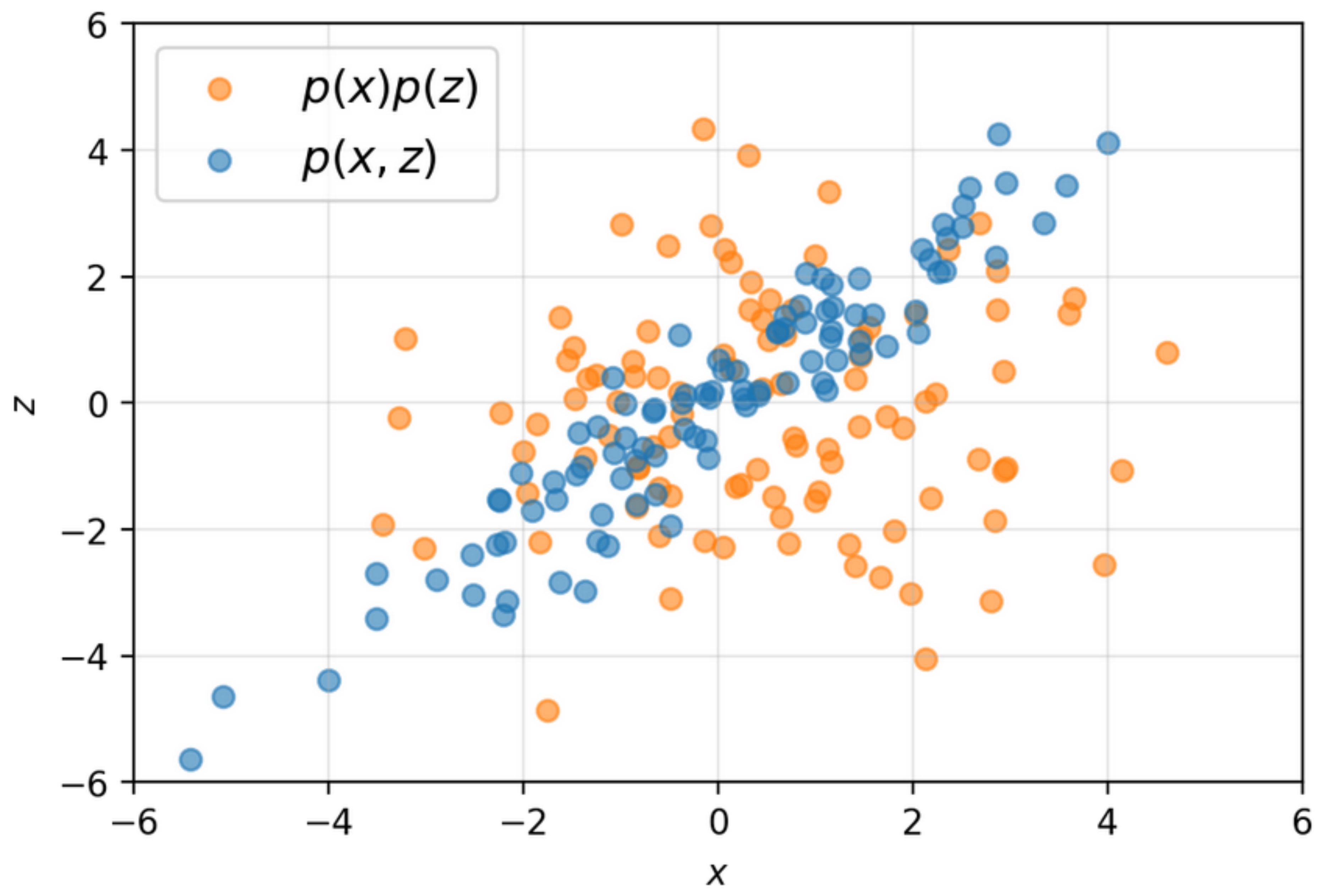
- Lets write $\mathbf{a} = (\mathbf{x}, \mathbf{z}); \quad p(\mathbf{a}\,|\,b=1) = p(\mathbf{x})p(\mathbf{z}); \quad p(\mathbf{a}\,|\,b=0) = p(\mathbf{x}, \mathbf{z})$



Density Ratio Estimation

- Lets write $\mathbf{a} = (\mathbf{x}, \mathbf{z}); \quad p(\mathbf{a} \,|\, b = 1) = p(\mathbf{x})p(\mathbf{z}); \quad p(\mathbf{a} \,|\, b = 0) = p(\mathbf{x}, \mathbf{z})$

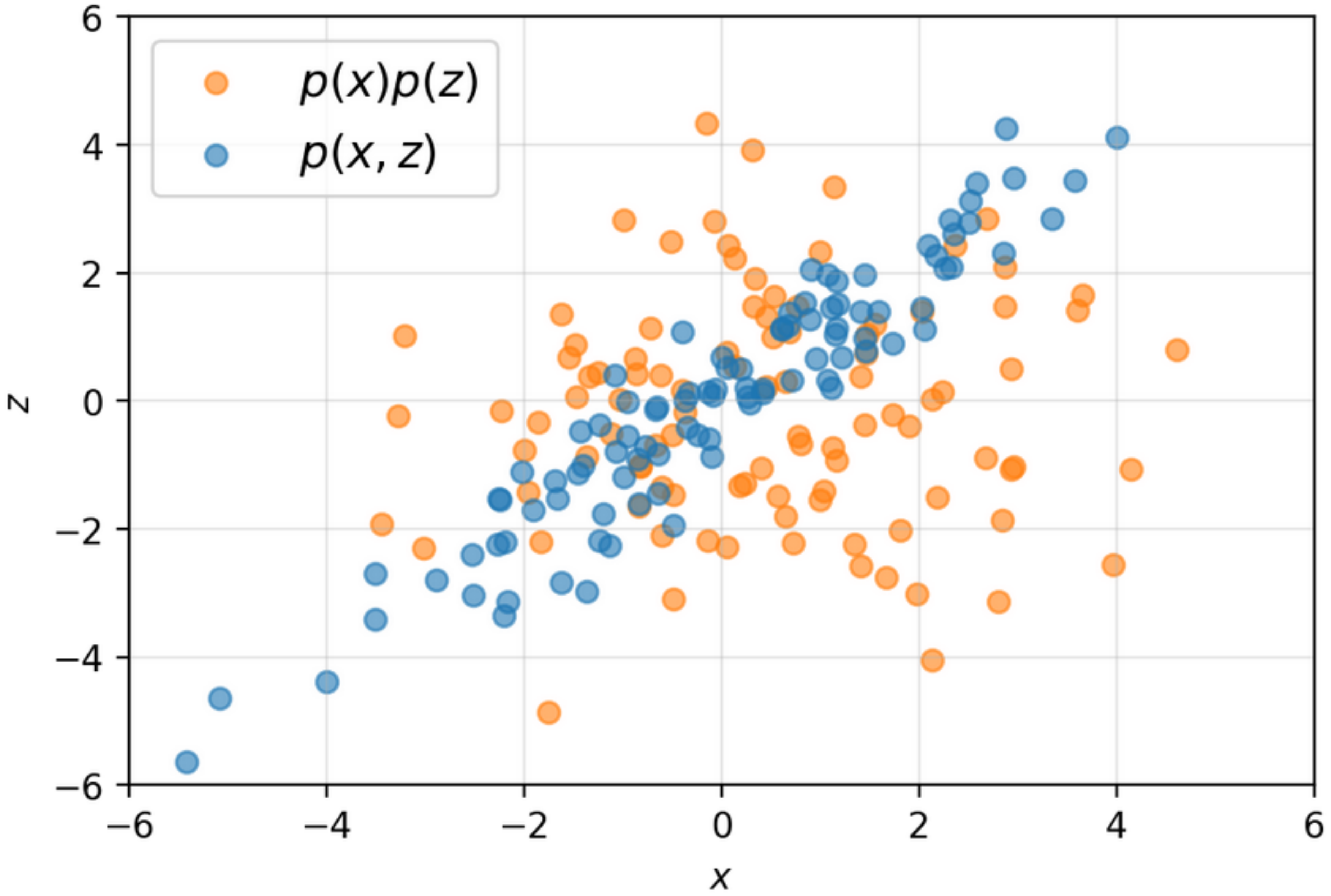
$$r(\mathbf{a}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})} = \frac{p(\mathbf{a} \,|\, b = 0)}{p(\mathbf{a} \,|\, b = 1)}$$



Density Ratio Estimation

- Lets write $\mathbf{a} = (\mathbf{x}, \mathbf{z}); \quad p(\mathbf{a} \,|\, b = 1) = p(\mathbf{x})p(\mathbf{z}); \quad p(\mathbf{a} \,|\, b = 0) = p(\mathbf{x}, \mathbf{z})$

$$r(\mathbf{a}) = \frac{p(\mathbf{a} \,|\, b = 0)}{p(\mathbf{a} \,|\, b = 1)} = \frac{p(b = 0 \,|\, \mathbf{a})p(\mathbf{a})p(b = 1)}{p(b = 1 \,|\, \mathbf{a})p(\mathbf{a})p(b = 0)} = \frac{p(b = 0 \,|\, \mathbf{a})}{p(b = 1 \,|\, \mathbf{a})} = \frac{p(b = 0 \,|\, \mathbf{a})}{1 - p(b = 0 \,|\, \mathbf{a})}$$



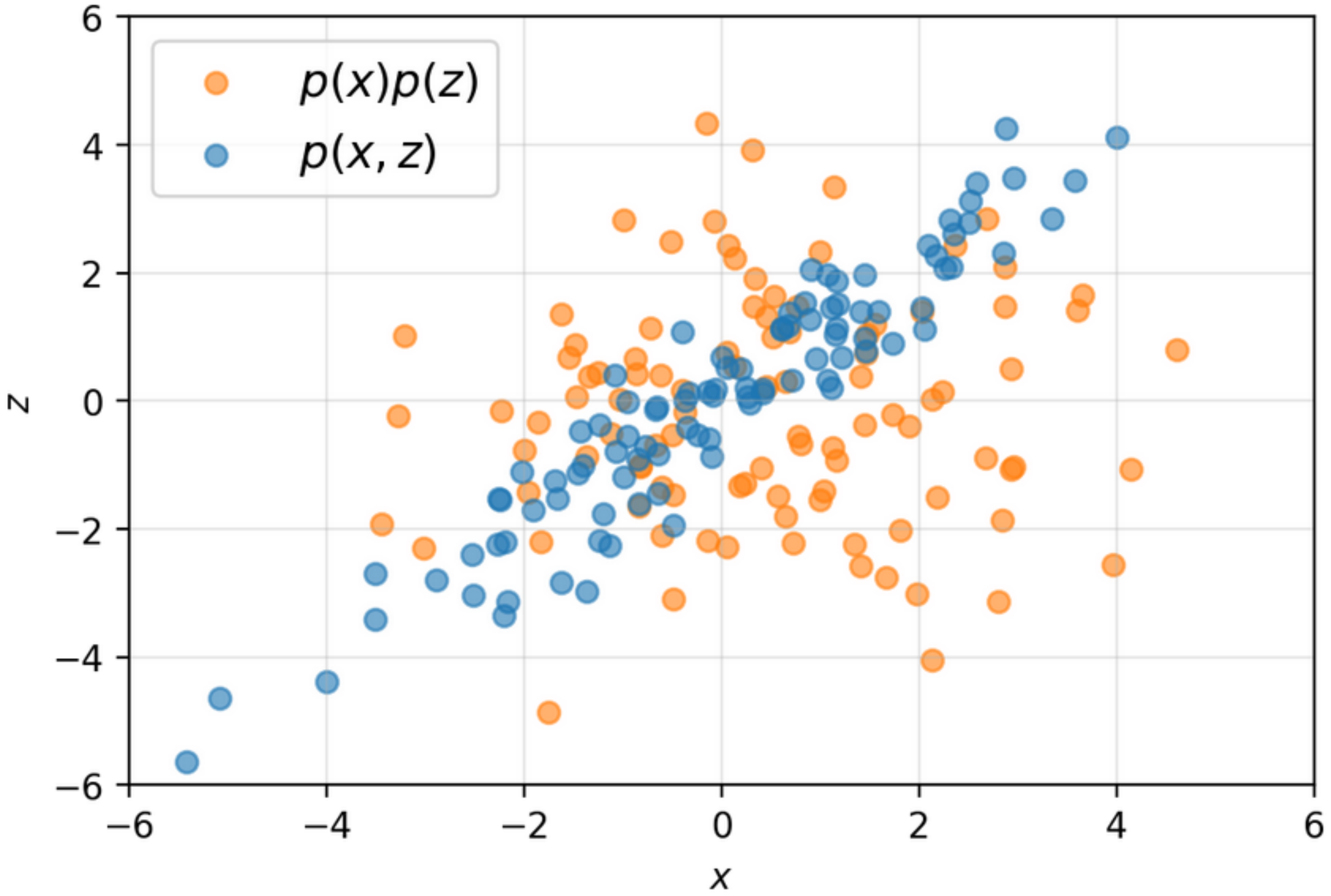
Density Ratio Estimation

- Lets write $\mathbf{a} = (\mathbf{x}, \mathbf{z}); \quad p(\mathbf{a} \,|\, b = 1) = p(\mathbf{x})p(\mathbf{z}); \quad p(\mathbf{a} \,|\, b = 0) = p(\mathbf{x}, \mathbf{z})$

$$r(\mathbf{a}) = \frac{p(\mathbf{a} \,|\, b = 0)}{p(\mathbf{a} \,|\, b = 1)} = \frac{p(b = 0 \,|\, \mathbf{a})p(\mathbf{a})p(b = 1)}{p(b = 1 \,|\, \mathbf{a})p(\mathbf{a})p(b = 0)} = \frac{p(b = 0 \,|\, \mathbf{a})}{p(b = 1 \,|\, \mathbf{a})} = \frac{p(b = 0 \,|\, \mathbf{a})}{1 - p(b = 0 \,|\, \mathbf{a})}$$

We often model conditional probabilities with parametric functions:

$$f_{NN}(\mathbf{a}) = p(b = 0 \,|\, \mathbf{a})$$



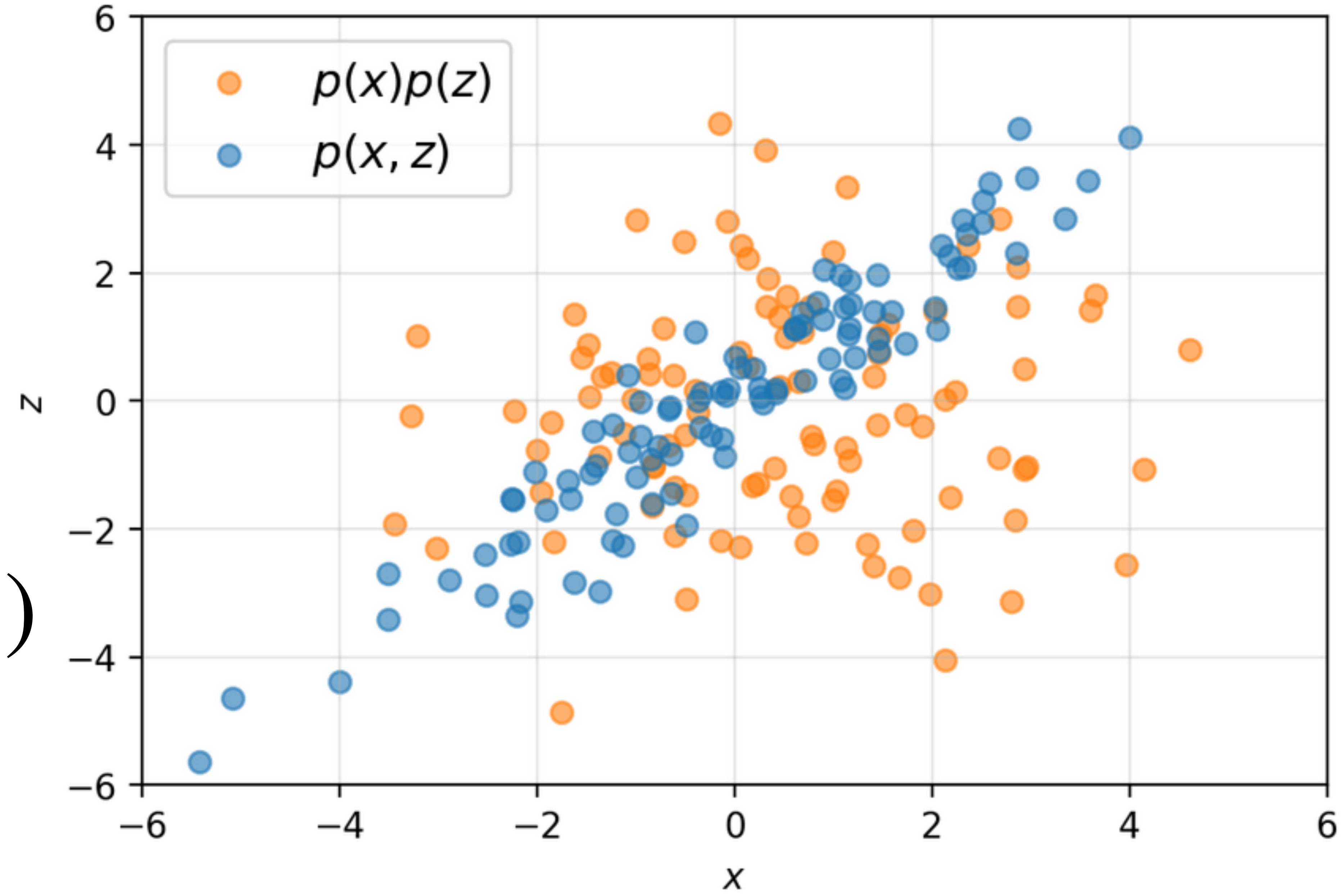
Density Ratio Estimation

- Lets write $\mathbf{a} = (\mathbf{x}, \mathbf{z}); \quad p(\mathbf{a} \mid b = 1) = p(\mathbf{x})p(\mathbf{z}); \quad p(\mathbf{a} \mid b = 0) = p(\mathbf{x}, \mathbf{z})$

$$r(\mathbf{a}) = \frac{p(\mathbf{a} \mid b = 0)}{p(\mathbf{a} \mid b = 1)} = \frac{p(b = 0 \mid \mathbf{a})p(\mathbf{a})p(b = 1)}{p(b = 1 \mid \mathbf{a})p(\mathbf{a})p(b = 0)} = \frac{p(b = 0 \mid \mathbf{a})}{p(b = 1 \mid \mathbf{a})} = \frac{p(b = 0 \mid \mathbf{a})}{1 - p(b = 0 \mid \mathbf{a})}$$

We can learn a parametric function that estimates log density ratios from samples!

$$f_{NN}(\mathbf{a}) = \log r(\mathbf{a}) = \log \frac{p(b = 0 \mid \mathbf{a})}{1 - p(b = 0 \mid \mathbf{a})} = \log(\frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})})$$



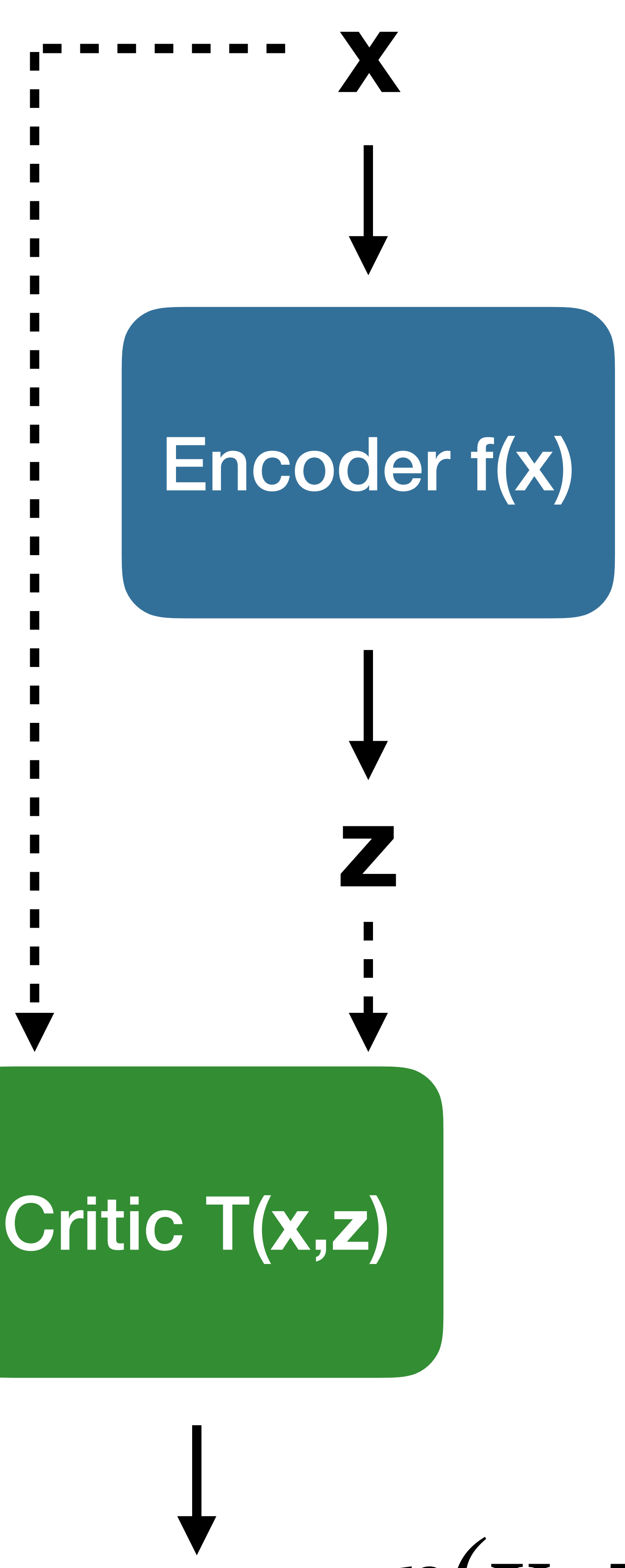
Representation Learning through Maximising I(x, z)

1. Sample from joint distribution:
   $(\mathbf{x}, \mathbf{z}) \sim p(\mathbf{x}, \mathbf{z})$ as $\mathbf{x} \sim p_{\mathscr{D}}(\mathbf{x})$; $\mathbf{z} = f_{\phi}(\mathbf{x})$

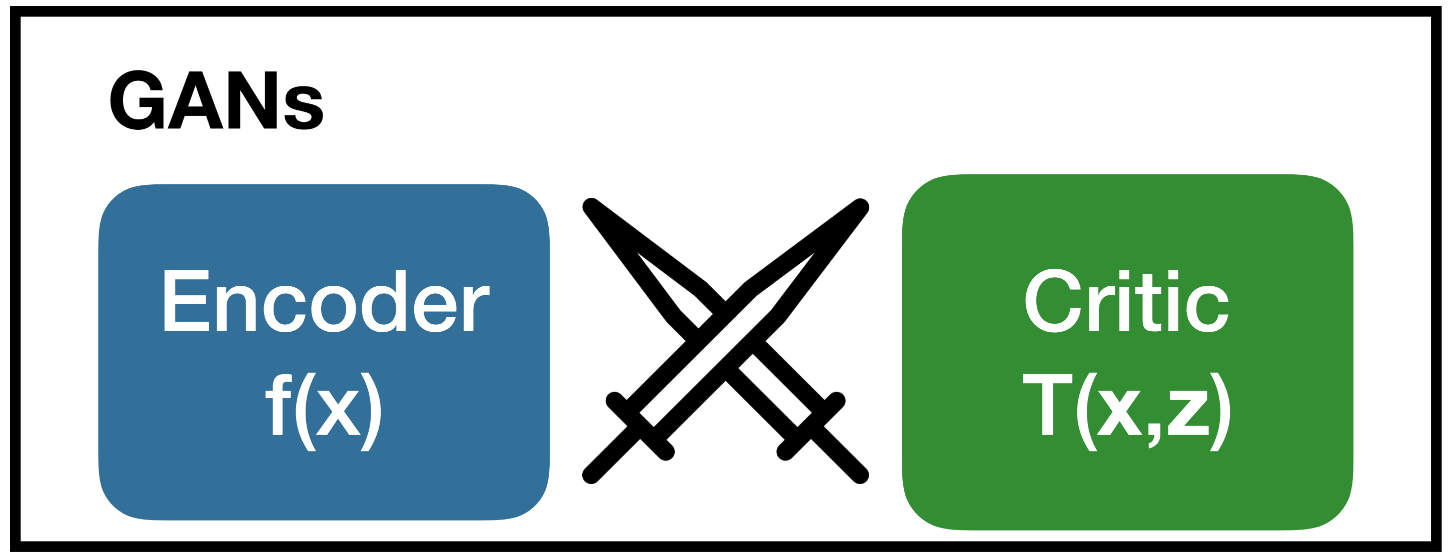
2. <u>Negative sample</u> from factorised distribution:
   $(\mathbf{x}, \mathbf{z}) \sim p(\mathbf{x})p(\mathbf{z})$ as $\mathbf{x}, \mathbf{x}' \sim p_{\mathscr{D}}(\mathbf{x})$; $\mathbf{z} = f_{\phi}(\mathbf{x}')$

3. Estimate $I(\mathbf{x}, \mathbf{z})$ with $T_{\theta}(\mathbf{x}, \mathbf{z})$

4.Optimise $arg\, max_{\theta,\phi}\, I(\mathbf{x}, \mathbf{z})$



$$T_{\theta}(\mathbf{x}, \mathbf{z}) \approx \log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}$$



*

Deep InfoMax

[Hjelm et. al.]

Deep InfoMax (DIM)

- **Max MI between Local and Global Representations**

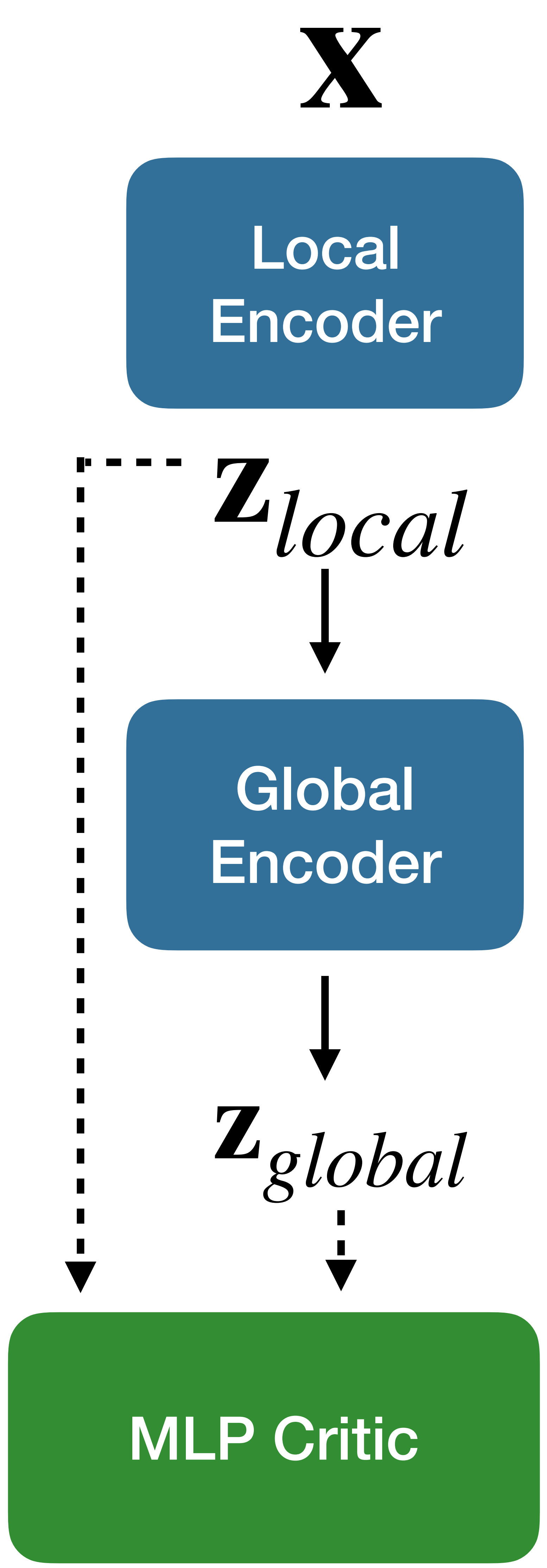
- Critic is an **MLP**:

*

Operation	**Size**	**Activation**
Input → Linear layer	512	ReLU
Linear layer	512	ReLU
Linear layer	1	

Linear SVM Classification Results:

Model	CIFAR10 conv	CIFAR10 fc (1024)	CIFAR10 $Y(64)$	CIFAR100 conv	CIFAR100 fc (1024)	CIFAR100 $Y(64)$
Fully supervised		75.39			42.27	
VAE	60.71	60.54	54.61	37.21	34.05	24.22
AE	62.19	55.78	54.47	31.50	23.89	27.44
β-VAE	62.4	57.89	55.43	32.28	26.89	28.96
AAE	59.44	57.19	52.81	36.22	33.38	23.25
BiGAN	62.57	62.74	52.54	37.59	33.34	21.49
NAT	56.19	51.29	31.16	29.18	24.57	9.72
DIM(G)	52.2	52.84	43.17	27.68	24.35	19.98
DIM(L) (DV)	**72.66**	**70.60**	**64.71**	**48.52**	**44.44**	**39.27**
DIM(L) (JSD)	**73.25**	**73.62**	**66.96**	**48.13**	**45.92**	**39.60**
DIM(L) (infoNCE)	**75.21**	**75.57**	**69.13**	**49.74**	**47.72**	**41.61**

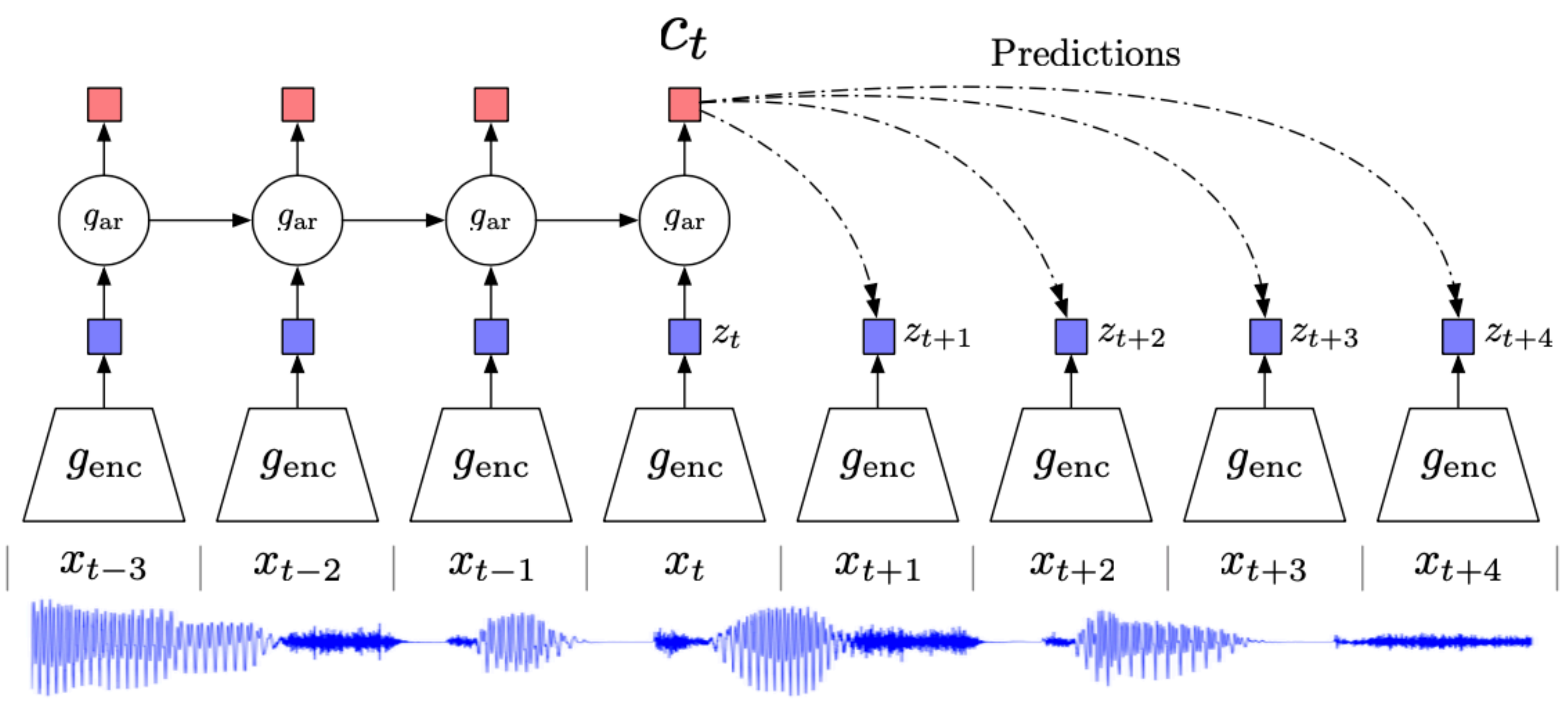


MI Max

[Hjelm et. al.]

Contrastive Predictive Coding

- **Max MI between representations of spatially/temporally similar inputs**
- $\max I(\mathbf{z}^{(\mathbf{t})}, \mathbf{z}^{(\mathbf{t+k})})$ with **(AR NN + Bilinear)** Critic $T(\mathbf{z}^{(\mathbf{1...k})}, \mathbf{z}^{(\mathbf{k+j})}) = g_{AR}(\mathbf{z}^{(1...k)})^{\intercal} W^{(j)} \mathbf{z}^{(k+j)}$



Audio Inputs

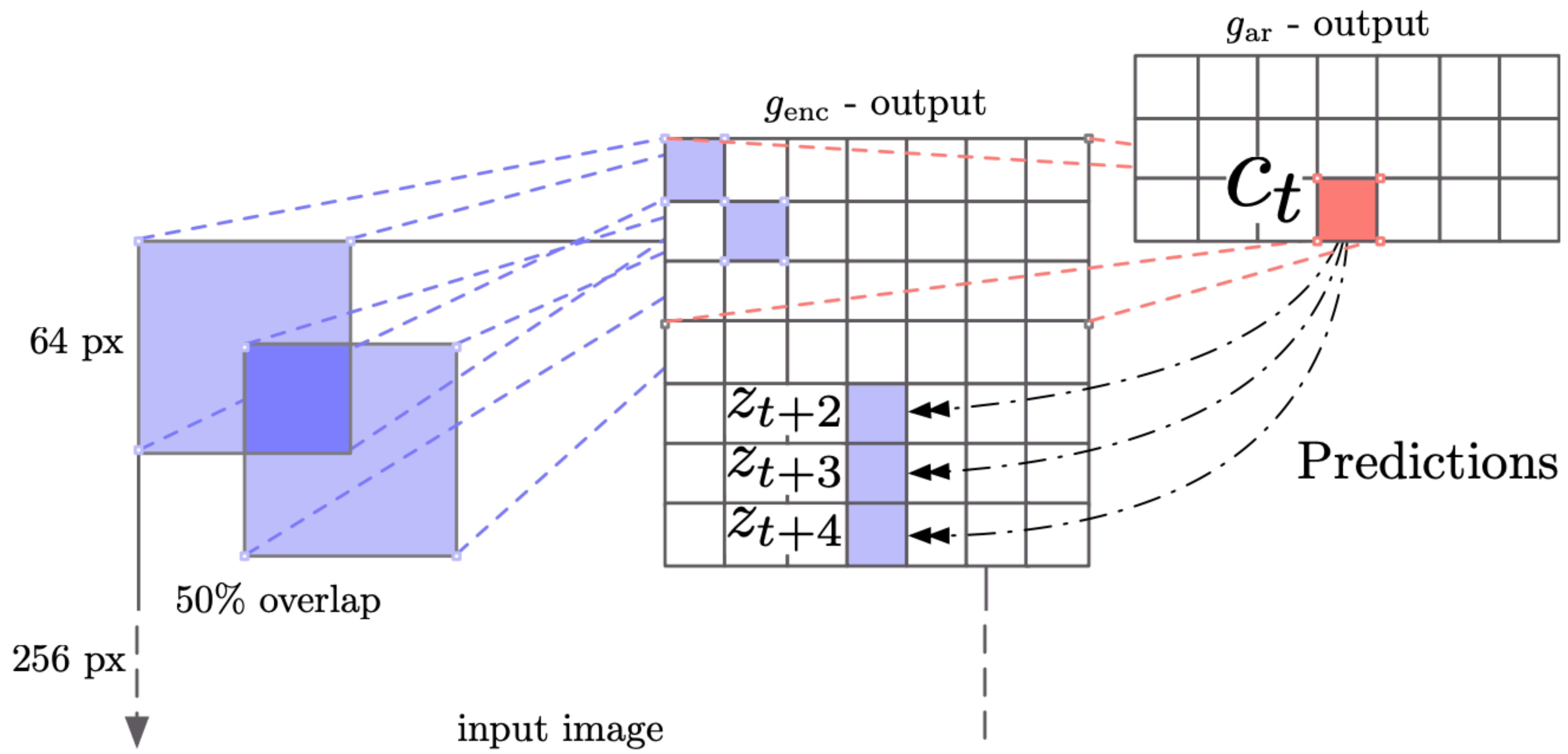


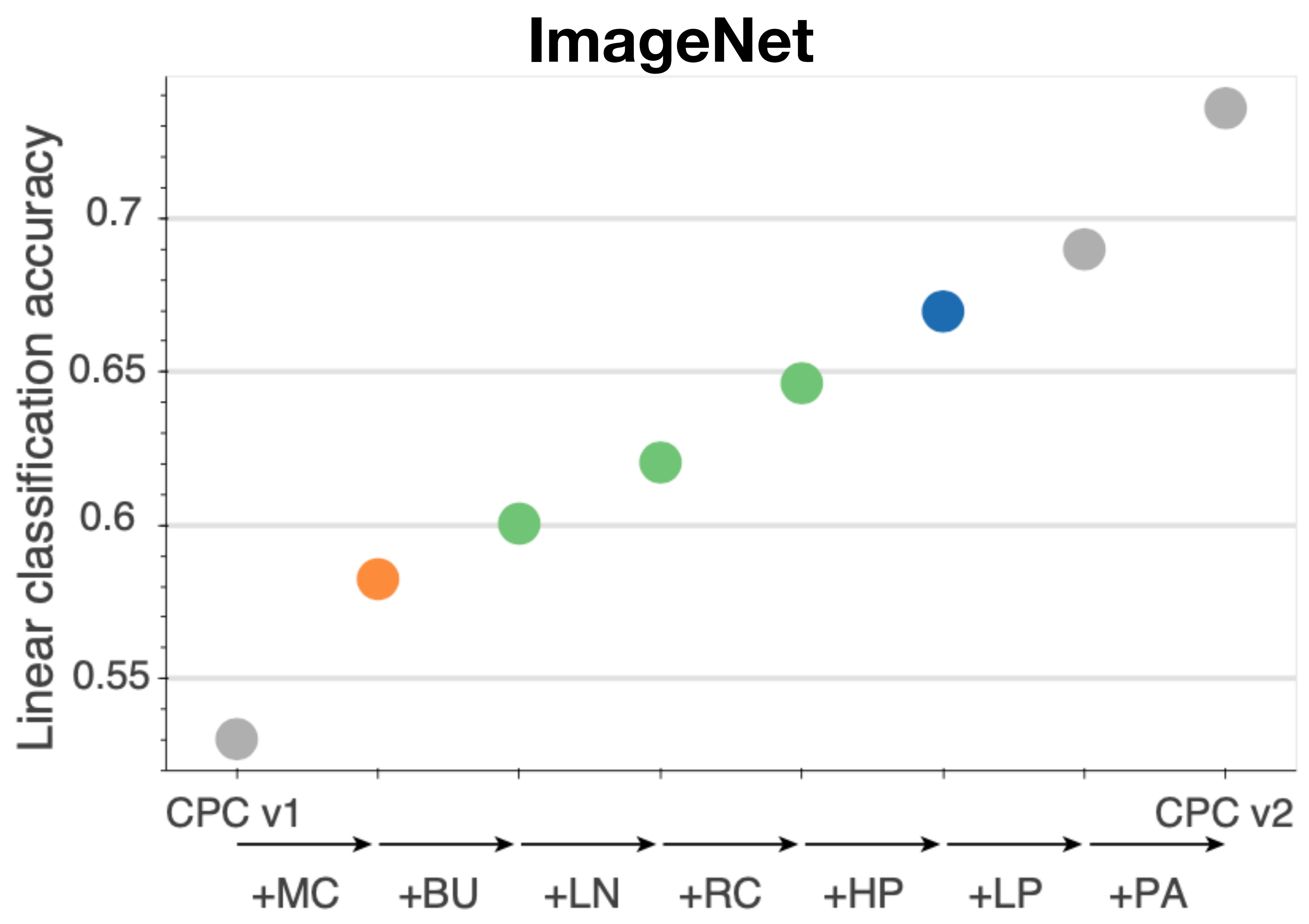
Image Inputs

Audio-Video matching and **Word2Vec** fit within this framework!

[van den Oord et. al.]

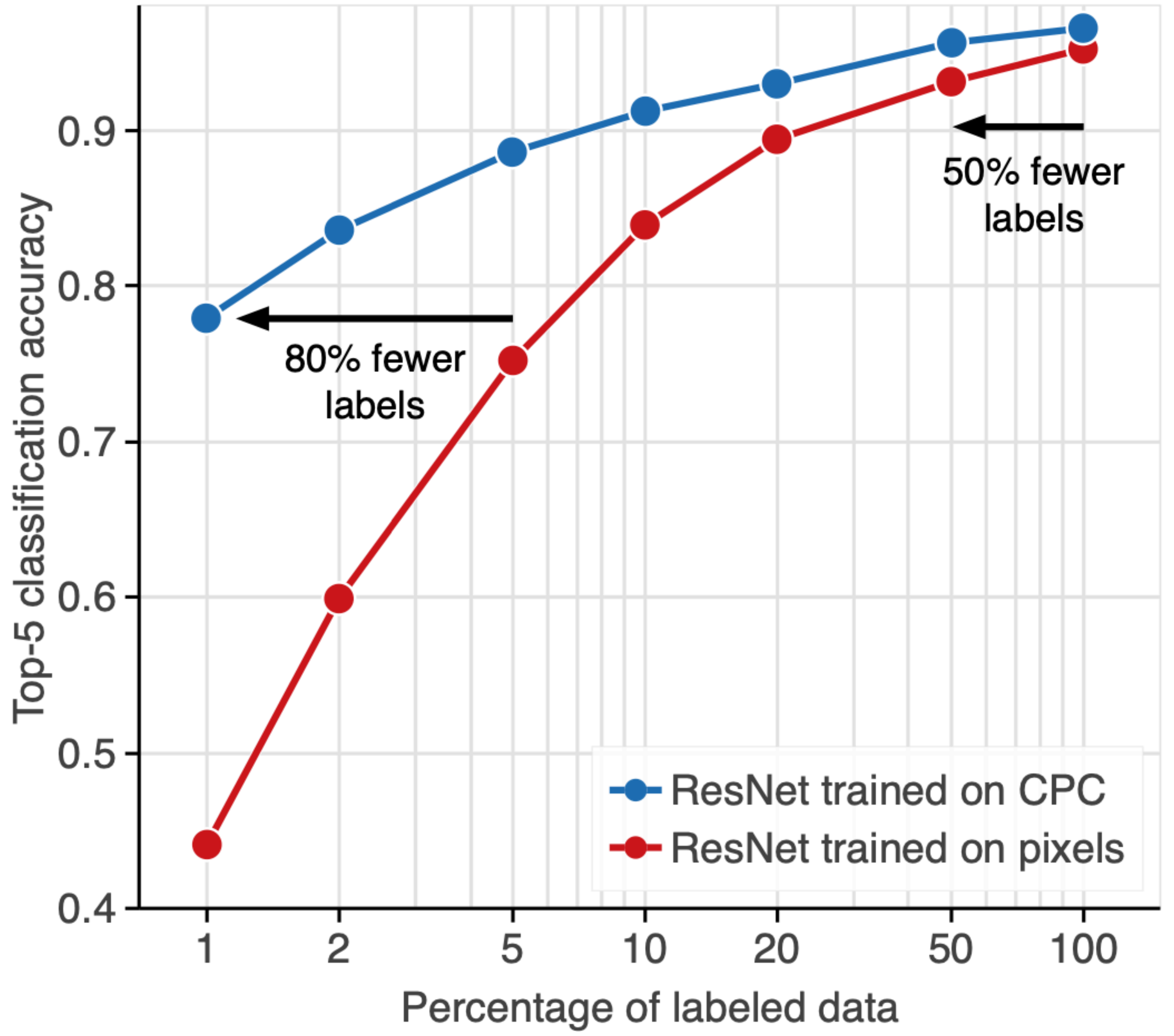
Better Contrastive Predictive Coding

- Large Batch Sizes
- Heavy Patch Augmentation
- Predict in every direction (not just forward)
- Larger Capacity NNs than when using labels
- Layer Norm **(Not Batch Norm)**



Text Document Classification

Method	MR	CR	Subj	MPQA	TREC
Paragraph-vector [40]	74.8	78.1	90.5	74.2	91.8
Skip-thought vector [26]	75.5	79.3	92.1	86.9	91.4
Skip-thought + LN [41]	79.5	82.6	93.4	89.0	-
CPC	76.9	80.1	91.2	87.7	96.8



[He´naff et. al.]

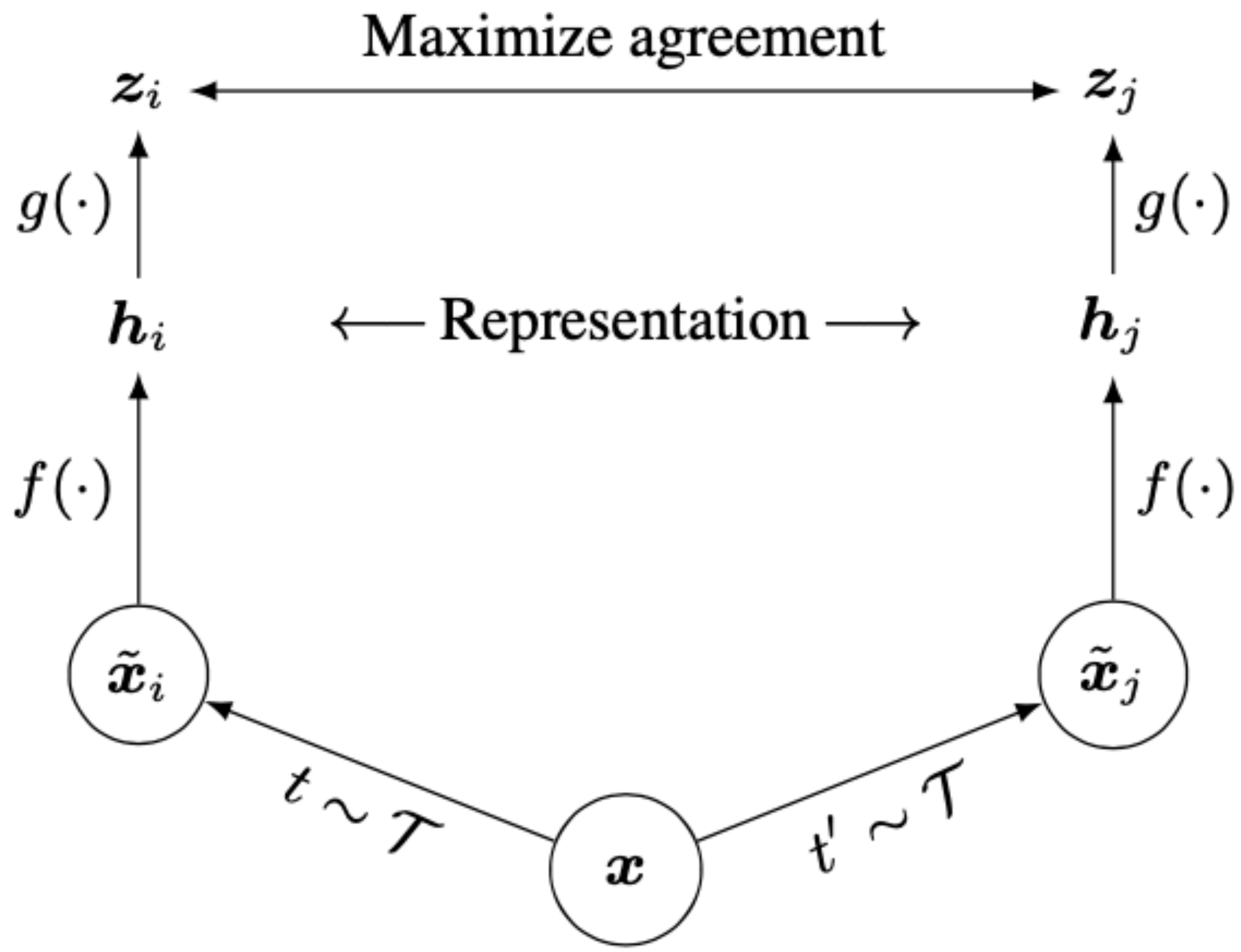
A Unifying Framework for Contrastive Learning

[Chen et. al.]

A Unifying Framework for Contrastive Learning

- Use ~InfoNCE to **max MI between augmented inputs**

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$



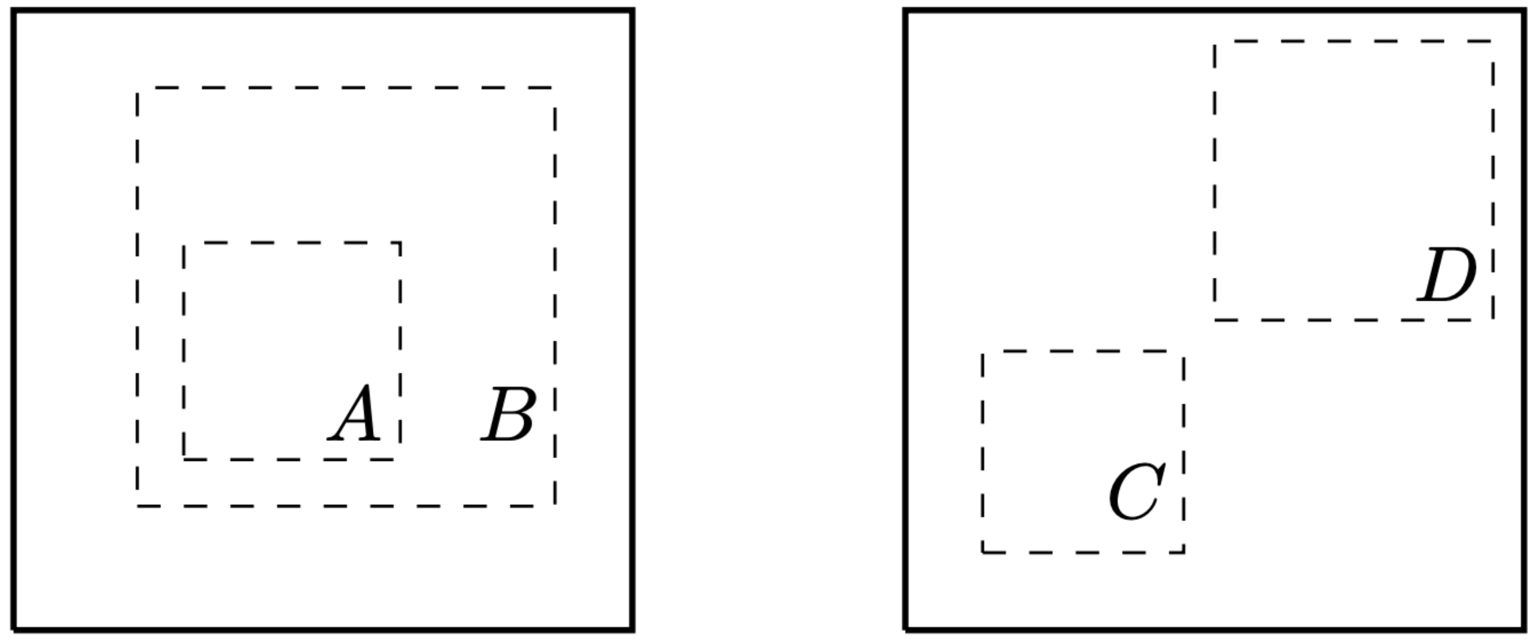
[Chen et. al.]

A Unifying Framework for Contrastive Learning

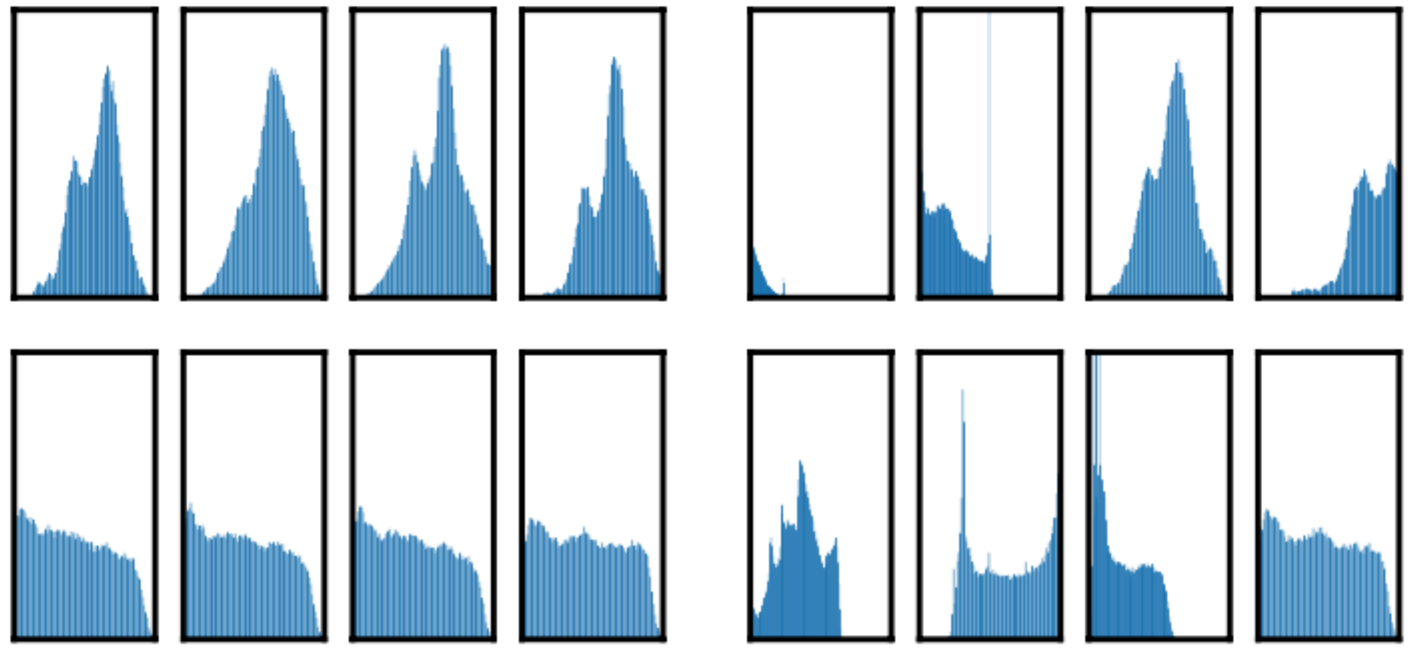
- Use ~InfoNCE to **max MI between augmented inputs**

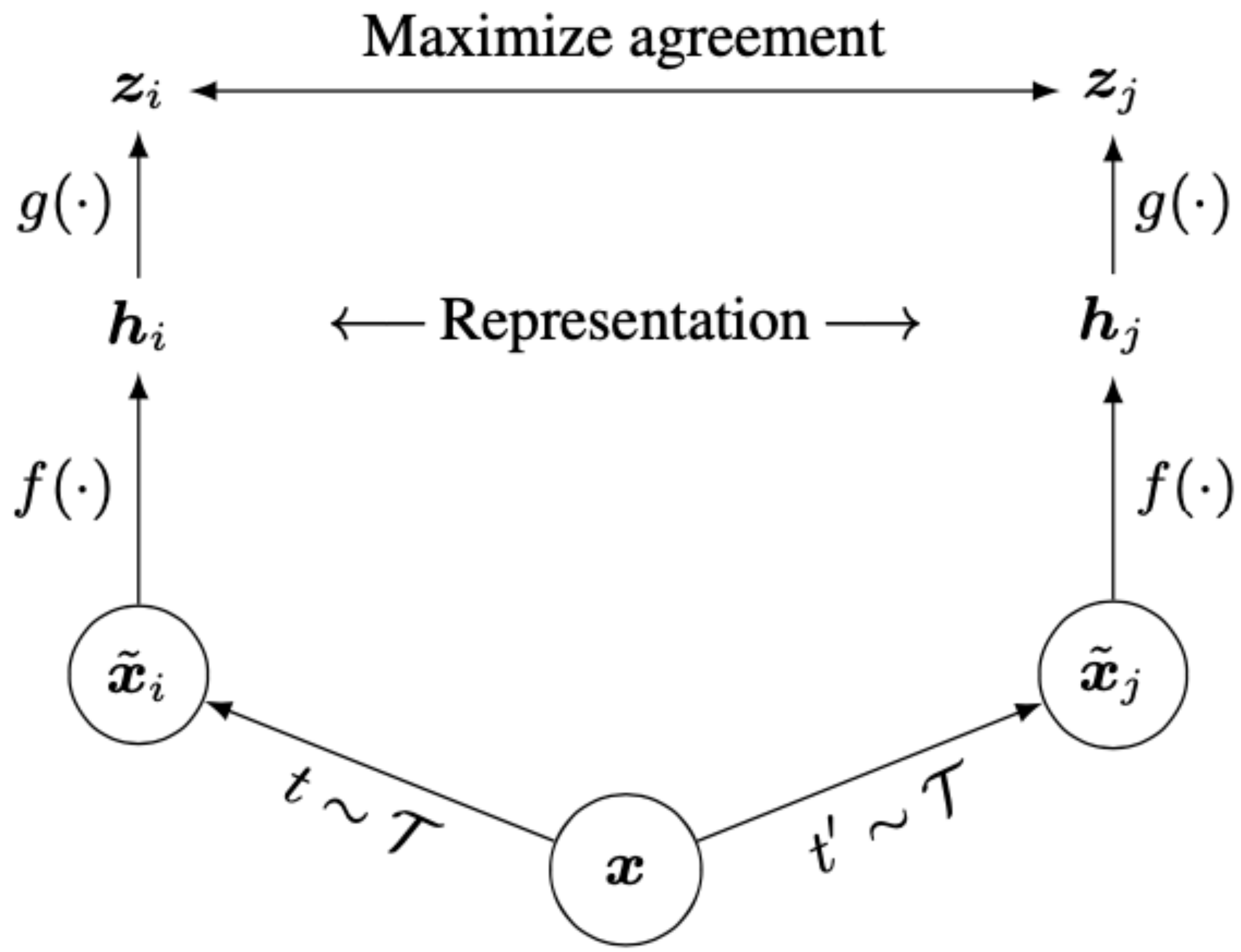
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

- Random Crop + Color Distortion Augmentation



(a) Global and local views. (b) Adjacent views.


(a) Without color distortion. (b) With color distortion.

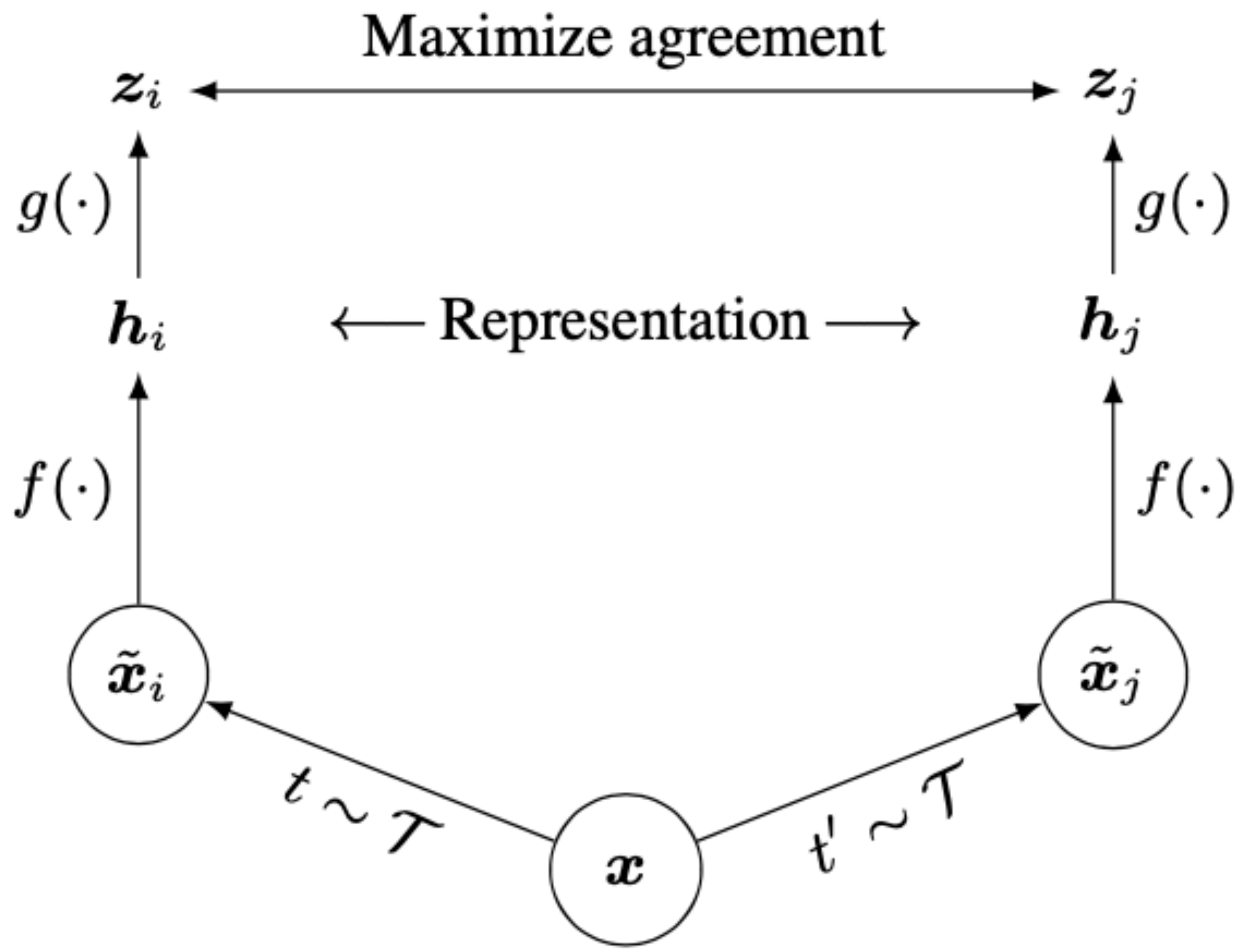


[Chen et. al.]

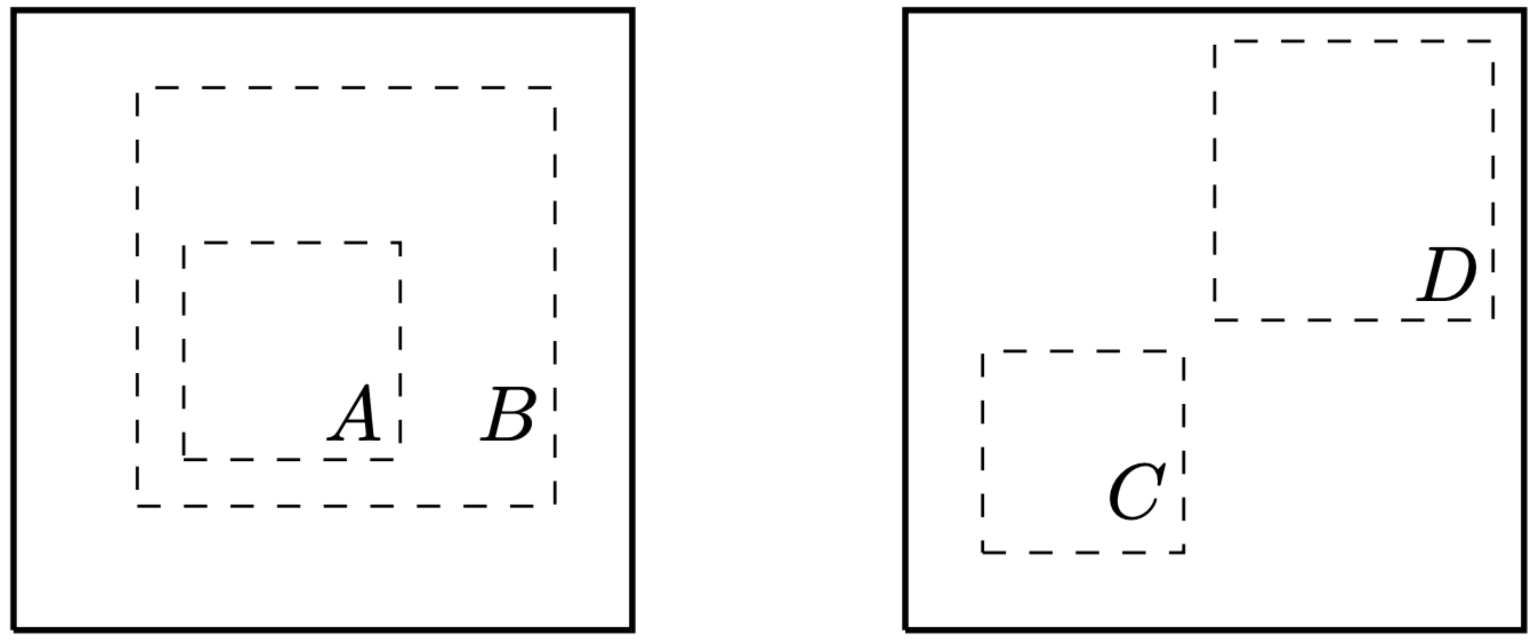
A Unifying Framework for Contrastive Learning

- Use ~InfoNCE to **max MI between augmented inputs**

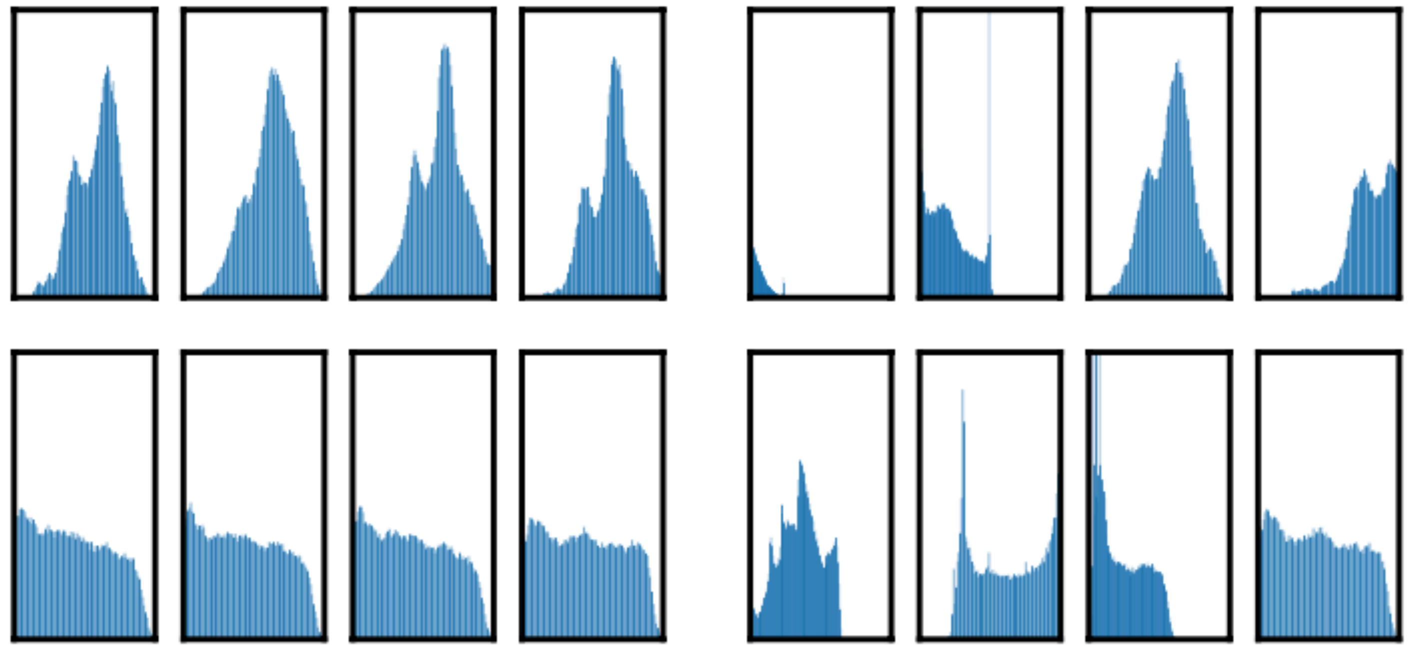
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$



- Random Crop + Color Distortion Augmentation



(a) Global and local views. (b) Adjacent views.



(a) Without color distortion. (b) With color distortion.

- Big Batches, Big Computers

[1]With 128 TPU v3 cores, it takes ~1.5 hours to train our ResNet-50 with a batch size of 4096 for 100 epochs.

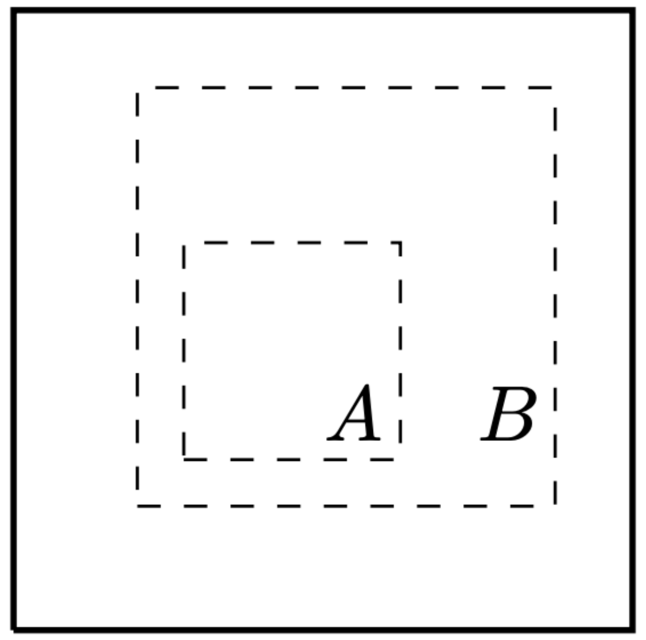
[Chen et. al.]

A Unifying Framework for Contrastive Learning

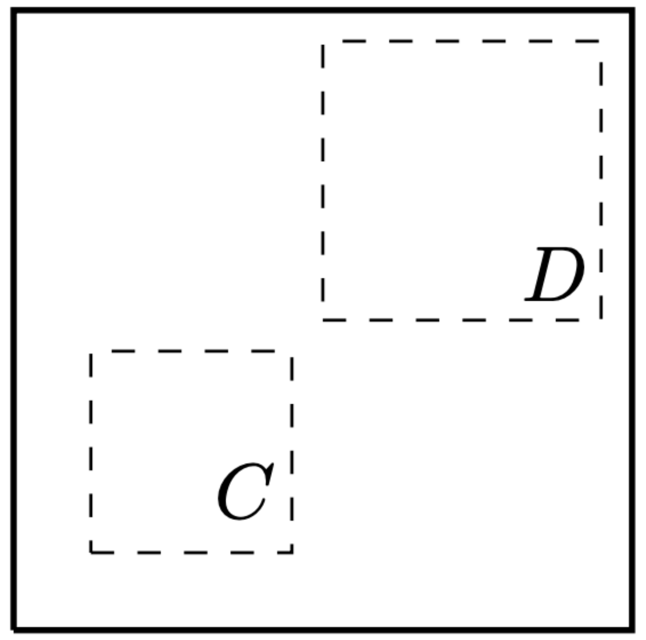
- Use ~InfoNCE to **max MI between augmented inputs**

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

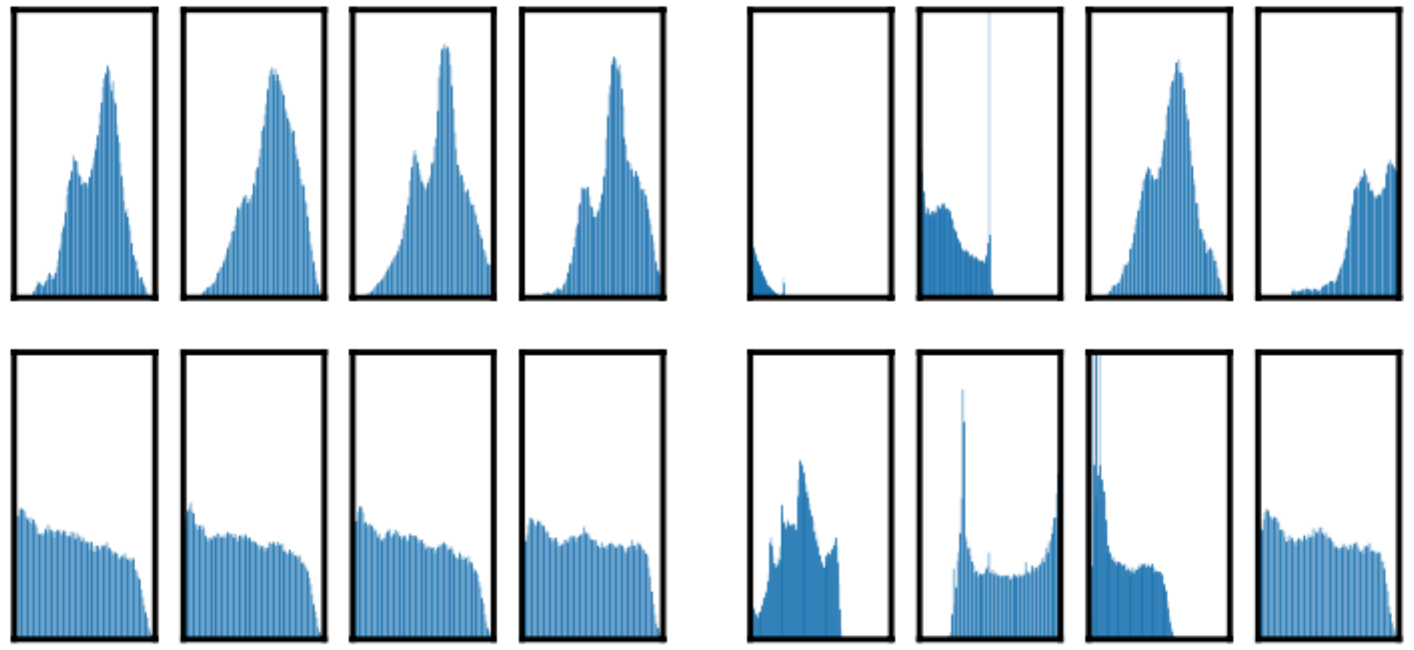
- Random Crop + Color Distortion Augmentation

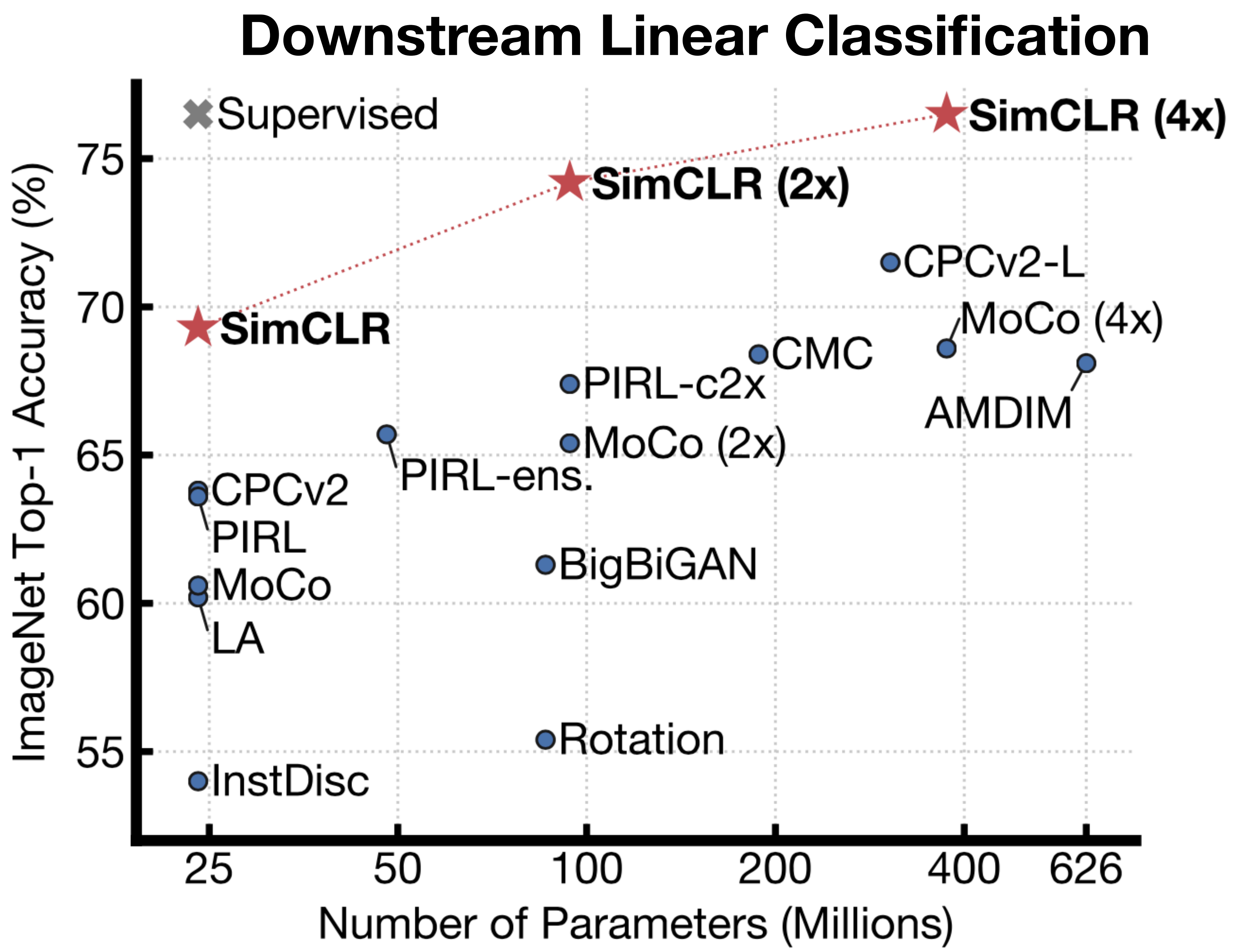


(a) Global and local views.



(b) Adjacent views.


(a) Without color distortion. (b) With color distortion.



- Big Batches, Big Computers

[1] With 128 TPU v3 cores, it takes ∼1.5 hours to train our ResNet-50 with a batch size of 4096 for 100 epochs.

[Chen et. al.]

Mutual Information Neural Estimation (Deep InfoMax)

Mutual Information Neural Estimation (Deep InfoMax)

$$I(\mathbf{x},\mathbf{z}) = E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x}\,|\,\mathbf{z})q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})q(\mathbf{x}\,|\,\mathbf{z})}\right] = \underbrace{E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})}\right]} + E_{p(\mathbf{z})}[KL(p(\mathbf{x}\,|\,\mathbf{z})\,||\,q(\mathbf{x}\,|\,\mathbf{z})]$$

Mutual Information Neural Estimation (Deep InfoMax)

$$I(\mathbf{x},\mathbf{z}) = E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x}\,|\,\mathbf{z})q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})q(\mathbf{x}\,|\,\mathbf{z})}\right] = \underbrace{E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})}\right]} + E_{p(\mathbf{z})}[KL(p(\mathbf{x}\,|\,\mathbf{z})\,||\,q(\mathbf{x}\,|\,\mathbf{z})]$$

$$\downarrow$$

$$I(\mathbf{x},\mathbf{z}) \geq H(x) + E_{p(\mathbf{x},\mathbf{z})}[\log q(\mathbf{x}\,|\,\mathbf{z})]$$

Mutual Information Neural Estimation (Deep InfoMax)

$$I(\mathbf{x},\mathbf{z}) = E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x}\,|\,\mathbf{z})q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})q(\mathbf{x}\,|\,\mathbf{z})}\right] = \underbrace{E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})}\right]} + E_{p(\mathbf{z})}[KL(p(\mathbf{x}\,|\,\mathbf{z})\,||\,q(\mathbf{x}\,|\,\mathbf{z})]$$

$$\downarrow$$

$$I(\mathbf{x},\mathbf{z}) \geq H(x) + E_{p(\mathbf{x},\mathbf{z})}[\log q(\mathbf{x}\,|\,\mathbf{z})]$$

We choose $q(\mathbf{x}\,|\,\mathbf{z}) = \dfrac{p(\mathbf{x})}{E_{p(\mathbf{x})}[e^{T(\mathbf{x},\mathbf{z})}]}e^{T(\mathbf{x},\mathbf{z})}$

Mutual Information Neural Estimation (Deep InfoMax)

$$I(\mathbf{x},\mathbf{z}) = E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x}\,|\,\mathbf{z})q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})q(\mathbf{x}\,|\,\mathbf{z})}\right] = \underbrace{E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})}\right]} + E_{p(\mathbf{z})}[KL(p(\mathbf{x}\,|\,\mathbf{z})\,||\,q(\mathbf{x}\,|\,\mathbf{z})]$$

$$\downarrow$$

$$I(\mathbf{x},\mathbf{z}) \geq H(x) + E_{p(\mathbf{x},\mathbf{z})}[\log q(\mathbf{x}\,|\,\mathbf{z})]$$

We choose $q(\mathbf{x}\,|\,\mathbf{z}) = \dfrac{p(\mathbf{x})}{E_{p(\mathbf{x})}[e^{T(\mathbf{x},\mathbf{z})}]}e^{T(\mathbf{x},\mathbf{z})}$

$$I(\mathbf{x},\mathbf{z}) \geq \max_{\theta} E_{p(\mathbf{x},\mathbf{z})}[T_\theta] - \log(E_{p(\mathbf{x})p(\mathbf{z})}[e^{T_\theta}])$$

Mutual Information Neural Estimation (Deep InfoMax)

$$I(\mathbf{x}, \mathbf{z}) = E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{p(\mathbf{x}\,|\,\mathbf{z})q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})q(\mathbf{x}\,|\,\mathbf{z})}\right] = \underbrace{E_{p(\mathbf{x},\mathbf{z})}\left[\log \frac{q(\mathbf{x}\,|\,\mathbf{z})}{p(\mathbf{x})}\right]} + E_{p(\mathbf{z})}[KL(p(\mathbf{x}\,|\,\mathbf{z})\,||\,q(\mathbf{x}\,|\,\mathbf{z})]$$

$$\downarrow$$

$$I(\mathbf{x}, \mathbf{z}) \geq H(x) + E_{p(\mathbf{x},\mathbf{z})}[\log q(\mathbf{x}\,|\,\mathbf{z})]$$

We choose $q(\mathbf{x}\,|\,\mathbf{z}) = \dfrac{p(\mathbf{x})}{E_{p(\mathbf{x})}[e^{T(\mathbf{x},\mathbf{z})}]}e^{T(\mathbf{x},\mathbf{z})}$

$$I(\mathbf{x}, \mathbf{z}) \geq \max_{\theta} E_{p(\mathbf{x},\mathbf{z})}[T_\theta] - \log(E_{p(\mathbf{x})p(\mathbf{z})}[e^{T_\theta}])$$

Bound tight if $T = \log \dfrac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})}$

InfoNCE (Contrastive Predictive Coding)

[van den Oord et. al.]

InfoNCE (Contrastive Predictive Coding)

Similar to MINE, but:
We draw an additional K-1 samples from x to a normalisation constant:

$$E_{p(\mathbf{x})}[e^T] \approx m(\mathbf{z} \,|\, \mathbf{x}_1 \ldots \mathbf{x}_k) = \frac{1}{K} \sum_{i=1}^{K} e^{T(\mathbf{x}_i, \mathbf{z})}$$

[van den Oord et. al.]

InfoNCE (Contrastive Predictive Coding)

Similar to MINE, but:
We draw an additional K-1 samples from x to a normalisation constant:

$$E_{p(\mathbf{x})}[e^T] \approx m(\mathbf{z}\,|\,\mathbf{x}_1 \ldots \mathbf{x}_k) = \frac{1}{K}\sum_{i=1}^{K} e^{T(\mathbf{x}_i,\mathbf{z})}$$

softmax cross-entropy

$$I(\mathbf{x},\mathbf{z}) \geq \max_{\theta} E_{p(\mathbf{z}|\mathbf{x}_1)p(\mathbf{x}_1 \ldots \mathbf{x}_K)}\left[\log \frac{e^{T_\theta(\mathbf{x}_1,\mathbf{z})}}{\sum_{j=1}^{K} e^{T_\theta(\mathbf{x}_j,\mathbf{z})}}\right] := I^{NCE}$$

[van den Oord et. al.]

InfoNCE (Contrastive Predictive Coding)

Similar to MINE, but:
We draw an additional K-1 samples from x to a normalisation constant:

$$E_{p(\mathbf{x})}[e^T] \approx m(\mathbf{z} \,|\, \mathbf{x}_1 \ldots \mathbf{x}_k) = \frac{1}{K}\sum_{i=1}^{K} e^{T(\mathbf{x}_i, \mathbf{z})}$$

softmax cross-entropy

$$I(\mathbf{x}, \mathbf{z}) \geq \max_{\theta} E_{p(\mathbf{z}|\mathbf{x}_1)p(\mathbf{x}_1 \ldots \mathbf{x}_K)}\left[\log \frac{e^{T_\theta(\mathbf{x}_1, \mathbf{z})}}{\sum_{j=1}^{K} e^{T_\theta(\mathbf{x}_j, \mathbf{z})}}\right] := I^{NCE}$$

Note that: $I(\mathbf{x}, \mathbf{z}) \geq I^{NCE} + \log K$

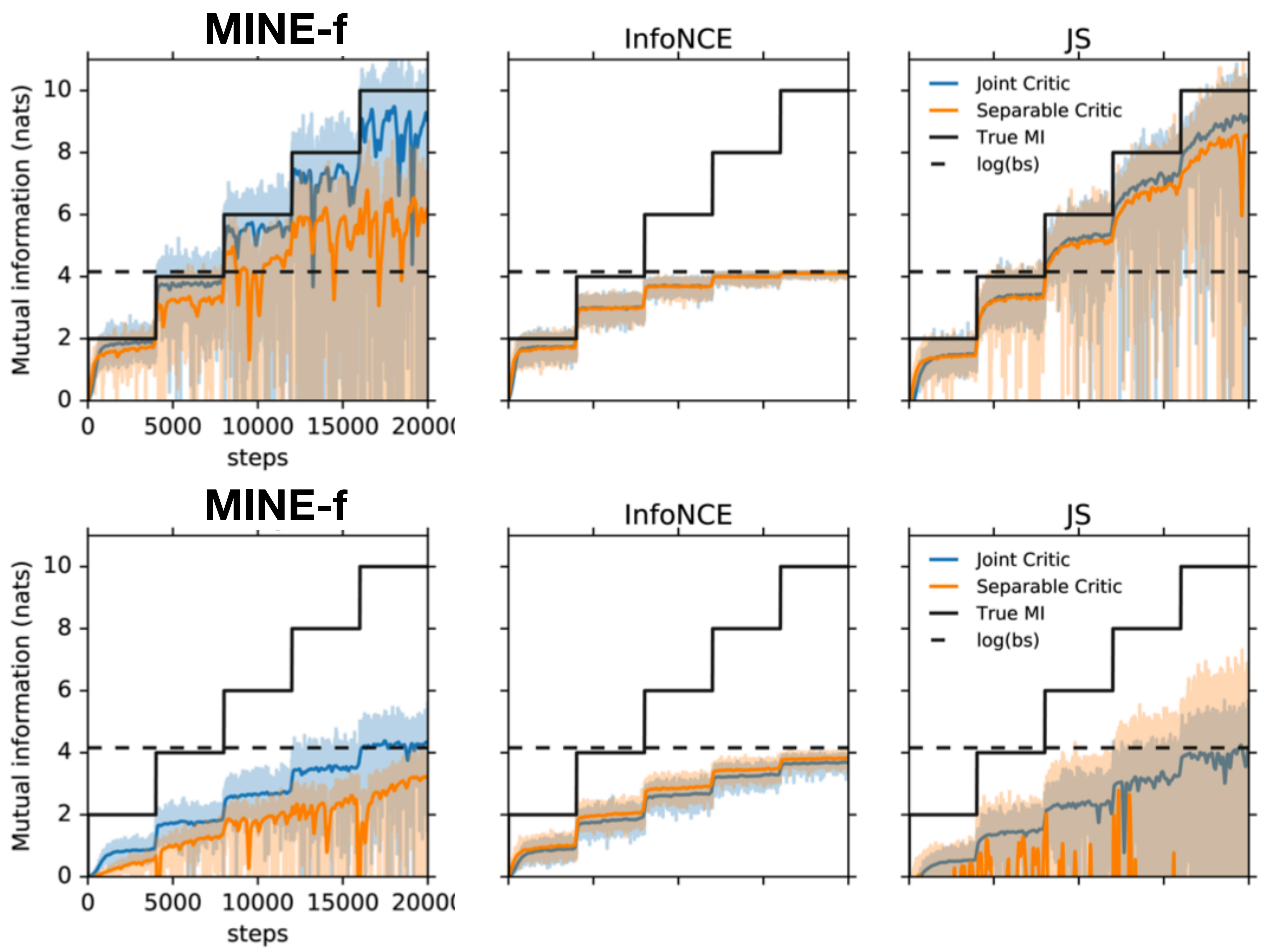
Larger K (batch sizes) will yield less biased estimates

[van den Oord et. al.]

Comparing MI estimators

Correlated Gaussian

$$\mathbf{z} = (W\mathbf{x})^3; \quad \mathbf{x} \sim \mathcal{N}(0,1)$$



log(64)

[Poole et. al.]

Which One Should I Use?

Batch-Size: N

2N Critic Evaluations

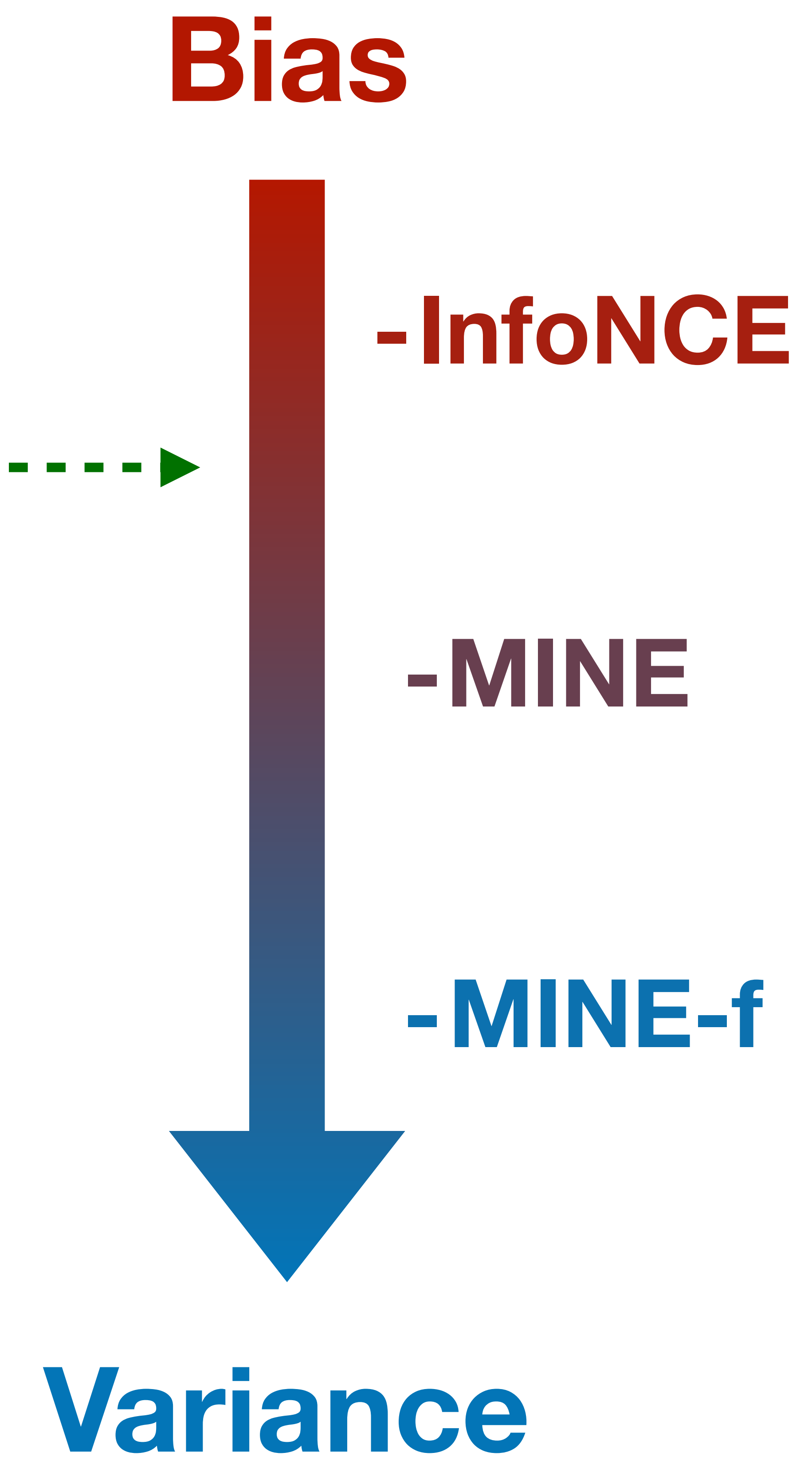
- JSD
- MINE
- MINE-f

N^2 Critic Evaluations

- InfoNCE

Not an estimator

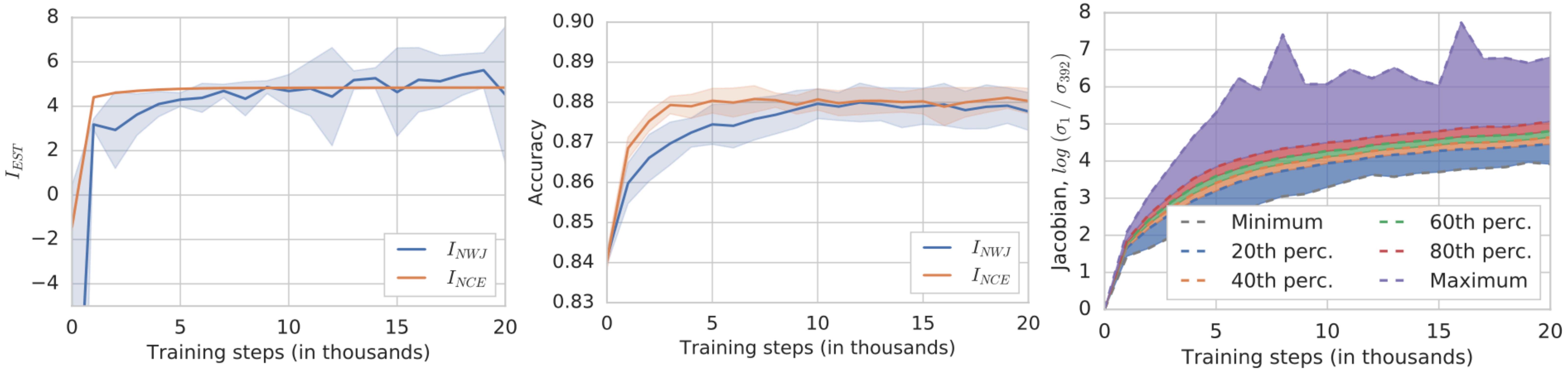
-JSD



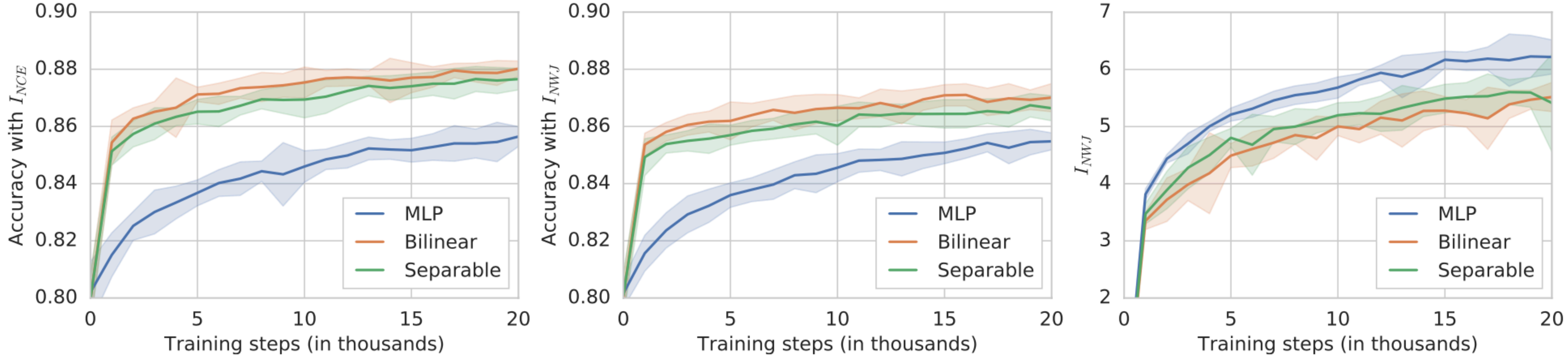
*** If possible use InfoNCE with large batch size**

Is MI Maximisation Really the Solution?

- During training encoders become less invertible



- More flexible critics can make for worse representations



Bilinear: $T(\mathbf{z}, \mathbf{x}) = \mathbf{z}^\intercal W \mathbf{x}$

Separable: $T(\mathbf{z}, \mathbf{x}) = \phi(\mathbf{z}^\intercal)\phi(\mathbf{x})$

[Tschannen et. al.]

Is MI Maximisation Really the Solution?

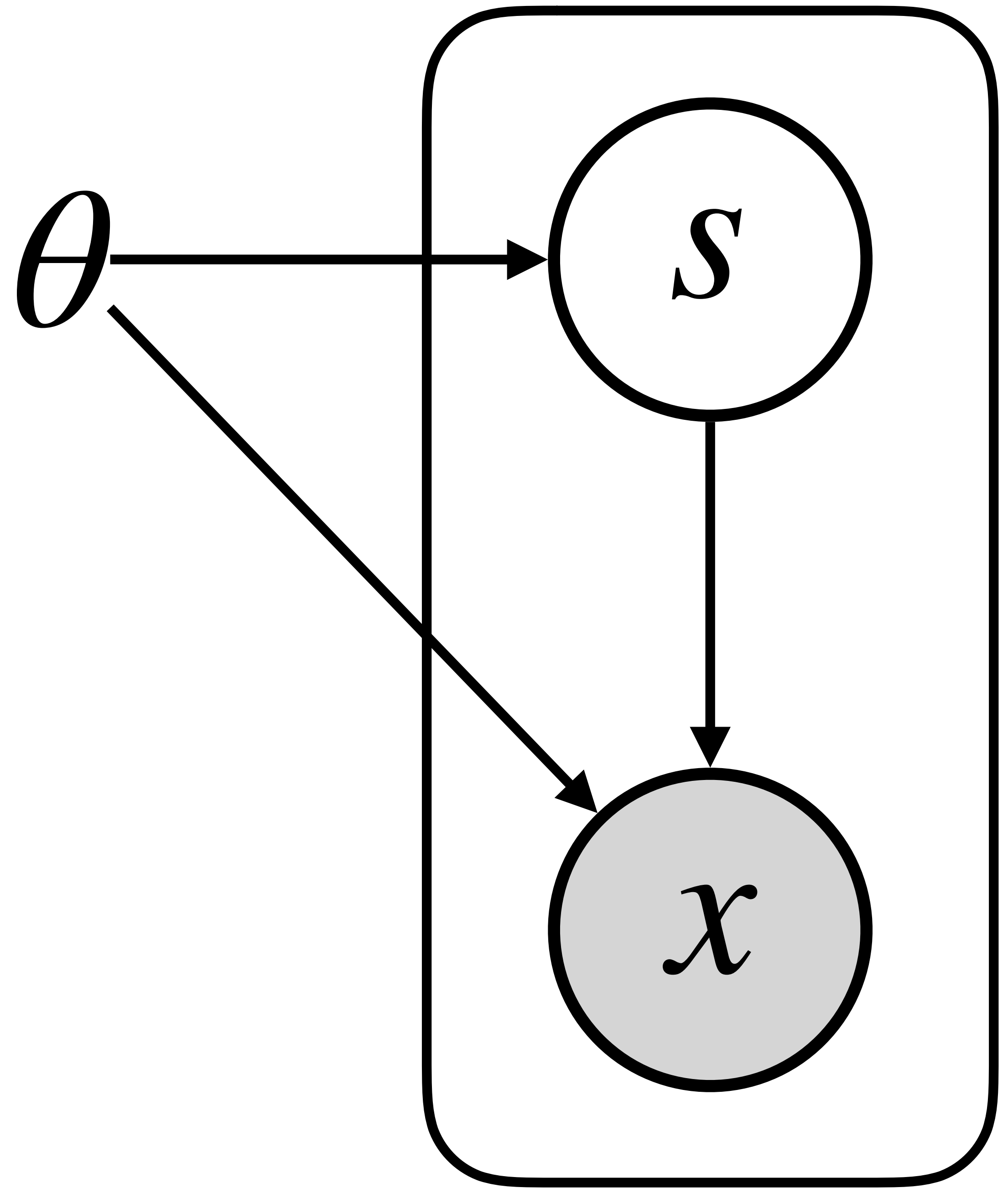
- A bijection can be applied to a useful representation making it not useful, maintaining $I(\mathbf{x}, \mathbf{z})$
- Utility of representation is a function of "**decodable information**", not MI
- This depends on **inductive biases** from:
  - MI estimator
  - Critic Function + Encoder Function
  - Objects between which MI is maximised

References

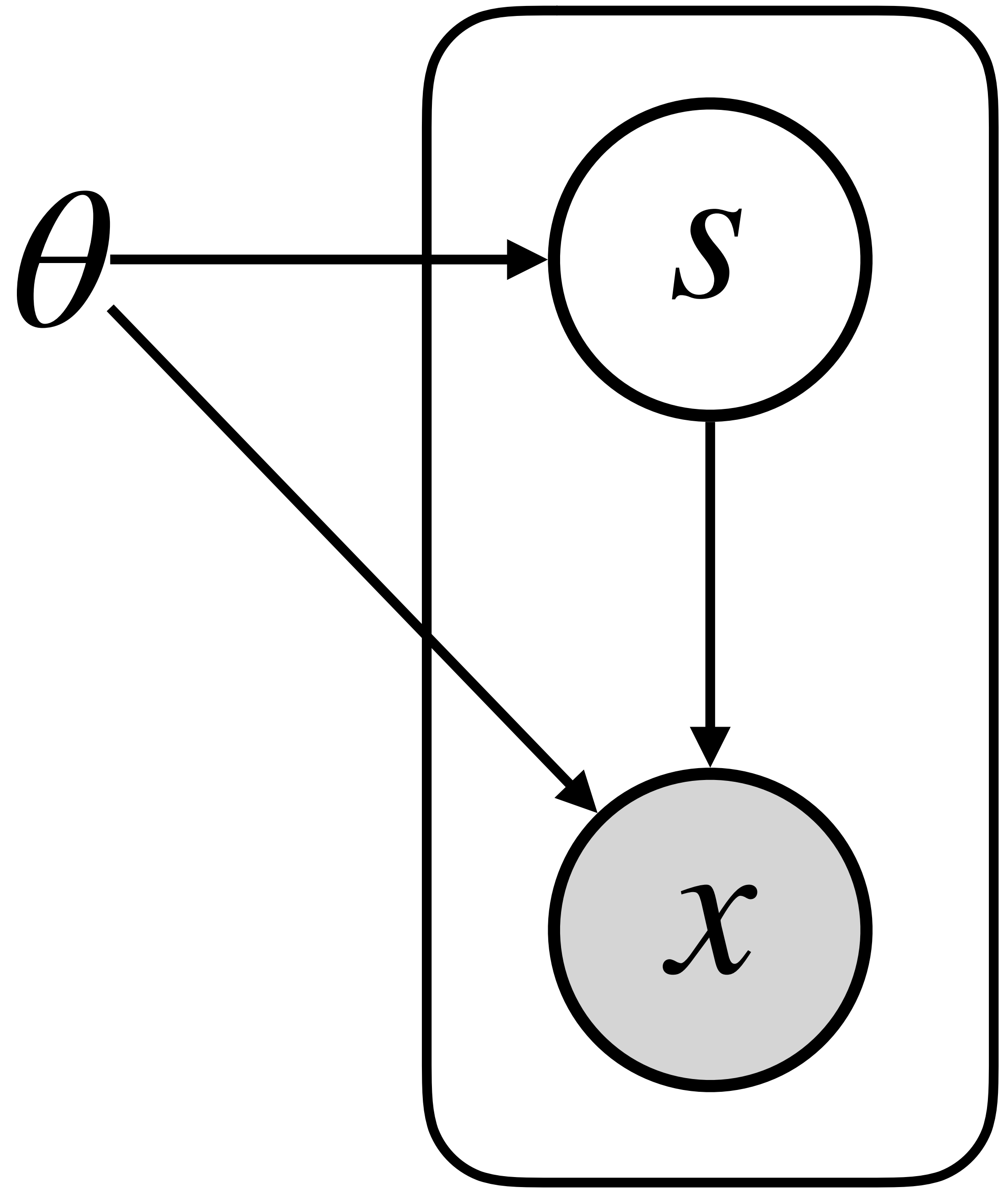
- [Larsson et. al.] Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles https://arxiv.org/pdf/1603.09246.pdf
- [Gidaris et. al.] Unsupervised Representation Learning by Predicting Image Rotations https://openreview.net/forum?id=S1v4N2l0-
- [Arandjelovic et. al.] Look, Listen and Learn https://arxiv.org/abs/1705.08168
- [Mikolov et. al.] Distributed Representations of Words and Phrases and their Compositionality https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf
- [Hjelm et. al.] Learning deep representations by mutual information estimation and maximization https://arxiv.org/abs/1808.06670
- [Belghazi et. al.] MINE: Mutual Information Neural Estimation https://arxiv.org/abs/1801.04062
- [van den Oord et. al.] Representation Learning with Contrastive Predictive Coding https://arxiv.org/abs/1807.03748
- [Henáff et. al.] Data-Efficient Image Recognition with Contrastive Predictive Coding https://arxiv.org/abs/1905.09272
- [Chen et. al.] A Simple Framework for Contrastive Learning of Visual Representations https://arxiv.org/abs/2002.05709
- [Poole et. al.] On Variational Bounds of Mutual Information https://arxiv.org/abs/1905.06922
- [Tschannen et. al.] On Mutual Information Maximization for Representation Learning https://arxiv.org/abs/1907.13625

A Generative Modelling Perspective

Latent Variable Generative Models



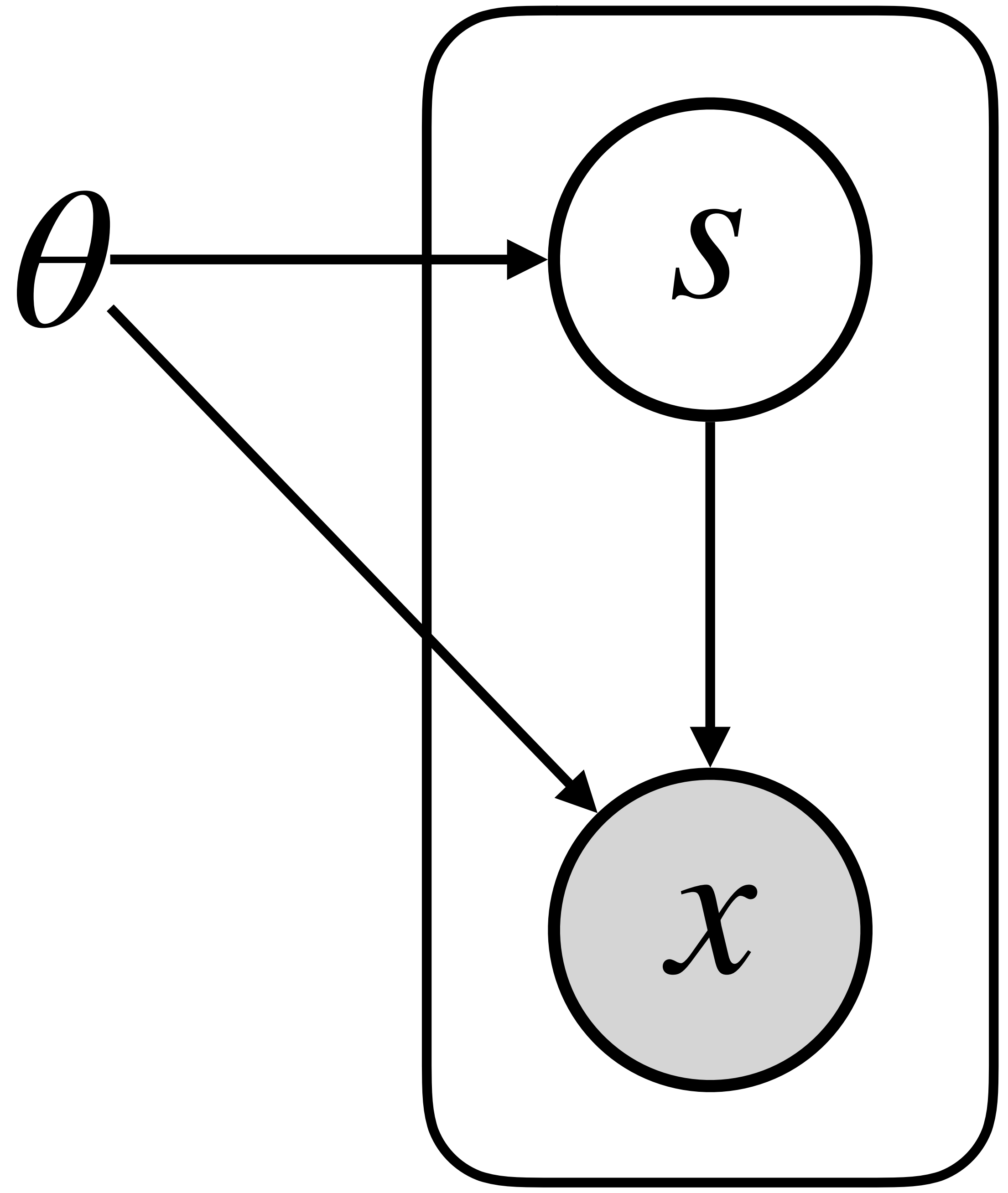
Latent Variable Generative Models



Goal: $\theta^* = \underset{\theta \in \Theta}{\textbf{arg max}} \log p_\theta(D)$

Latent Variable Generative Models

Goal: $\theta^* = \mathop{\textbf{arg max}}_{\theta \in \Theta} \log p_\theta(D)$

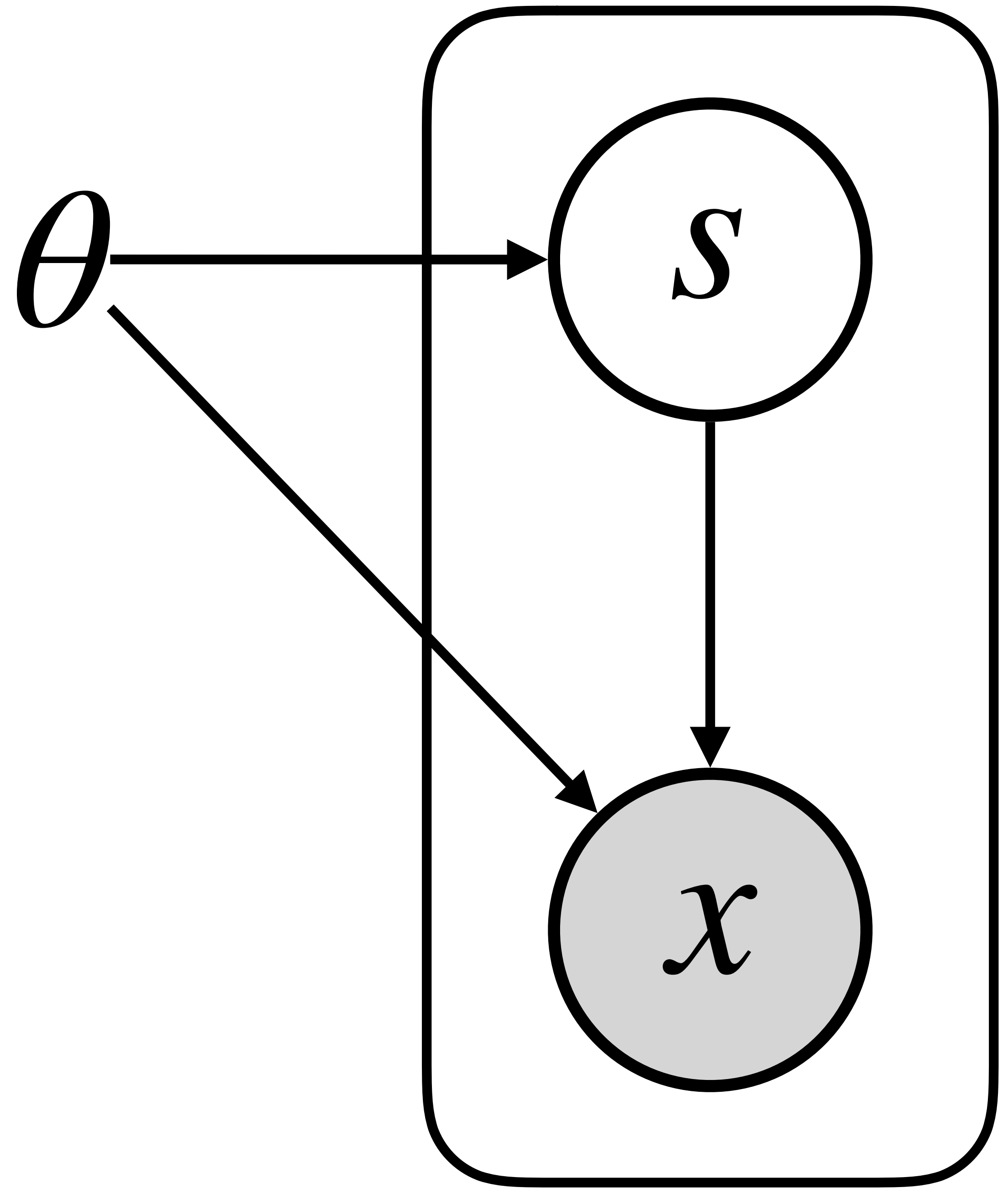


Probabilistic PCA

$$p_\theta(s) = \mathcal{N}(s; 0, I)$$

$$p_\theta(x \mid s) = \mathcal{N}\left(x; Ws + \mu, \sigma^2 I\right)$$

Latent Variable Generative Models



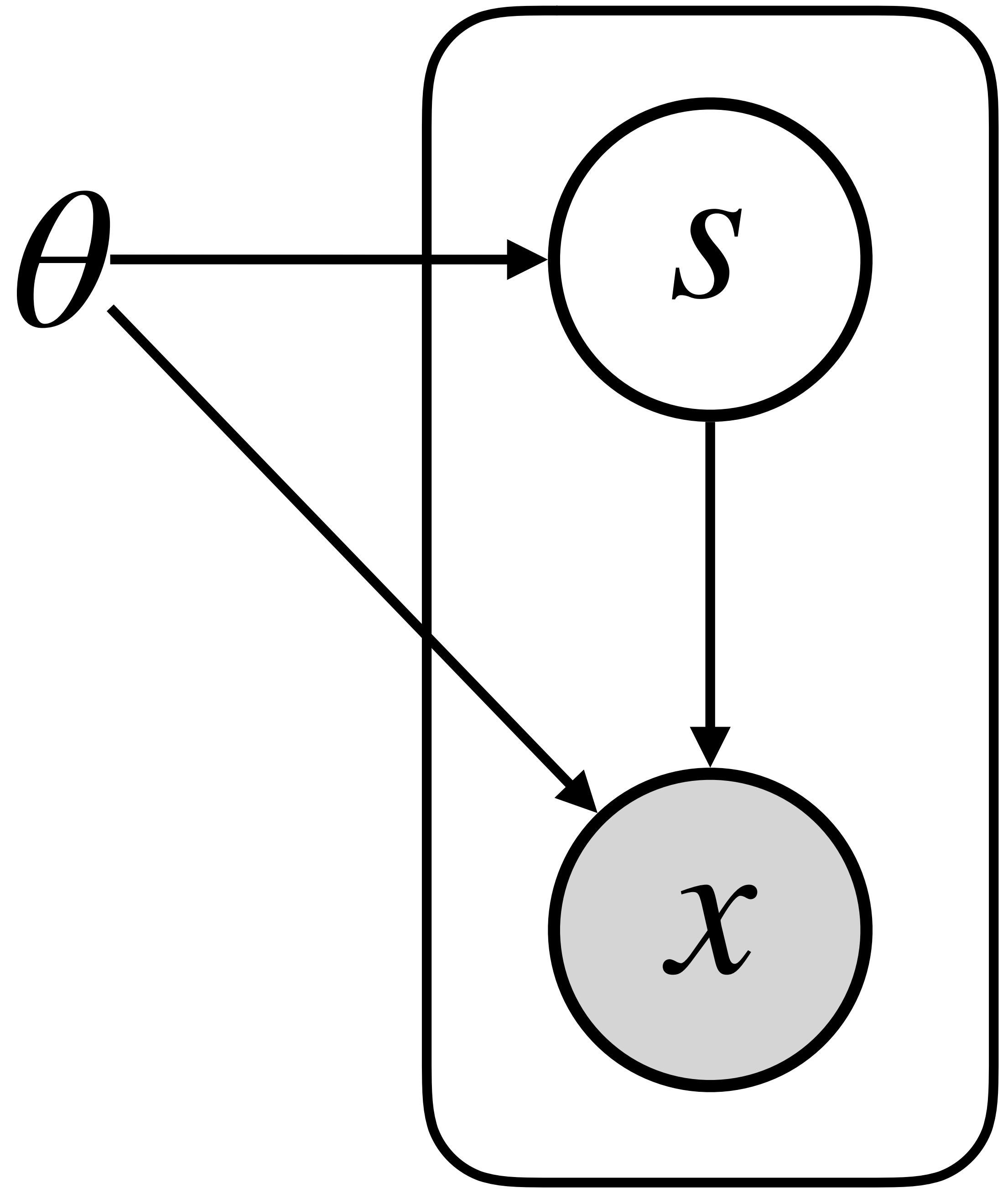
Goal: $\theta^* = \underset{\theta \in \Theta}{\textbf{arg max}} \log p_\theta(D)$

Factor Analysis

$$p_\theta(s) = \mathcal{N}(s; 0, I)$$

$$p_\theta(x \mid s) = \mathcal{N}\left(x; Ws + \mu, \Sigma\right)$$

Latent Variable Generative Models



Goal: $\theta^* = \underset{\theta \in \Theta}{\textbf{arg max}} \log p_\theta(D)$

VAE

$$p_\theta(s) = \mathcal{N}(s; 0, I)$$

$$p_\theta(x \mid s) = \mathcal{N}\left(x; f_{nn}^{\mu}(s), f_{nn}^{\sigma^2}(s)\right)$$

Identifiability

I. Khemakhem, et al. 2019

Identifiability

Identifiability: the "true" parameters will be recovered given infinite observations

I. Khemakhem, et al. 2019

Identifiability

Identifiability: the "true" parameters will be recovered given infinite observations

For generative models: $p_\theta(x) = p_{\theta'}(x) \implies \theta = \theta'$

I. Khemakhem, et al. 2019

Identifiability

Identifiability: the "true" parameters will be recovered given infinite observations

For generative models: $p_\theta(x) = p_{\theta'}(x) \implies \theta = \theta'$

Identifiable models provide:

- *A principled approach to representation learning (as opposed to "just" generative modelling).*
- *Links to other unsupervised desiderata (e.g., "disentanglement")*

I. Khemakhem, et al. 2019

PCA Non-Identifiability

PCA Non-Identifiability

PCA / Factor Analysis:

PCA Non-Identifiability

PCA / Factor Analysis:

Scaling: $$WS = \frac{\alpha}{\alpha} WS = (\alpha^{-1} W)(\alpha S)$$

PCA Non-Identifiability

PCA / Factor Analysis:

Scaling: $$WS = \frac{\alpha}{\alpha} WS = (\alpha^{-1} W)(\alpha S)$$

Rotation: $$WS = WRR^{-1}S = W'S'$$

VAE Non-identifiability

F. Locatello et al. 2019.

VAE Non-identifiability

Theorem. Let $p(s) = \prod p_d(s_d)$**, then there exists an infinite family of bijections of the form** $f\colon \mathcal{S} \to \mathcal{S}$ **such that**

$$\int p_\theta(x \mid s)p(s)\mathrm{d}s = \int p_{\theta'}(x \mid f(s))p(f(s))\mathrm{d}s$$

for some alternative generative model with parameters θ'**.**

F. Locatello et al. 2019.

VAE Non-identifiability

Theorem. Let $p(s) = \prod p_d(s_d)$ **, then there exists an infinite family of bijections of the form** $f\colon \mathcal{S} \rightarrow \mathcal{S}$ **such that**

$$\int p_\theta(x \,|\, s) p(s) \mathrm{d}s = \int p_{\theta'}(x \,|\, f(s)) p(f(s)) \mathrm{d}s$$

for some alternative generative model with parameters θ'**.**

$p_\theta(x)$

F. Locatello et al. 2019.

VAE Non-identifiability

Theorem. Let $p(s) = \prod p_d(s_d)$ **, then there exists an infinite family of bijections of the form** $f: \mathcal{S} \rightarrow \mathcal{S}$ **such that**

$$\int p_\theta(x \mid s) p(s) \mathrm{d}s = \int p_{\theta'}(x \mid f(s)) p(f(s)) \mathrm{d}s$$

for some alternative generative model with parameters θ'**.**

$$p_\theta(x) \approx p_{\theta^*}(x)$$

F. Locatello et al. 2019.

VAE Non-identifiability

Theorem. Let $p(s) = \prod p_d(s_d)$ **, then there exists an infinite family of bijections of the form** $f\colon \mathcal{S} \to \mathcal{S}$ **such that**

$$\int p_\theta(x \mid s) p(s) \mathrm{d}s = \int p_{\theta'}(x \mid f(s)) p(f(s)) \mathrm{d}s$$

for some alternative generative model with parameters θ'**.**

$$p_\theta(x) \approx p_{\theta^*}(x)$$

$$p_\theta(s, x)$$

F. Locatello et al. 2019.

VAE Non-identifiability

Theorem. Let $p(s) = \prod p_d(s_d)$ **, then there exists an infinite family of bijections of the form** $f\colon \mathcal{S} \to \mathcal{S}$ **such that**

$$\int p_\theta(x \mid s) p(s) \mathrm{d}s = \int p_{\theta'}(x \mid f(s)) p(f(s)) \mathrm{d}s$$

for some alternative generative model with parameters θ'**.**

$$p_\theta(x) \approx p_{\theta^*}(x)$$

$$p_\theta(s, x) \not\approx p_{\theta^*}(s, x)$$

F. Locatello et al. 2019.

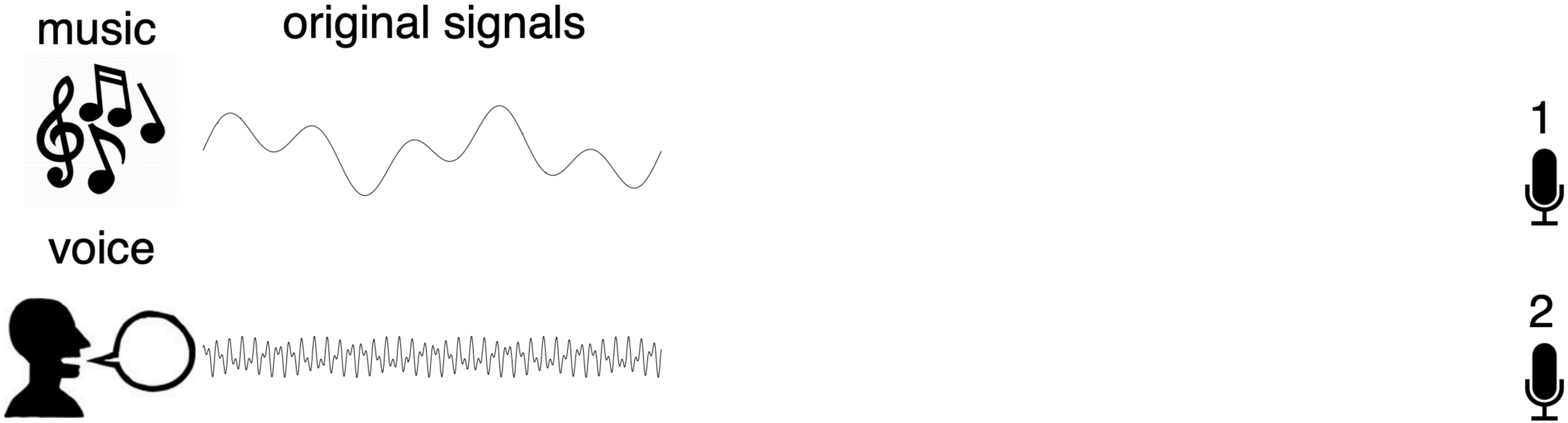
The Cocktail Party Problem

J. Shlens. 2014

The Cocktail Party Problem

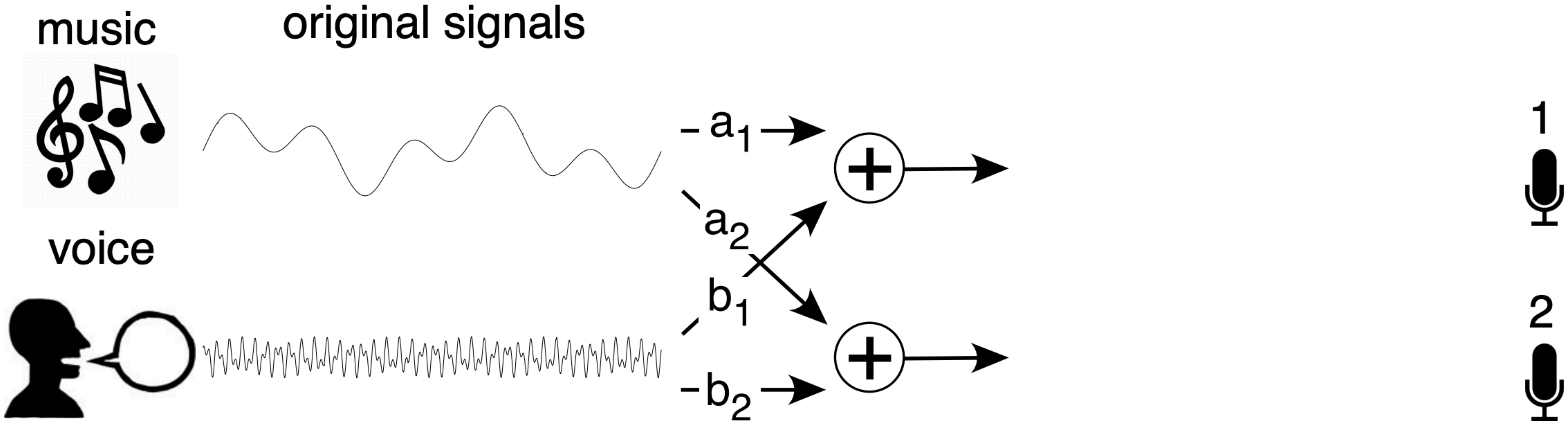
J. Shlens. 2014

The Cocktail Party Problem



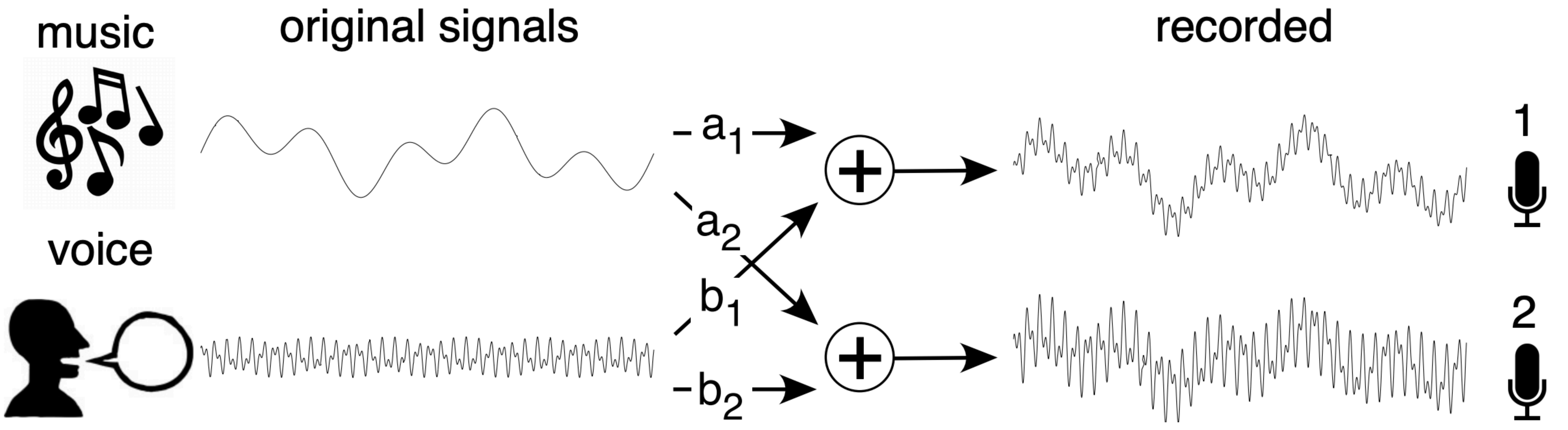
J. Shlens. 2014

The Cocktail Party Problem



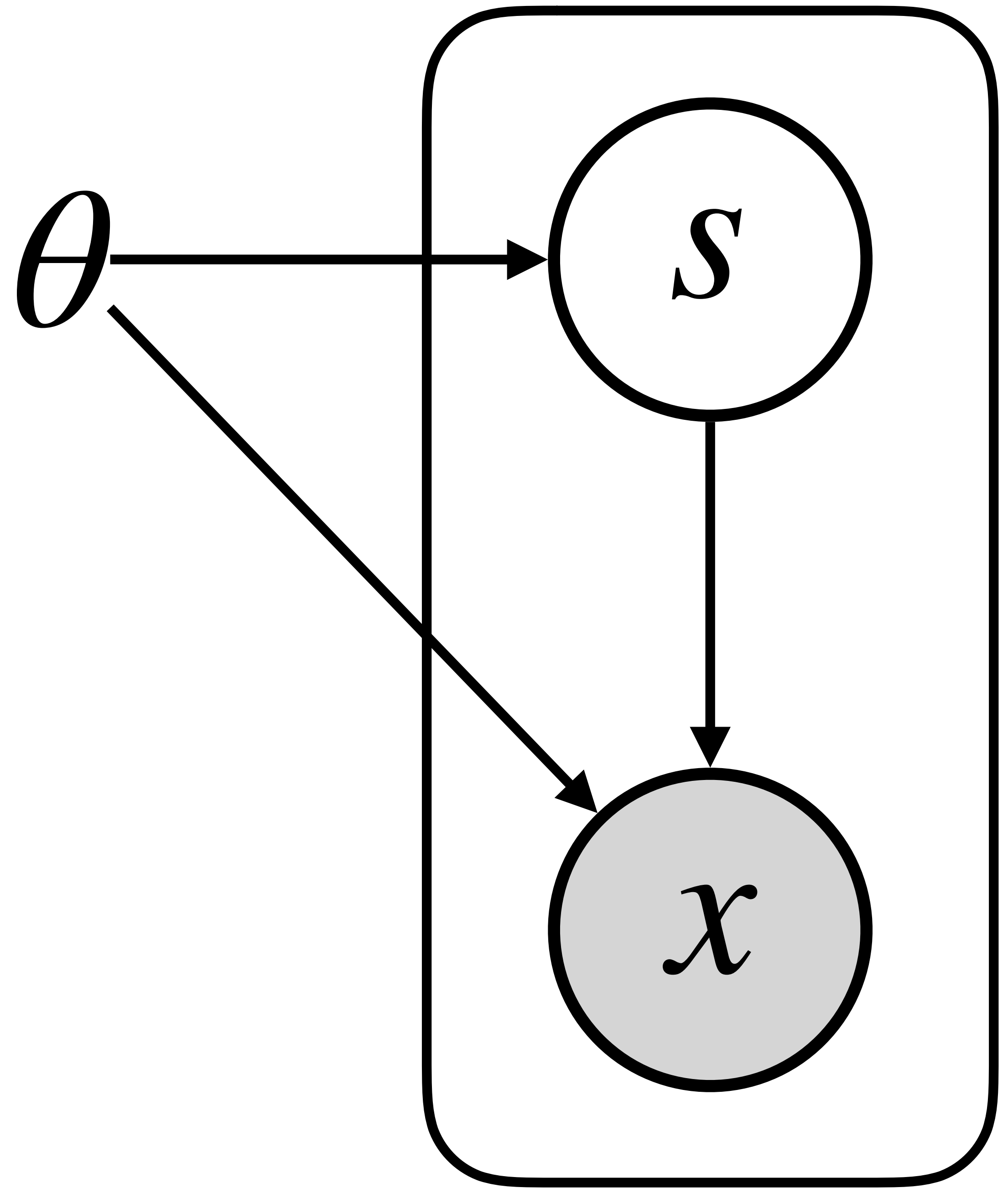
J. Shlens. 2014

The Cocktail Party Problem

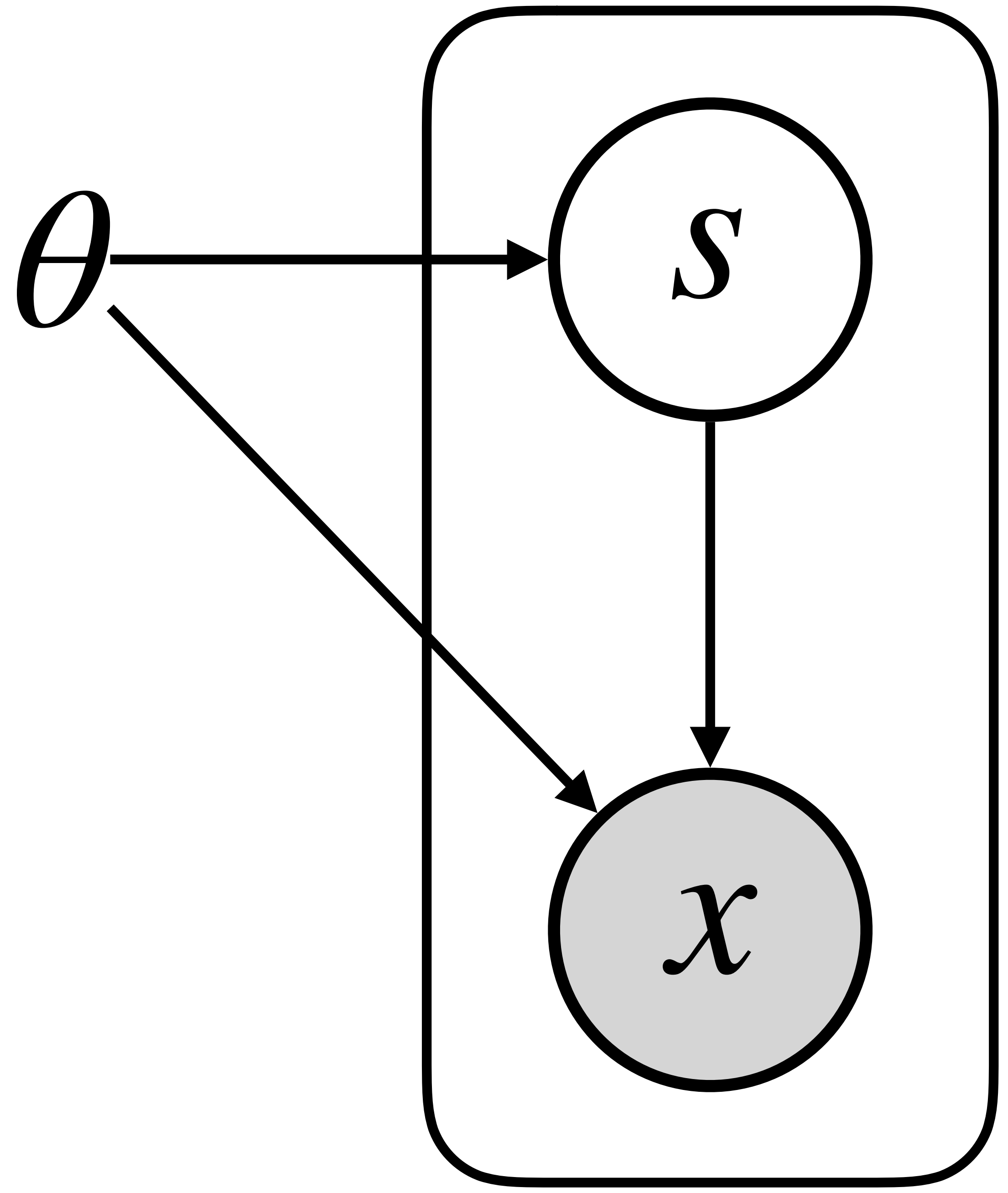


J. Shlens. 2014

Linear ICA

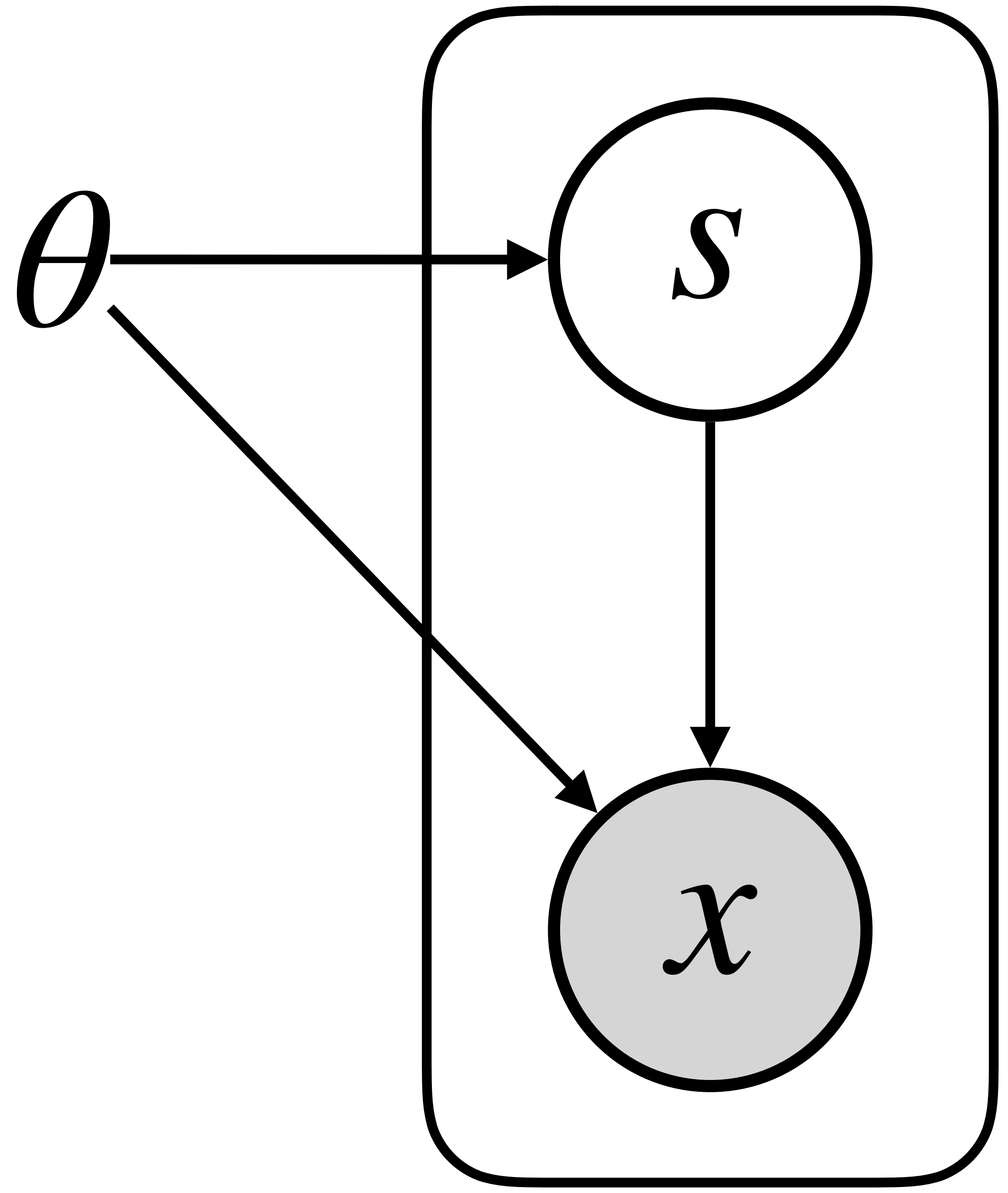


Linear ICA



$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i; \theta)$$

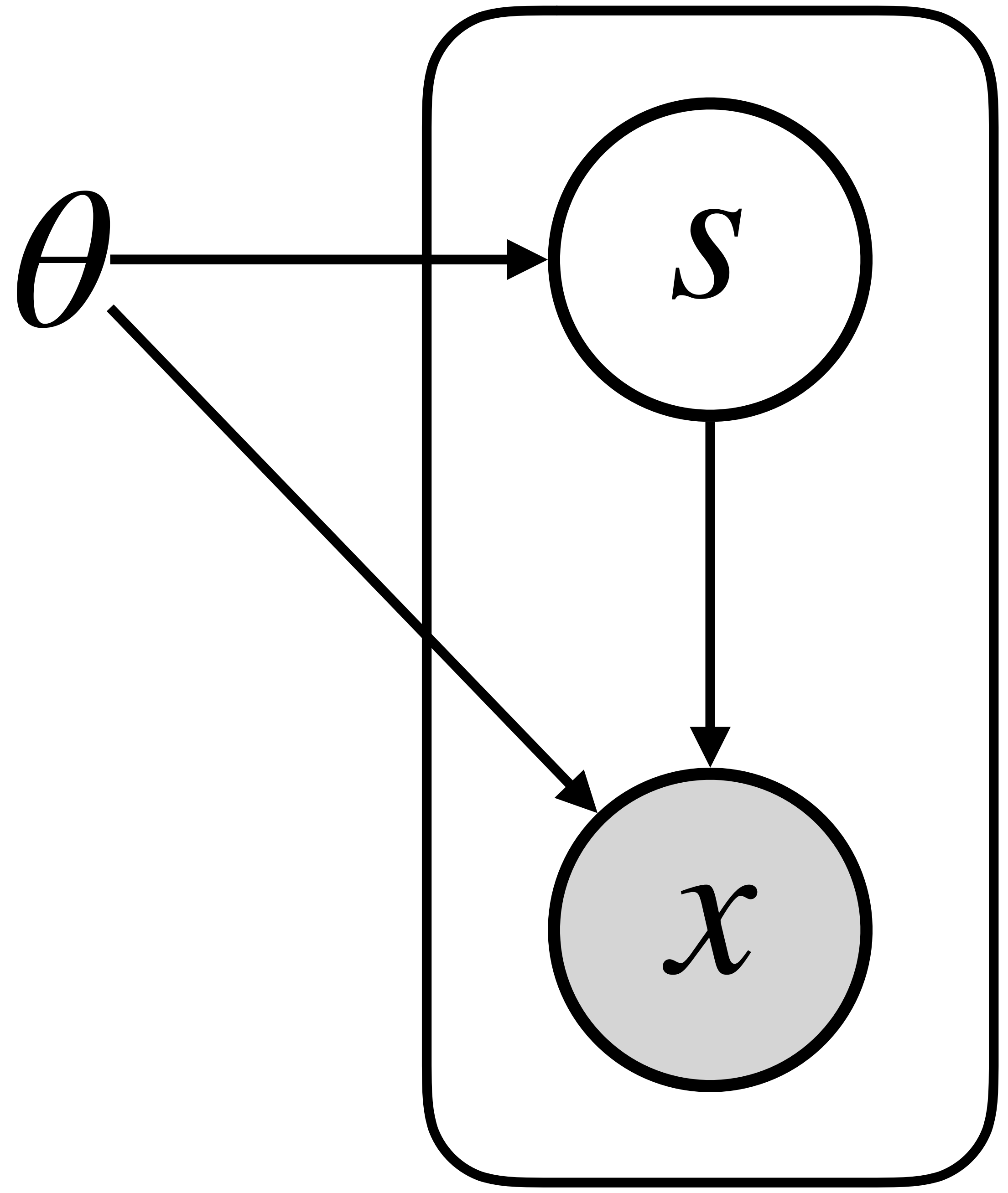
Linear ICA



$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i; \theta)$$

$$x = As; \quad A \in \mathbb{R}^{d \times d}$$

Linear ICA



Goal: recover $A^{-1} \implies s = A^{-1}x$

$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i; \theta)$$

$$x = As; \quad A \in \mathbb{R}^{d \times d}$$

Maximum Likelihood Solution

D. Mackay. 2003.

Maximum Likelihood Solution

$$W = A^{-1};$$

D. Mackay. 2003.

Maximum Likelihood Solution

$$W = A^{-1}; \qquad a_i = \sum_j W_{ij} x_j$$

D. Mackay. 2003.

Maximum Likelihood Solution

$$W = A^{-1};$$

$$a_i = \sum_j W_{ij} x_j$$

$$\phi(a_i) = \frac{\mathrm{d}}{\mathrm{d}a_i} \log p_i(a_i)$$

D. Mackay. 2003.

Maximum Likelihood Solution

$$W = A^{-1};\qquad a_i = \sum_j W_{ij} x_j \qquad \phi(a_i) = \frac{\mathrm{d}}{\mathrm{d}a_i} \log p_i(a_i)$$

$$\log p(x^{(n)} \,|\, A) = \log |\det W| + \sum_i \log p_i(a_i^{(n)})$$

D. Mackay. 2003.

Maximum Likelihood Solution

$$W = A^{-1};$$

$$a_i = \sum_j W_{ij} x_j$$

$$\phi(a_i) = \frac{\mathrm{d}}{\mathrm{d}a_i} \log p_i(a_i)$$

$$\log p(x^{(n)} | A) = \log |\det W| + \sum_i \log p_i(a_i^{(n)})$$

$$\frac{\partial}{\partial W_{ij}} \log p(x^{(n)} | A) = [W^T]_{ij}^{-1} + x_j^{(n)} \phi(a_i^{(n)})$$

D. Mackay. 2003.

Maximum Likelihood Solution

Initialize W

for $x \in X; \mathrm{do}$

$$a \leftarrow Wx$$

$$s \leftarrow \phi(a)$$

$$W \leftarrow W + \eta([W^T]^{-1} + sx^T)$$

D. Mackay. 2003.

Maximum Likelihood Solution

Initialize W

for $x \in X;\mathrm{do}$

$$a \leftarrow Wx$$

$$s \leftarrow \phi(a)$$

$$W \leftarrow W + \eta([W^T]^{-1} + sx^T)$$

Choice of $p \iff \phi$

D. Mackay. 2003.

Maximum Likelihood Solution

Initialize W

for $x \in X; \mathrm{do}$

$$a \leftarrow Wx$$

$$s \leftarrow \phi(a)$$

$$W \leftarrow W + \eta([W^T]^{-1} + sx^T)$$

Choice of $p \iff \phi$

Identifiable when:

1. p_i *is not Gaussian (except perhaps one).*
2. $p_i \perp p_j$ *for all* i, j

D. Mackay. 2003.

Non-linear ICA

A. Hyvarinen and P. Pajunen. 1998.

Non-linear ICA

$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i; \theta)$$

A. Hyvarinen and P. Pajunen. 1998.

Non-linear ICA

$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i; \theta)$$

$$x = f(s; \theta); \quad f\colon \mathbb{R}^d \to \mathbb{R}^d$$

A. Hyvarinen and P. Pajunen. 1998.

Non-linear ICA

$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i; \theta)$$

$$x = f(s; \theta); \quad f\colon \mathbb{R}^d \to \mathbb{R}^d$$

Goal: recover f^{-1}

A. Hyvarinen and P. Pajunen. 1998.

Non-linear ICA

Existence: *for any random variable* $x \in \mathbb{R}^d$*, there exists a function* $g: \mathscr{X} \to \mathscr{S}$ *such that* $s_1, \ldots, s_d$ ***have density*** $p_\theta(s)$ **(constructive).**

$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i; \theta)$$

$$x = f(s; \theta); \quad f\colon \mathbb{R}^d \to \mathbb{R}^d$$

Goal: recover f^{-1}

A. Hyvarinen and P. Pajunen. 1998.

Non-linear ICA

Existence: *for any random variable* $x \in \mathbb{R}^d$, *there exists a function* $g: \mathcal{X} \to \mathcal{S}$ *such that* $s_1, \ldots, s_d$ ***have density*** $p_\theta(s)$ **(constructive).**

Non-uniqueness: *the number of solutions* g *is at least as large as the class of measure-preserving functions* $h: [0,1]^n \to [0,1]^n$.

$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i; \theta)$$

$$x = f(s; \theta); \quad f: \mathbb{R}^d \to \mathbb{R}^d$$

Goal: recover f^{-1}

A. Hyvarinen and P. Pajunen. 1998.

Nonstationary Nonlinear ICA

A. Hyvarinen and H. Morioka. 2016.

Nonstationary Nonlinear ICA

$$s_t = \left(s_1(t), \ldots, s_n(t)\right)$$

A. Hyvarinen and H. Morioka. 2016.

Nonstationary Nonlinear ICA

$$s_t = \big(s_1(t), \ldots, s_n(t)\big)$$

$$p_{i,\tau}(s_i) \propto \lambda_i(\tau) q_i(s_i)$$

A. Hyvarinen and H. Morioka. 2016.

Nonstationary Nonlinear ICA

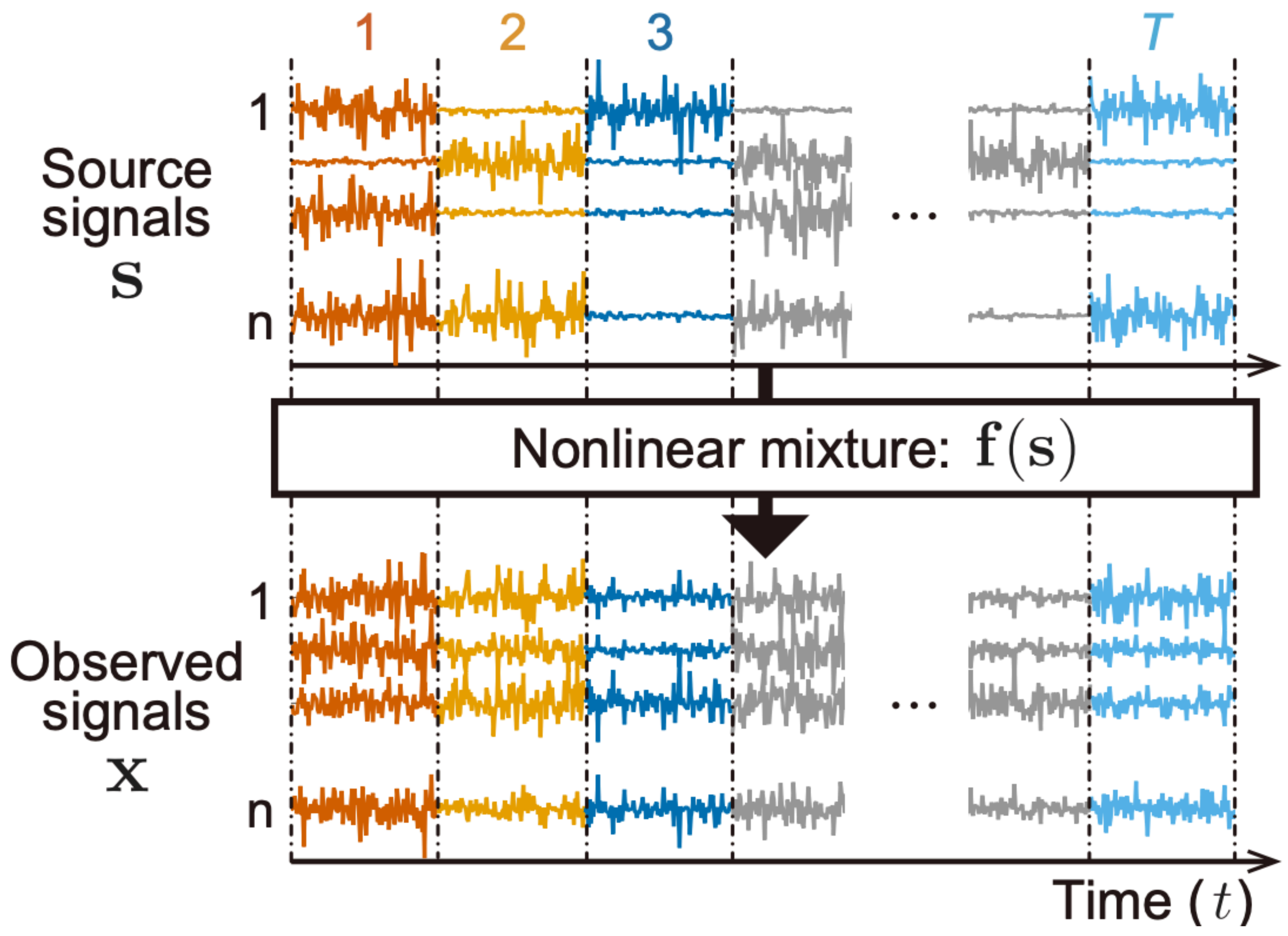
$$s_t = \left(s_1(t), \ldots, s_n(t)\right)$$

$$p_{i,\tau}(s_i) \propto \lambda_i(\tau) q_i(s_i)$$

$$x_t = f(s_t)$$

A. Hyvarinen and H. Morioka. 2016.

Nonstationary Nonlinear ICA



$$s_t = \big(s_1(t), \ldots, s_n(t)\big)$$

$$p_{i,\tau}(s_i) \propto \lambda_i(\tau) q_i(s_i)$$

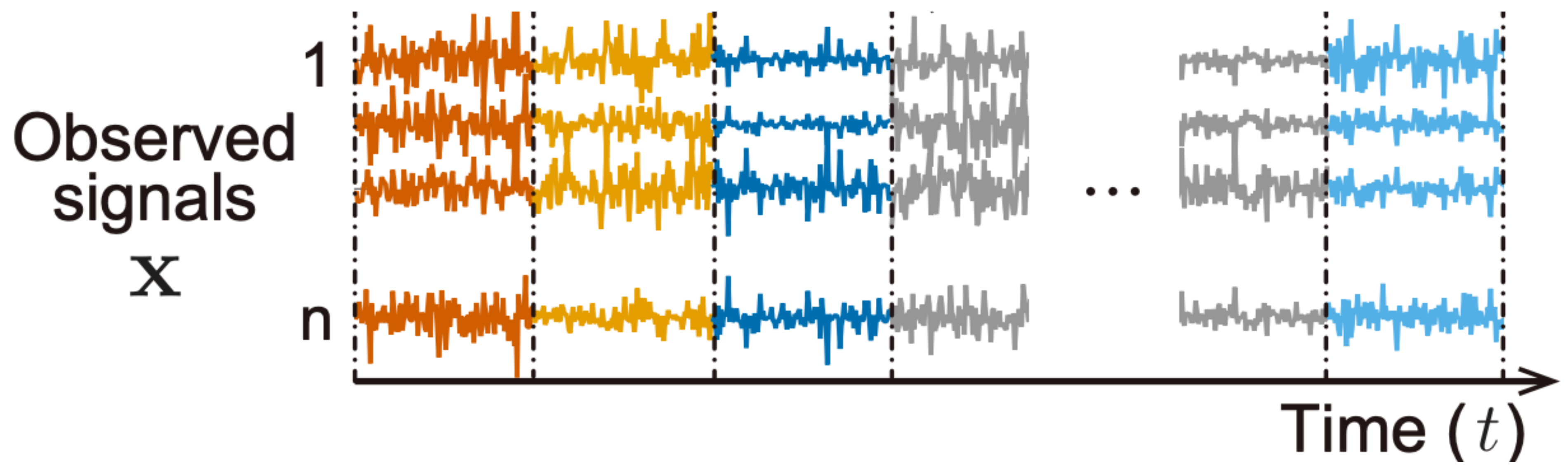
$$x_t = f(s_t)$$

A. Hyvarinen and H. Morioka. 2016.

Time-Contrastive Learning

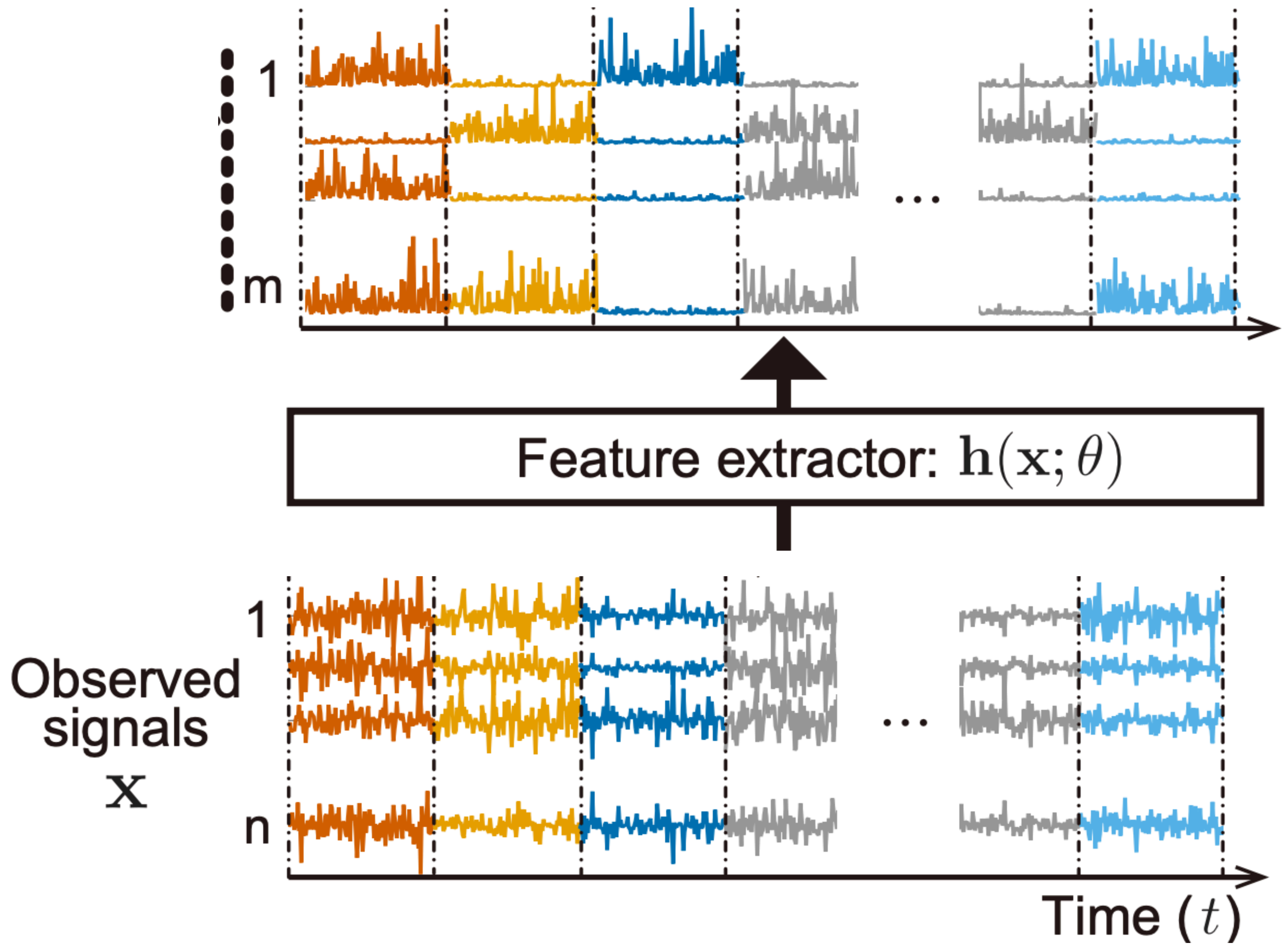
A. Hyvarinen and H. Morioka. 2016.

Time-Contrastive Learning



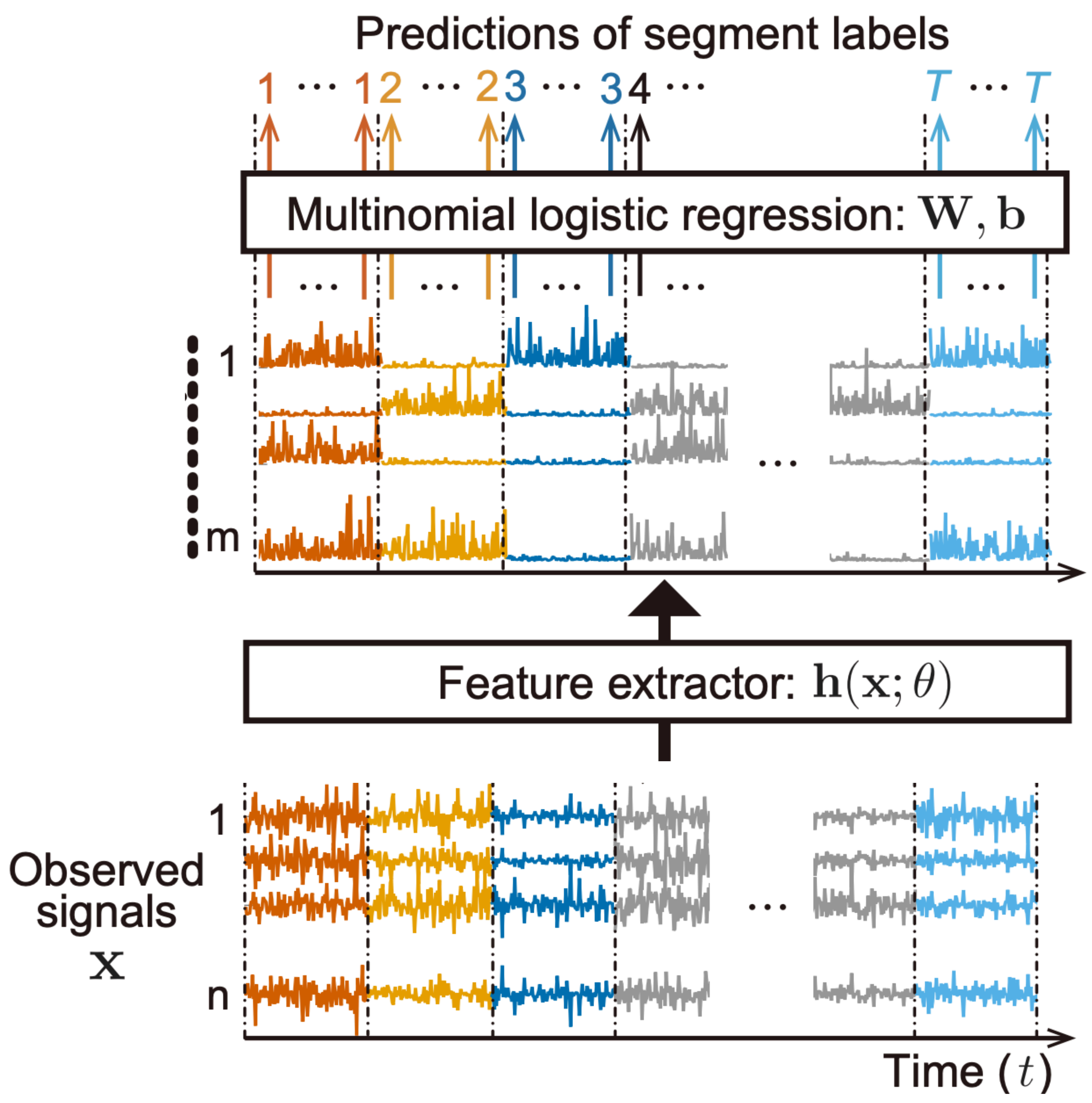
A. Hyvarinen and H. Morioka. 2016.

Time-Contrastive Learning



A. Hyvarinen and H. Morioka. 2016.

Time-Contrastive Learning



A. Hyvarinen and H. Morioka. 2016.

Identifiability via TCL

A. Hyvarinen and H. Morioka. 2016.

Identifiability via TCL

Theorem. Assume that

- *(i)* **the observed data are generated from the detailed model,**
- *(ii)* **TCL is applied to learn a feature extractor $h(x_t; \theta)$, and**
- *(iii)* **the parameters $\lambda_{i,v}$ are "well-behaved".**

Then, with infinite data we have that

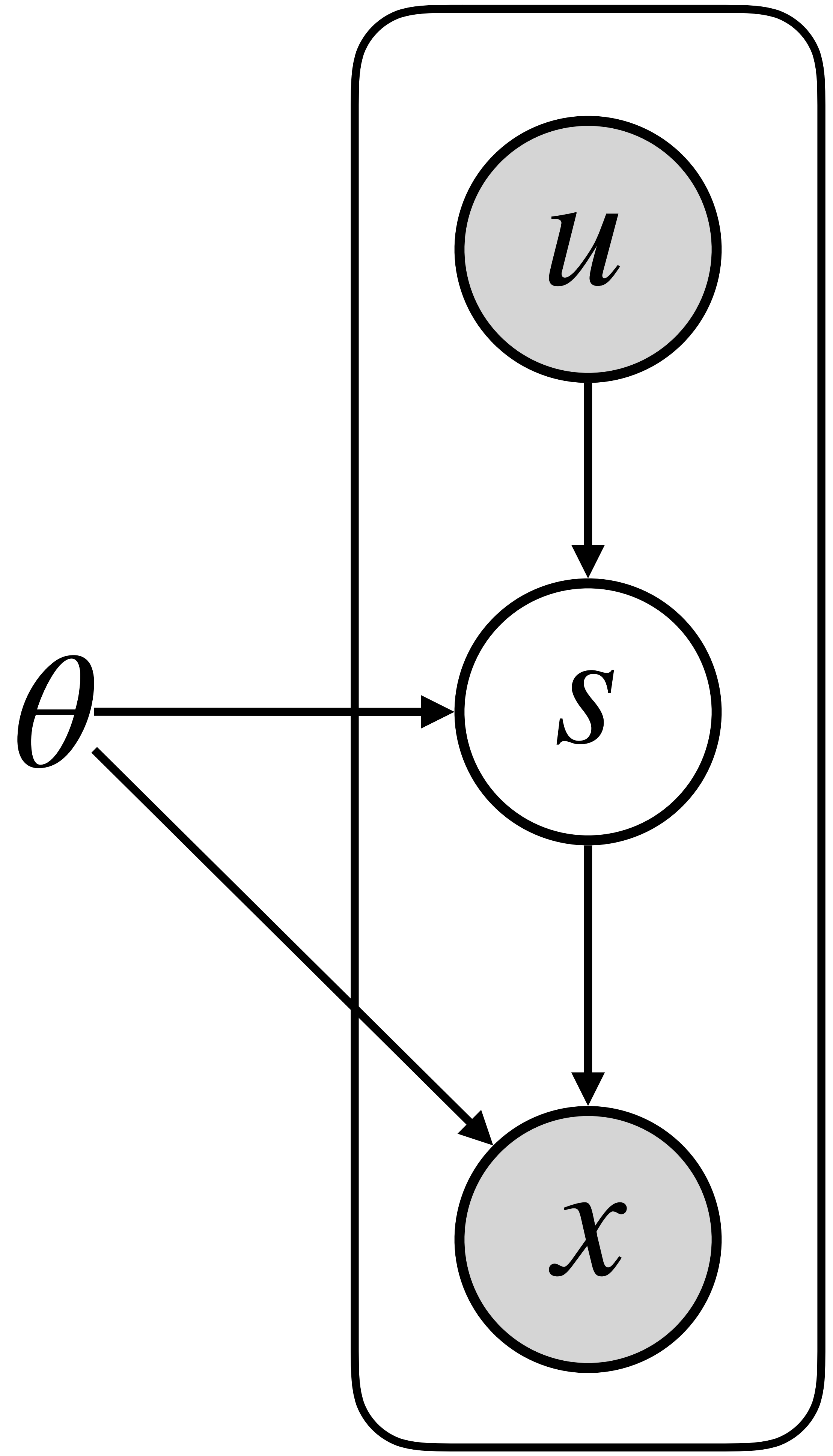
$$q(s_t) = Ah(x; \theta) + b\,.$$

A. Hyvarinen and H. Morioka. 2016.

Generalization with Auxiliary Variables

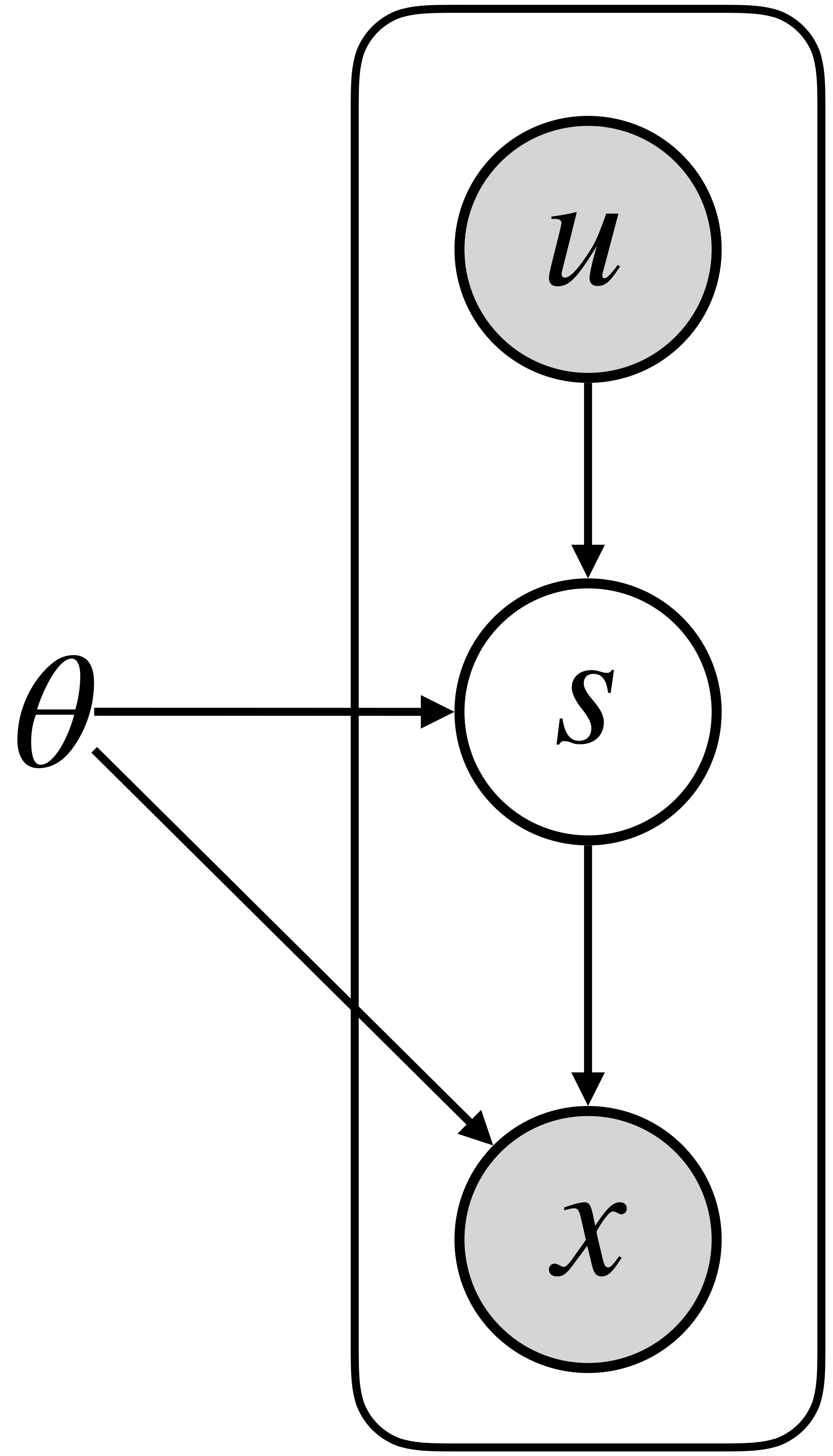
A. Hyvarinen et al. 2019., Khemakhem et al., 2019

Generalization with Auxiliary Variables



A. Hyvarinen et al. 2019., Khemakhem et al., 2019

Generalization with Auxiliary Variables

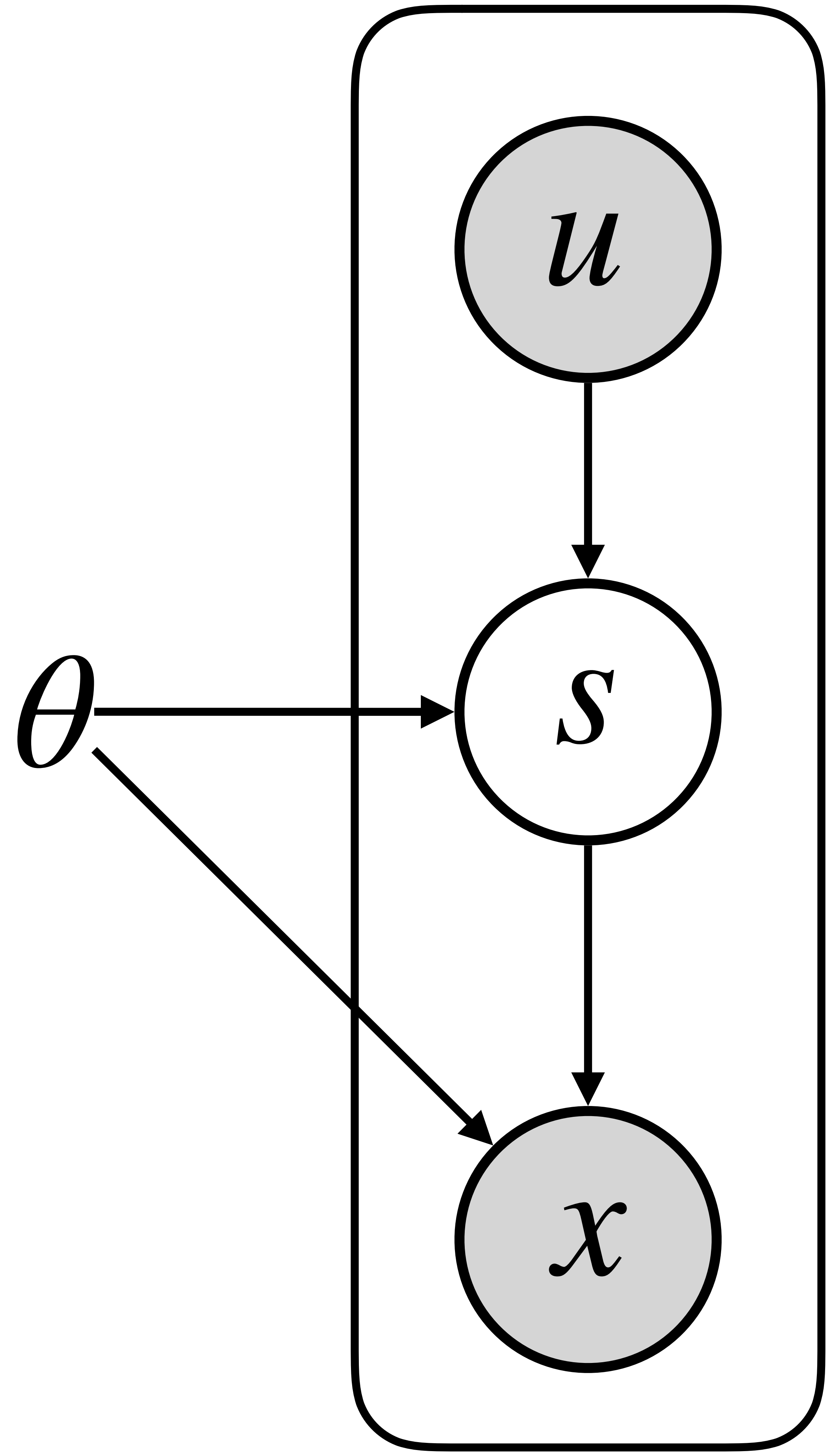


$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i \mid u; \theta)$$

$$x = f(s; \theta); \quad f\colon \mathbb{R}^d \to \mathbb{R}^d$$

A. Hyvarinen et al. 2019., Khemakhem et al., 2019

Generalization with Auxiliary Variables



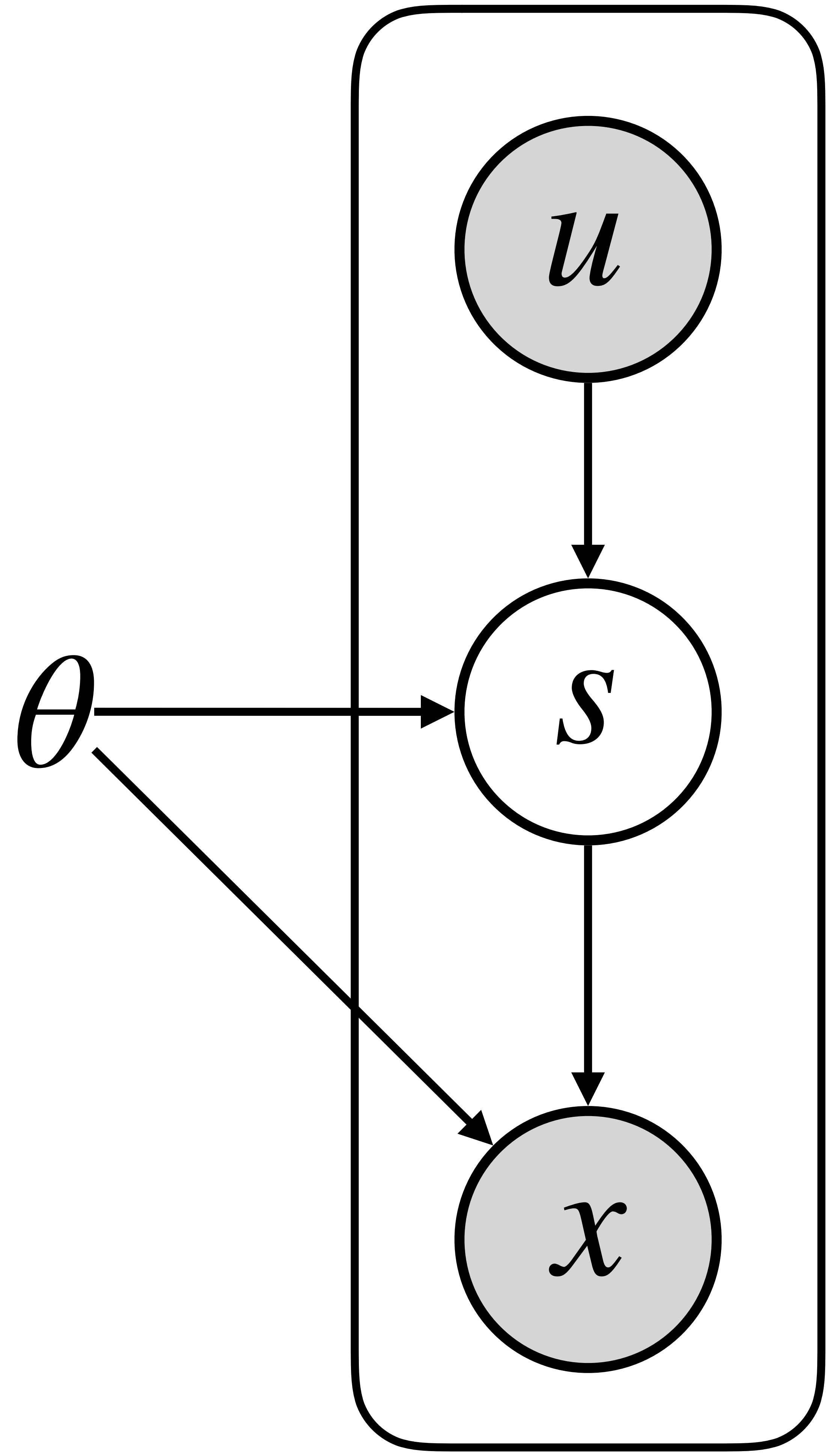
$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i \mid u; \theta)$$

$$x = f(s; \theta); \quad f\colon \mathbb{R}^d \to \mathbb{R}^d$$

$$\tilde{x} = (x, u); \quad \tilde{x}^* = (x, u^*)$$

A. Hyvarinen et al. 2019., Khemakhem et al., 2019

Generalization with Auxiliary Variables



$$p_\theta(s) = \prod_{i=1}^{d} p_i(s_i \,|\, u; \theta)$$

$$x = f(s; \theta); \quad f\colon\, \mathbb{R}^d \to \mathbb{R}^d$$

$$r(x, u) = \sum_{i=1}^{n} \psi_i(h(x; \theta), u)$$

$$\tilde{x} = (x, u); \quad \tilde{x}^* = (x, u^*)$$

A. Hyvarinen et al. 2019., Khemakhem et al., 2019

Summary

Summary

- *Contrastive learning —> Self-supervised "heuristics"*

Summary

- *Contrastive learning —> Self-supervised “heuristics”*
- *SSL methods seem to lead to useful representations*

Summary

- *Contrastive learning —> Self-supervised “heuristics”*
- *SSL methods seem to lead to useful representations*
- *Methods largely motivated (heuristically) by mutual information*

Summary

- *Contrastive learning —> Self-supervised “heuristics”*
- *SSL methods seem to lead to useful representations*
- *Methods largely motivated (heuristically) by mutual information*
- *Self-supervised learning <—> identifiability in generative models*

References

- F. Locatello et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. 2019.
- J. Shlens. A Tutorial on Independent Component Analysis. 2014.
- D. Mackay. Information Theory, Inference, and Learning Algorithms. 2003.
- A. Hyvarinen and P. Pajunen. Nonlinear ICA: Existence and Uniqueness Results. 1998.
- A. Hyvarinen and H. Morioka. Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA. 2016.
- A. Hyvarinen et al. Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning. 2019.
- I. Khemakhem et al. VAEs and Nonlinear ICA: a Unifying Framework. 2019.