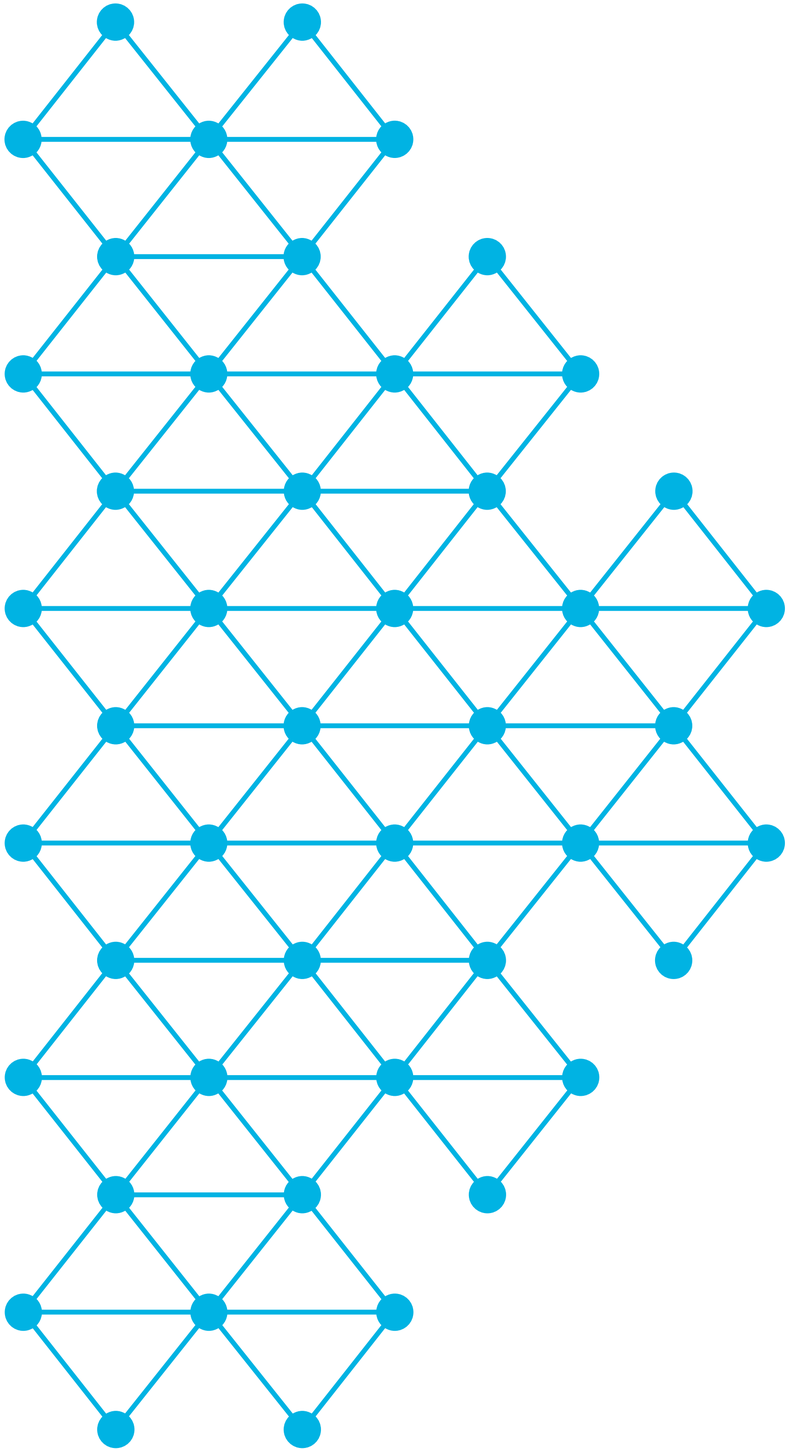
Overview of Self-supervised Representation Learning

Aaron Courville
Université de Montréal

IFT6268 - Self-Supervised Representation Learning

Slides and slide material from Devon Hjelm, Samuel Lavoie and Faruk Ahmed.



Self-supervised learning as an instance of Transfer Learning

Transfer Learning

- The **domain** $\mathcal{D}$ consists of: a *feature space* $\mathcal{X}$ and a *marginal probability distribution* $P(X)$, where $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \in \mathcal{X}$.
- Given a specific domain, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a **task** consists of two components: a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$ (denoted by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$), which is learned from the training data, consisting of pairs $\{\boldsymbol{x}_i, y_i\}$, where $\boldsymbol{x}_i \in X$ and $y_i \in \mathcal{Y}$.
- The function $f(\cdot)$ can be used to predict the corresponding label, $f(\boldsymbol{x})$, of a new instance $\boldsymbol{x}$.
- Given a **source domain** $\mathcal{D}_S$ and **learning task** $\mathcal{T}_S$, a **target domain** $\mathcal{D}_T$ and **learning task** $\mathcal{T}_T$:

> **Transfer learning** aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$ or both.

Lin, Yuan-Pin; Jung, Tzyy-Ping (27 June 2017). "Improving EEG-Based Emotion Classification Using Conditional Transfer Learning". *Frontiers in Human Neuroscience*. **11**: 334. From Wikipedia.

Flavours of Transfer Learning

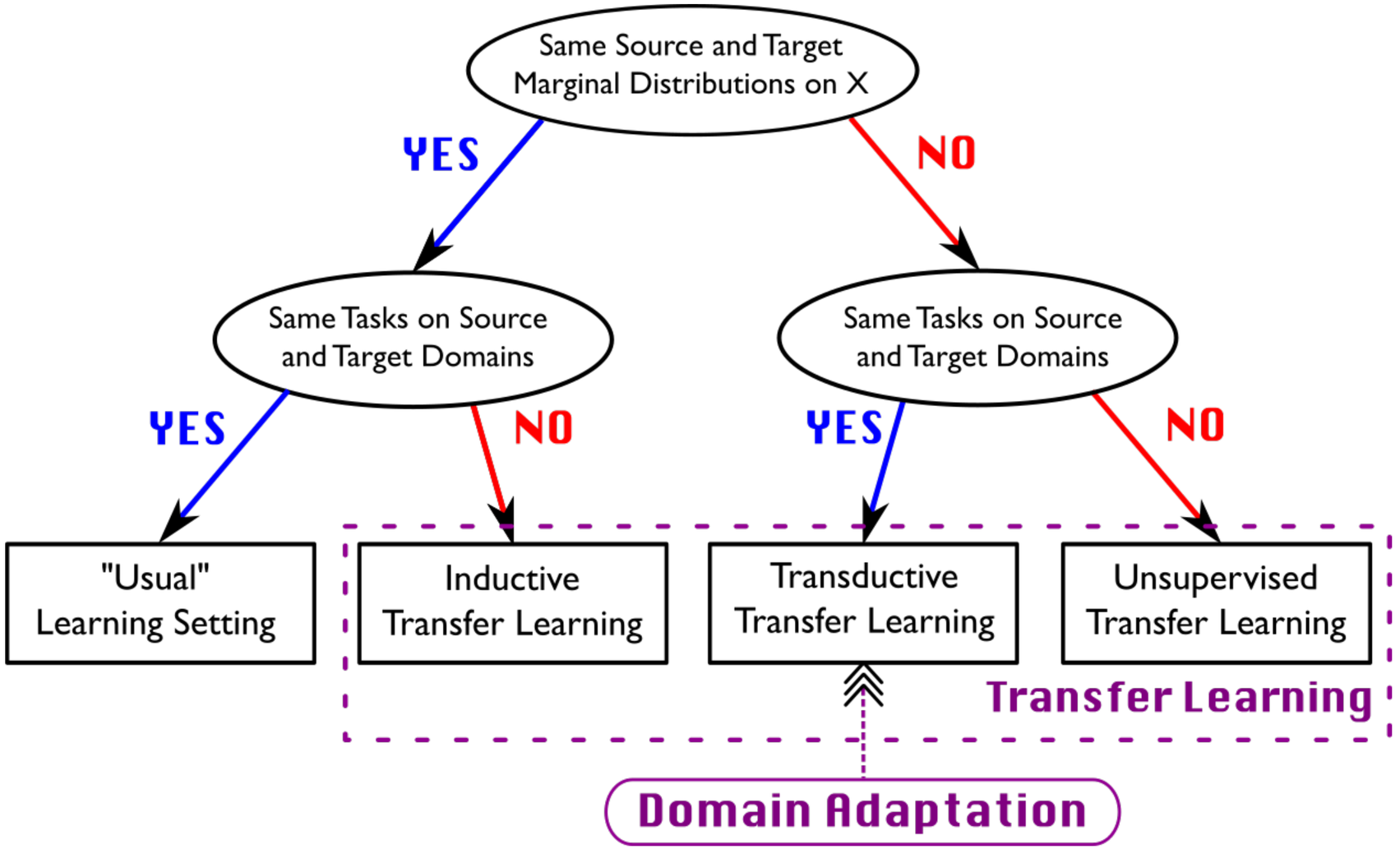


Image credit: Emilie Morvant - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=58812416

Transfer Learning / Domain Adaptation

- We can further subdivide the problem based on what side we have labeled data (source or target).
- In all cases, we want something that does well on the target.

- The four possibilities:
  - Source labeled, target labeled (S+T+)
  - Source labeled, target only unlabeled (S+T-)
  - Source only unlabeled, target labeled (S-T+)
  - Source only unlabeled, target only unlabeled (S-T-) — straight unsupervised learning?

Unsupervised DA

Self-supervised Learning

Hal Daume's Blog: https://nlpers.blogspot.com/2007/11/domain-adaptation-vs-transfer-learning.html

Flavours of Transfer Learning

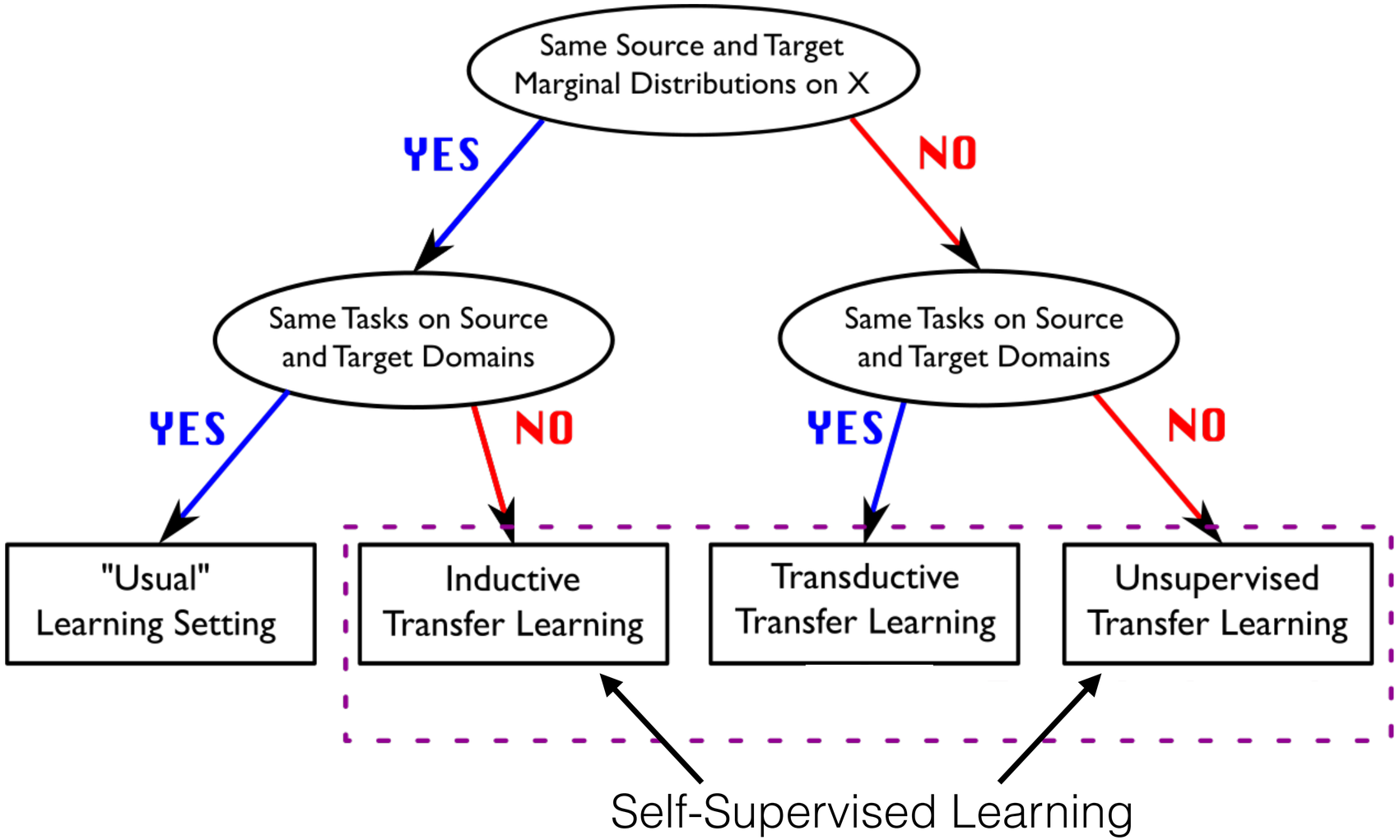


Image credit: Emilie Morvant - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=58812416

A transfer view on self-supervised learning

- Self-supervised learning is an instance of **unsupervised learning** methods in the sense that no external labels are needed to train the model parameters.

- Self-supervised learning is an instance of transfer learning where:

$$\mathcal{T}_S \neq \mathcal{T}_T$$

- Key characteristic of self-supervised learning is found in the definition of the source (training) task: $\mathcal{T}_S$

A transfer view on self-supervised learning

- For self-supervised learning, the source task, $\mathcal{T}_S$, is:

1. Unsupervised, I.e. it does not require external labels for training.

2. Designed to learn a representation that will extract discriminative (semantic) information from the input.

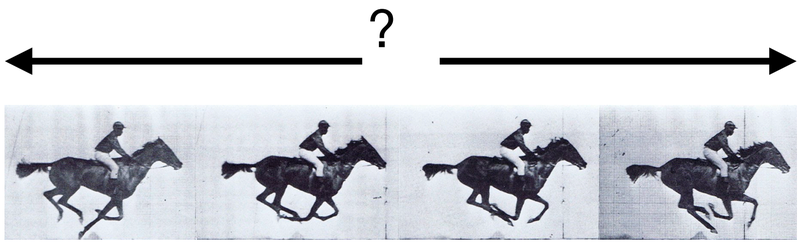
3. Designed to learn a representation that will possess appropriate invariances / equivariances
   - Eg. Invariance to low-level transformations of the input.

IFT6268: self-supervised representation learning

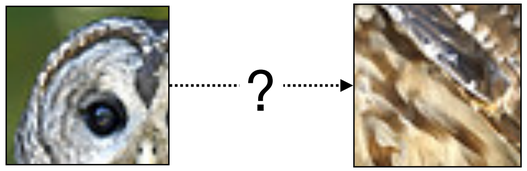
- In this class, take a broad-view of self-supervised learning.
- We are interested in the wide-array of methods that exist to learn, without labels, useful / effective representations for downstream tasks (eg. RL tasks, classification, structure output prediction, etc.)

Turning questions into self-supervision tasks

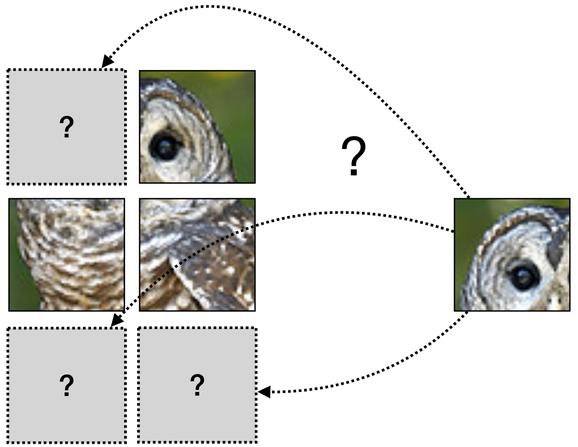
- **Question:** What direction is the video running?



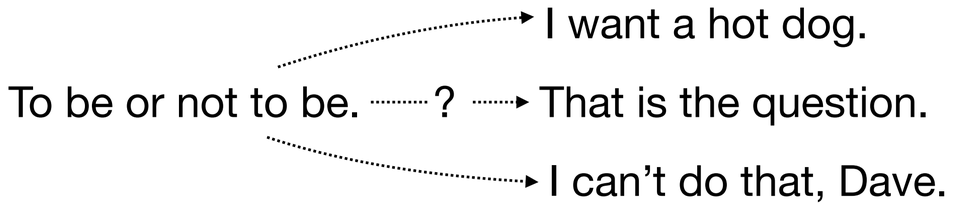
- **Question:** Do these image patches go together (context prediction)?



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

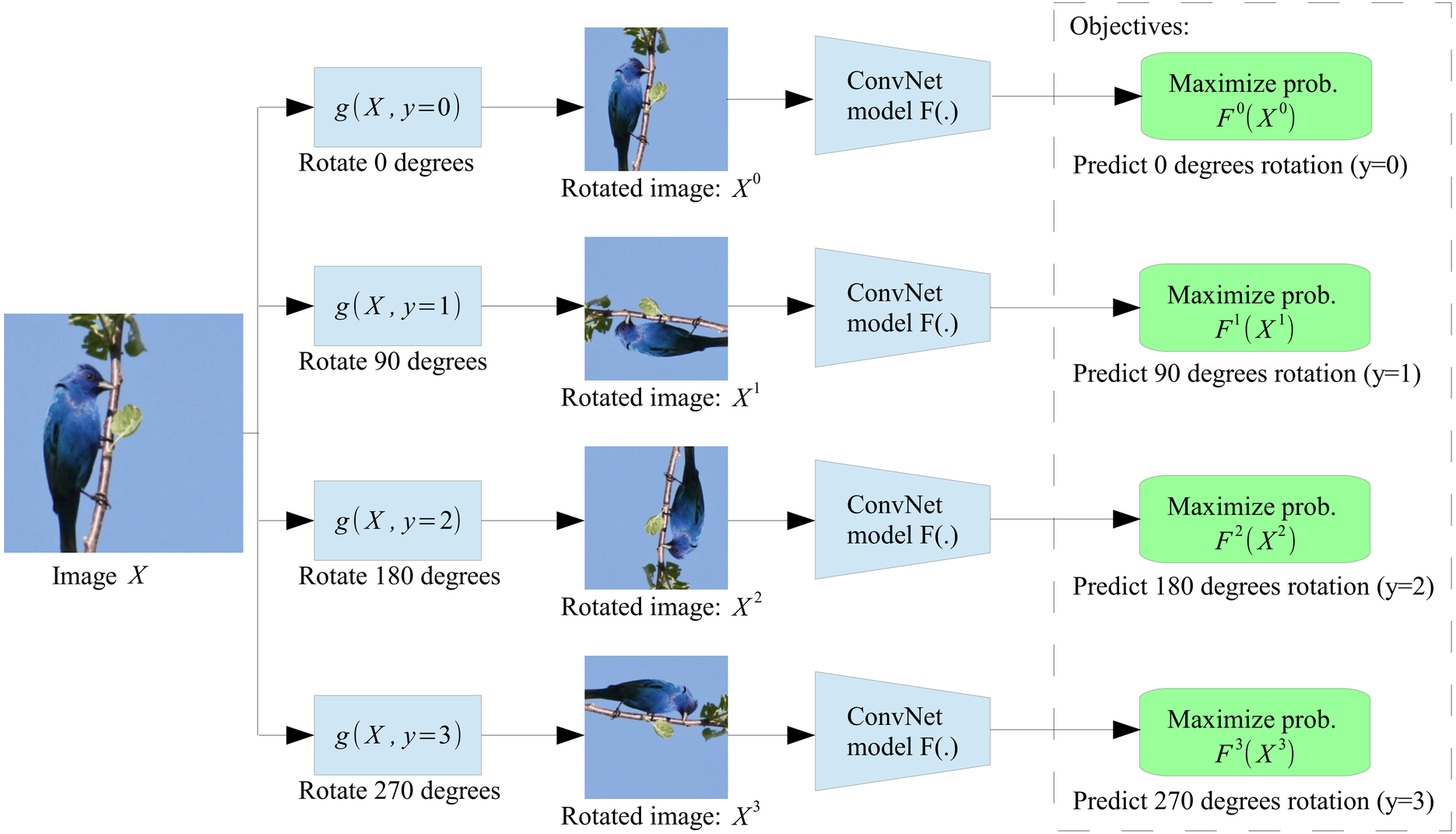


- **Question:** Which sentence follows this first one (Quick-thoughts)?



Rotation Prediction

Gidaris, Singh, and Komodakis. Unsupervised representation learning by predicting image rotations. ICLR 2018

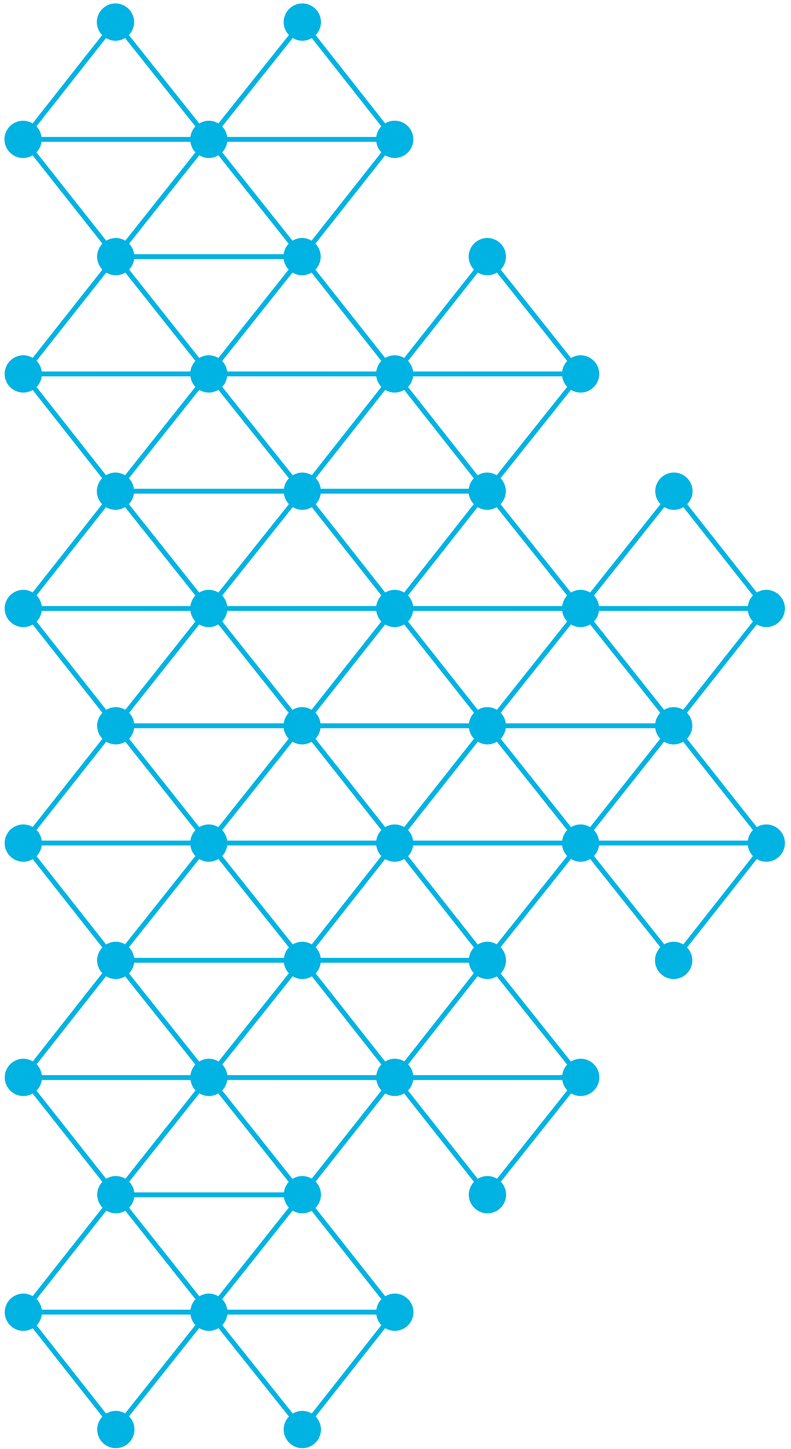


Rotation Prediction

Table 3: Evaluation of unsupervised feature learning methods on CIFAR-10. The *Supervised NIN* and the *(Ours) RotNet + conv* entries have exactly the same architecture but the first was trained fully supervised while on the second the first 2 conv. blocks were trained unsupervised with our rotation prediction task and the 3rd block only was trained in a supervised manner. In the *Random Init. + conv* entry a conv. classifier (similar to that of *(Ours) RotNet + conv*) is trained on top of two NIN conv. blocks that are randomly initialized and stay frozen. Note that each of the prior approaches has a different ConvNet architecture and thus the comparison with them is just indicative.

Method	Accuracy
Supervised NIN	92.80
Random Init. + conv	72.50
(Ours) RotNet + non-linear	89.06
(Ours) RotNet + conv	**91.16**
(Ours) RotNet + non-linear (fine-tuned)	91.73
(Ours) RotNet + conv (fine-tuned)	92.17
Roto-Scat + SVM Oyallon & Mallat (2015)	82.3
ExemplarCNN Dosovitskiy et al. (2014)	84.3
DCGAN Radford et al. (2015)	82.8
Scattering Oyallon et al. (2017)	84.7

Gidaris, Singh, and Komodakis. Unsupervised representation learning by predicting image rotations. ICLR 2018

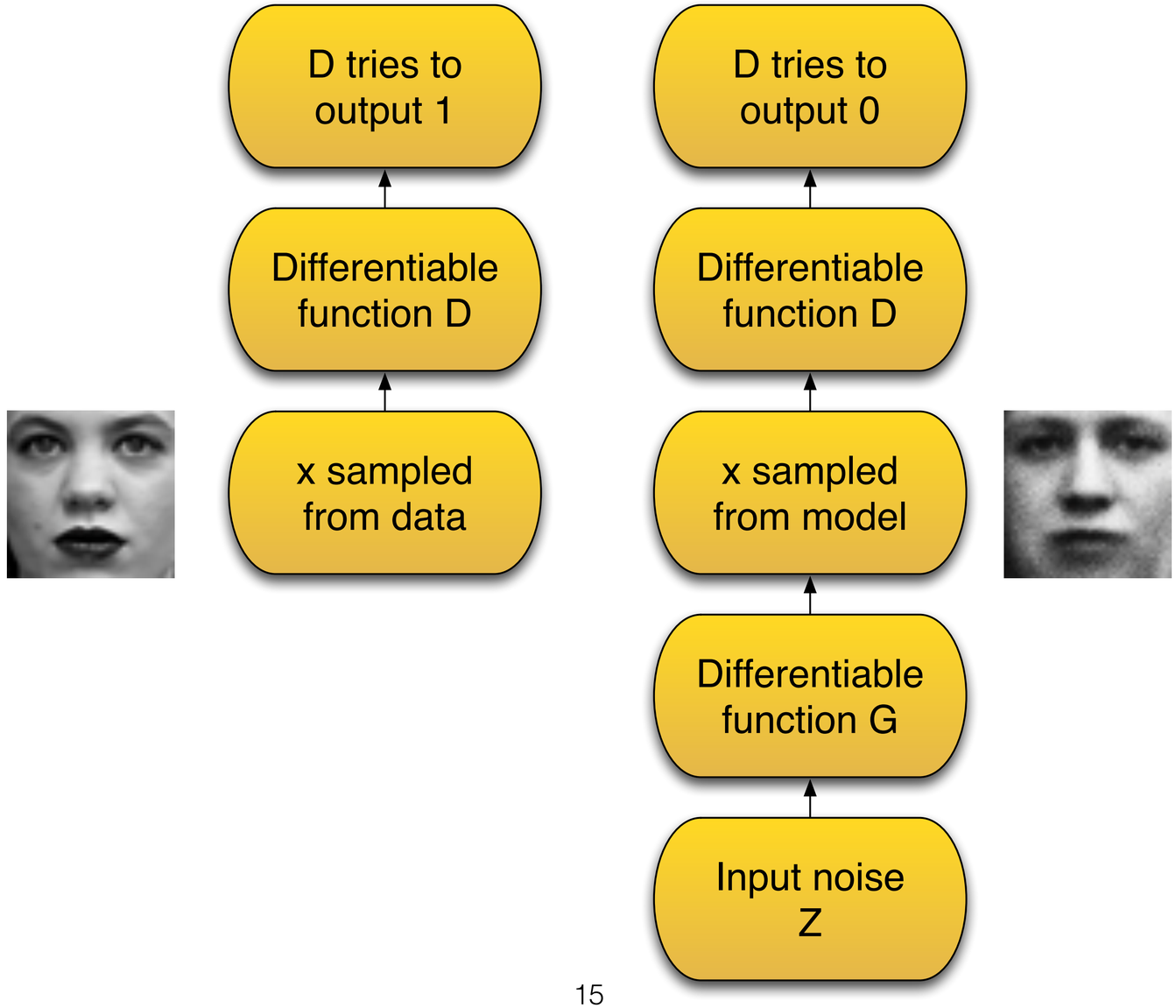


GAN: a self-supervised method?

Are GANs self-supervised learning methods?

- How open is your definition of self-supervised learning methods?
- If you take all unsupervised methods to be self-supervised, then yes, trivially.
  - Since GANs are an instance of an unsupervised learning method.
- In this class we take a slightly more restricted definition of self-supervised learning (as does most of the research community) that excluded generative modelling.
- So a GAN is not self-supervised methods, right?
- **It depends on how you use your GAN.**

Generative Adversarial Networks



GAN Objective

- Formally, express the game between discriminator D and generator G with the minimax objective:

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [\log(1 - D(\tilde{\boldsymbol{x}}))].$$

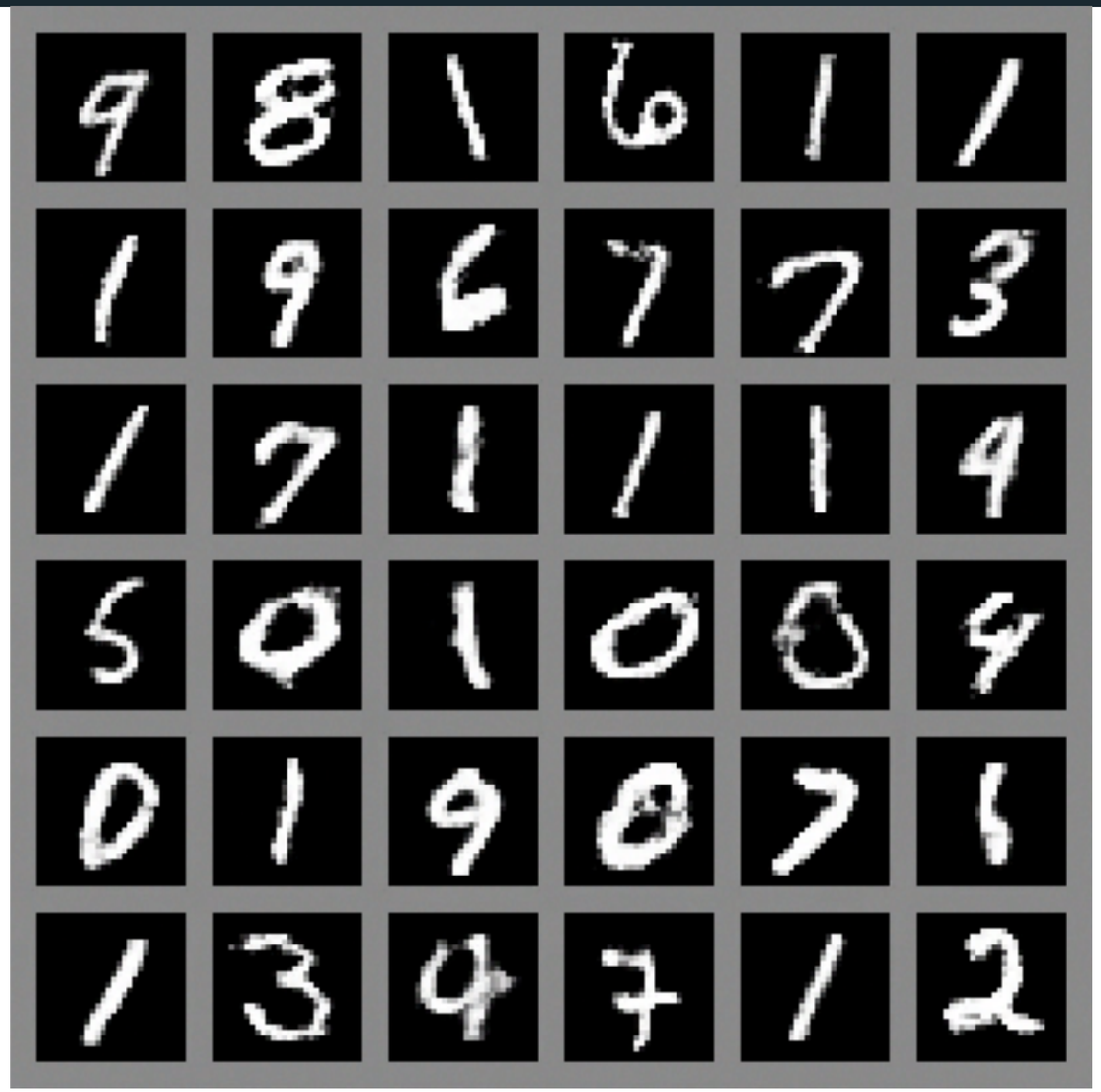
  where:

  - $\mathbb{P}_r$ is the data distribution
  - $\mathbb{P}_g$ is the model distribution implicitly defined by:

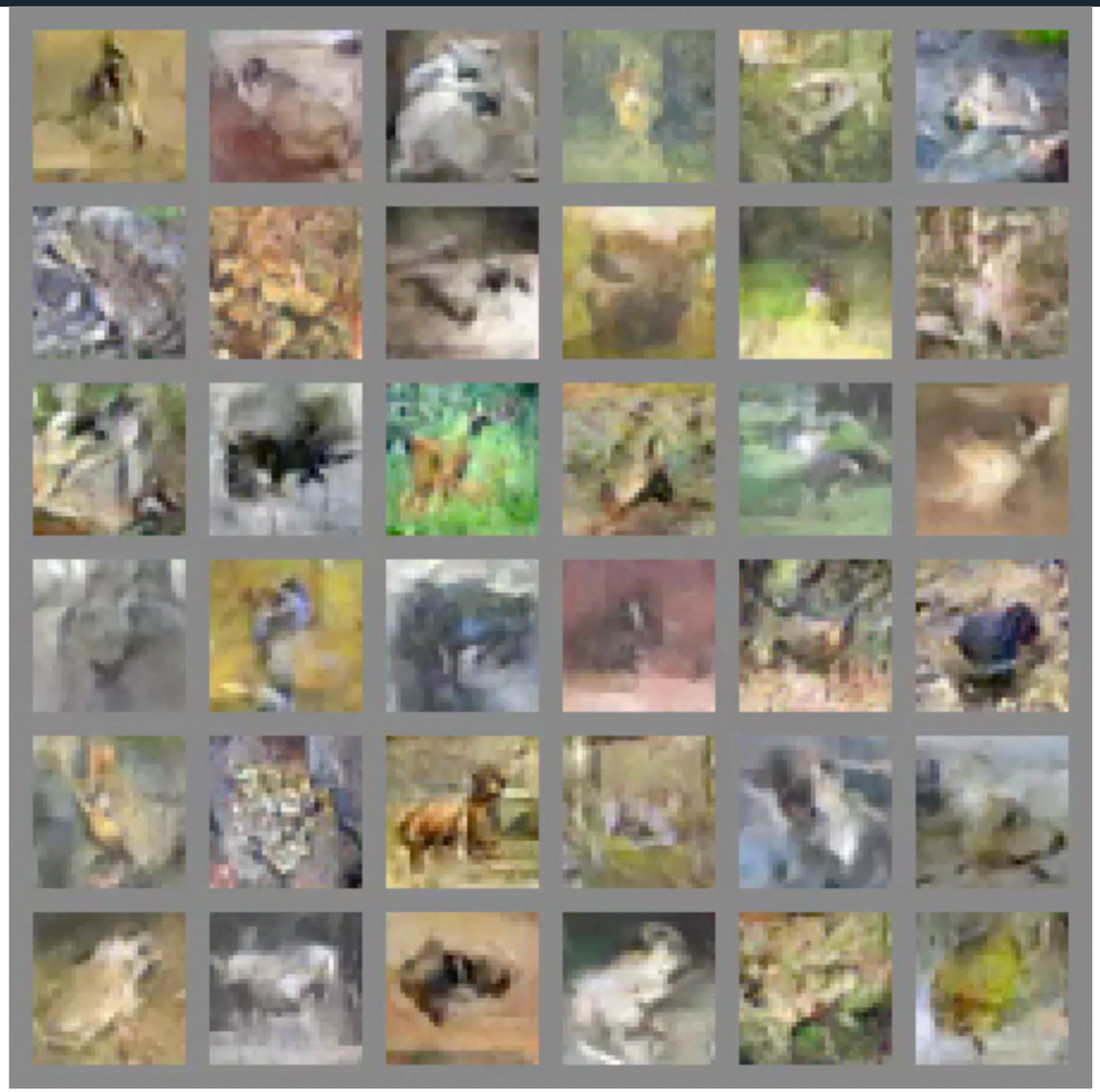
$$\tilde{\boldsymbol{x}} = G(\boldsymbol{z}), \quad \boldsymbol{z} \sim p(\boldsymbol{z})$$

  - the generator input $\boldsymbol{z}$ is sampled from some simple noise distribution, (e.g. uniform or Gaussian).

GAN samples



MNIST

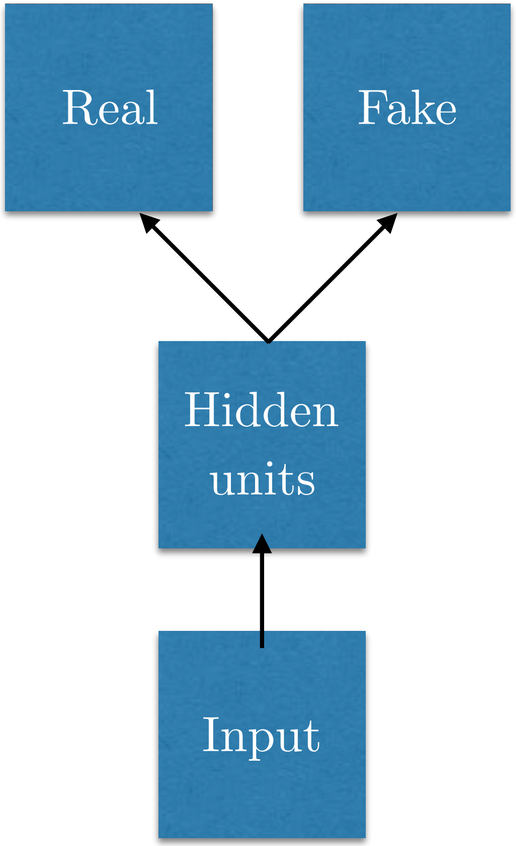


CIFAR-10

DCGAN samples (Radford, Metz and Chintala; 2016)

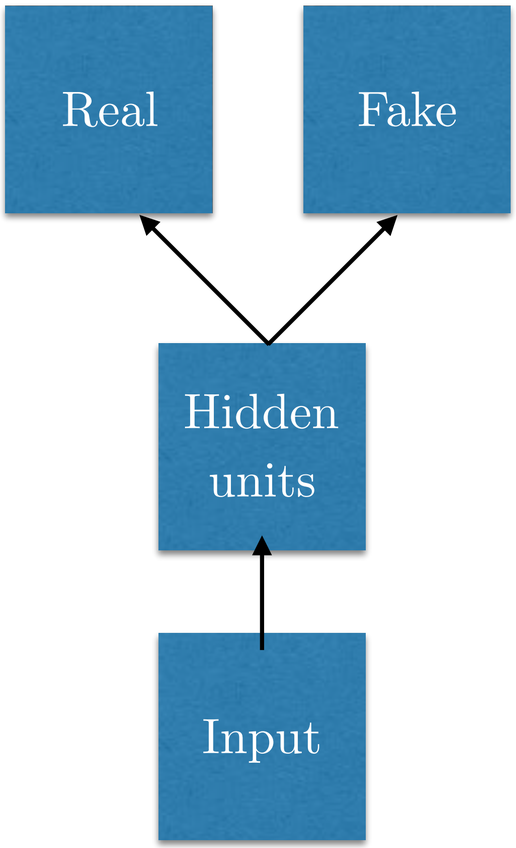
LSUN bedroom scenes

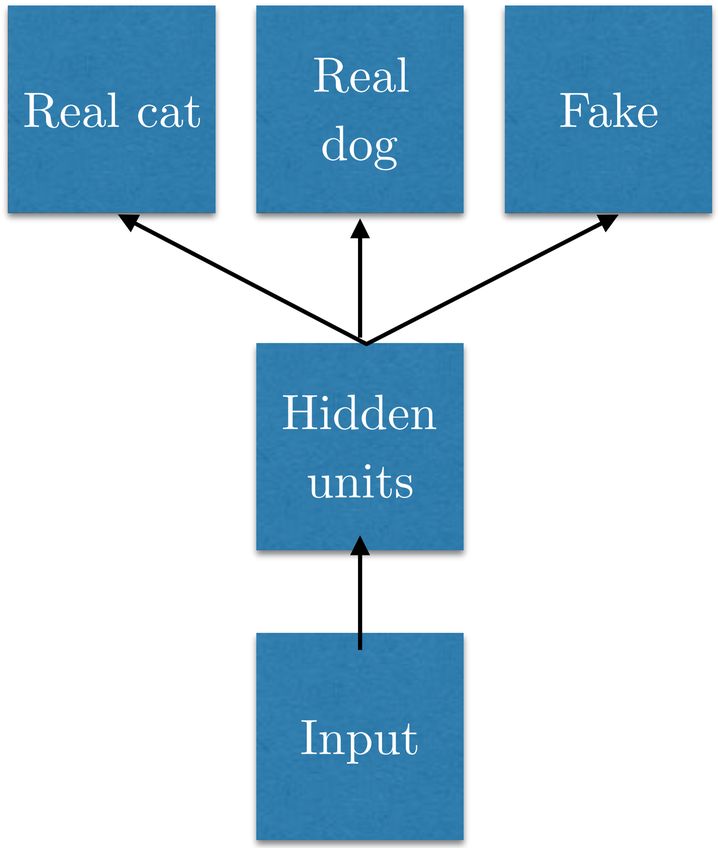
Supervised Discriminator



(Odena 2016, Salimans et al 2016)

Supervised Discriminator





(Odena 2016, Salimans et al 2016)

Semi-Supervised Classification

MILA

MNIST (Permutation Invariant)

Model	Number of incorrectly predicted test examples for a given number of labeled samples			
	20	50	100	200
DGN [21]			333 ± 14	
Virtual Adversarial [22]			212	
CatGAN [14]			191 ± 10	
Skip Deep Generative Model [23]			132 ± 7	
Ladder network [24]			106 ± 37	
Auxiliary Deep Generative Model [23]			96 ± 2	
Our model	1677 ± 452	221 ± 136	93 ± 6.5	90 ± 4.2
Ensemble of 10 of our models	1134 ± 445	142 ± 96	86 ± 5.6	81 ± 4.3

(Salimans et al 2016)

Semi-Supervised Classification

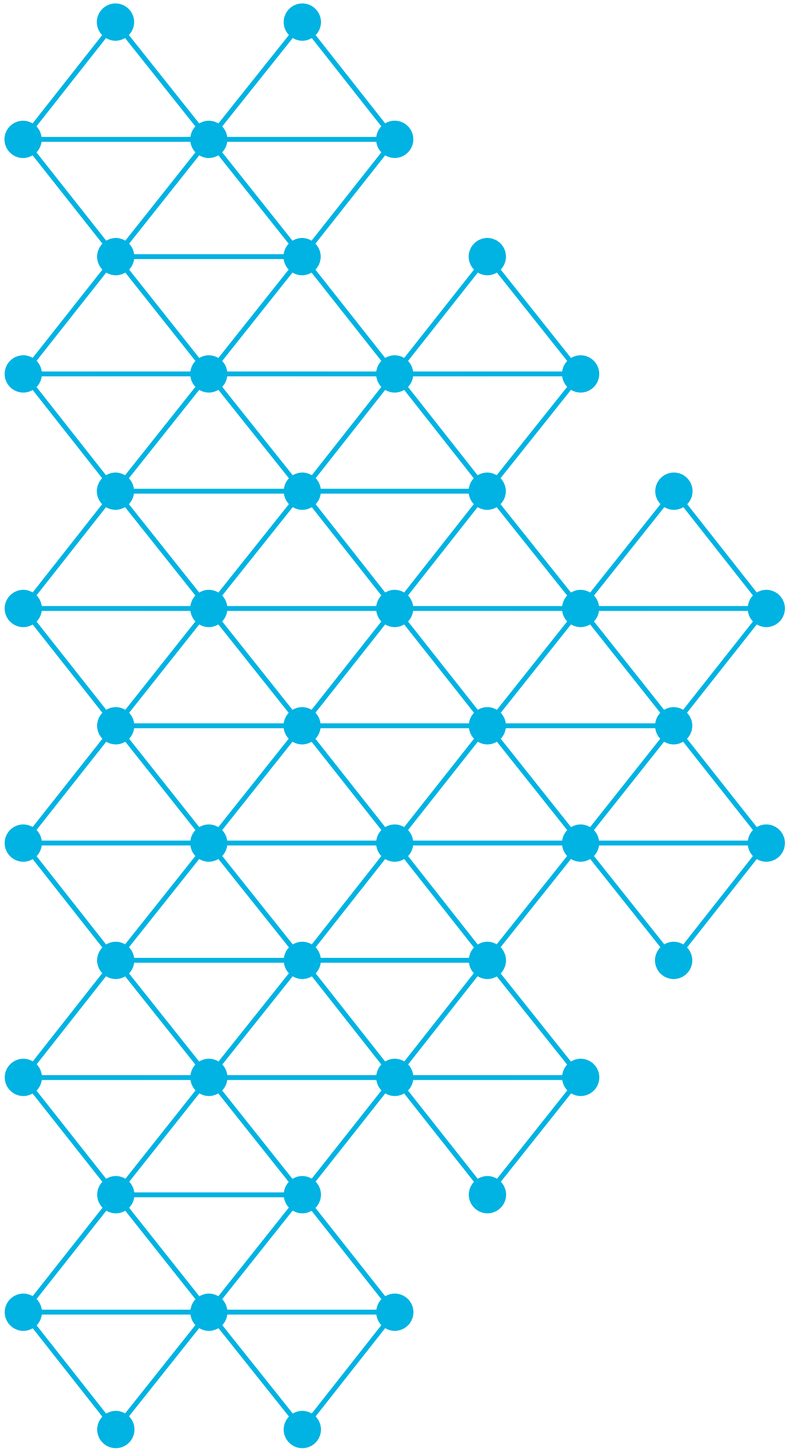
CIFAR-10

Model	Test error rate for a given number of labeled samples			
	1000	2000	4000	8000
Ladder network [24]			20.40 ± 0.47	
CatGAN [14]			19.58 ± 0.46	
Our model	21.83 ± 2.01	19.61 ± 2.09	18.63 ± 2.32	17.72 ± 1.82
Ensemble of 10 of our models	19.22 ± 0.54	17.25 ± 0.66	15.59 ± 0.47	14.87 ± 0.89

SVHN

Model	Percentage of incorrectly predicted test examples for a given number of labeled samples		
	500	1000	2000
DGN [21]		36.02 ± 0.10	
Virtual Adversarial [22]		24.63	
Auxiliary Deep Generative Model [23]		22.86	
Skip Deep Generative Model [23]		16.61 ± 0.24	
Our model	18.44 ± 4.8	8.11 ± 1.3	6.16 ± 0.58
Ensemble of 10 of our models		5.88 ± 1.0	

(Salimans et al 2016)

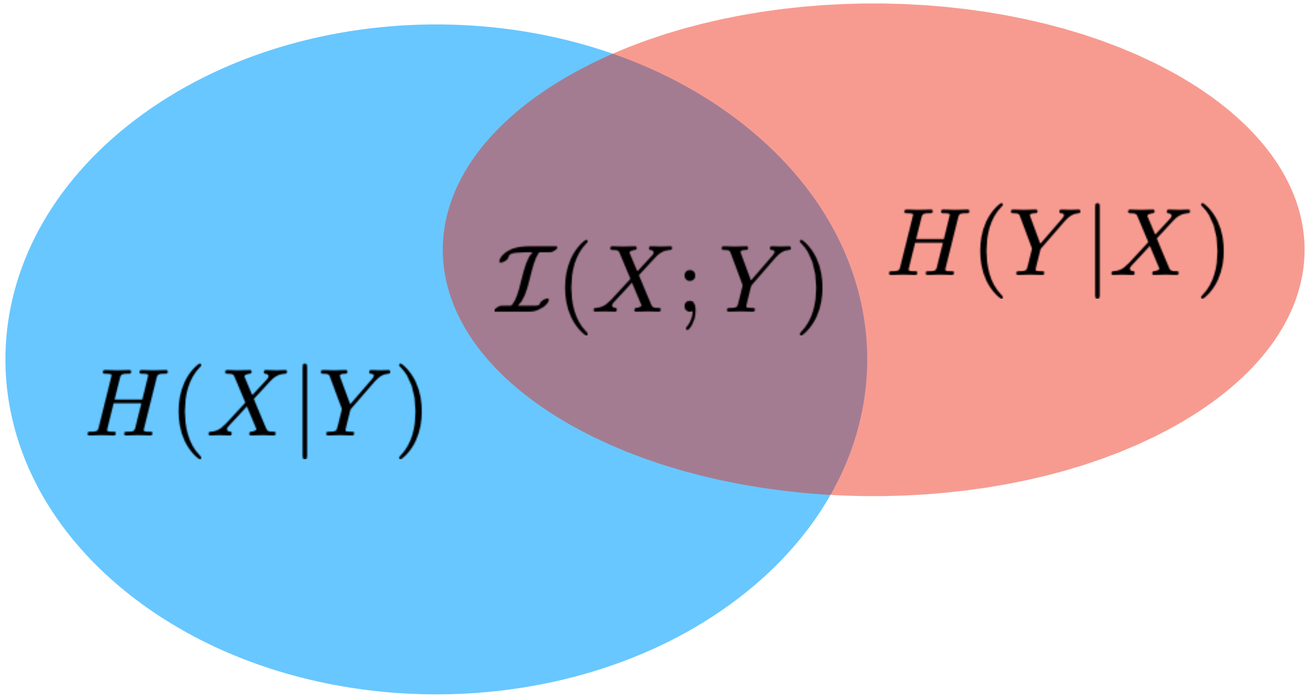


Contrastive Self-Supervised Learning Methods

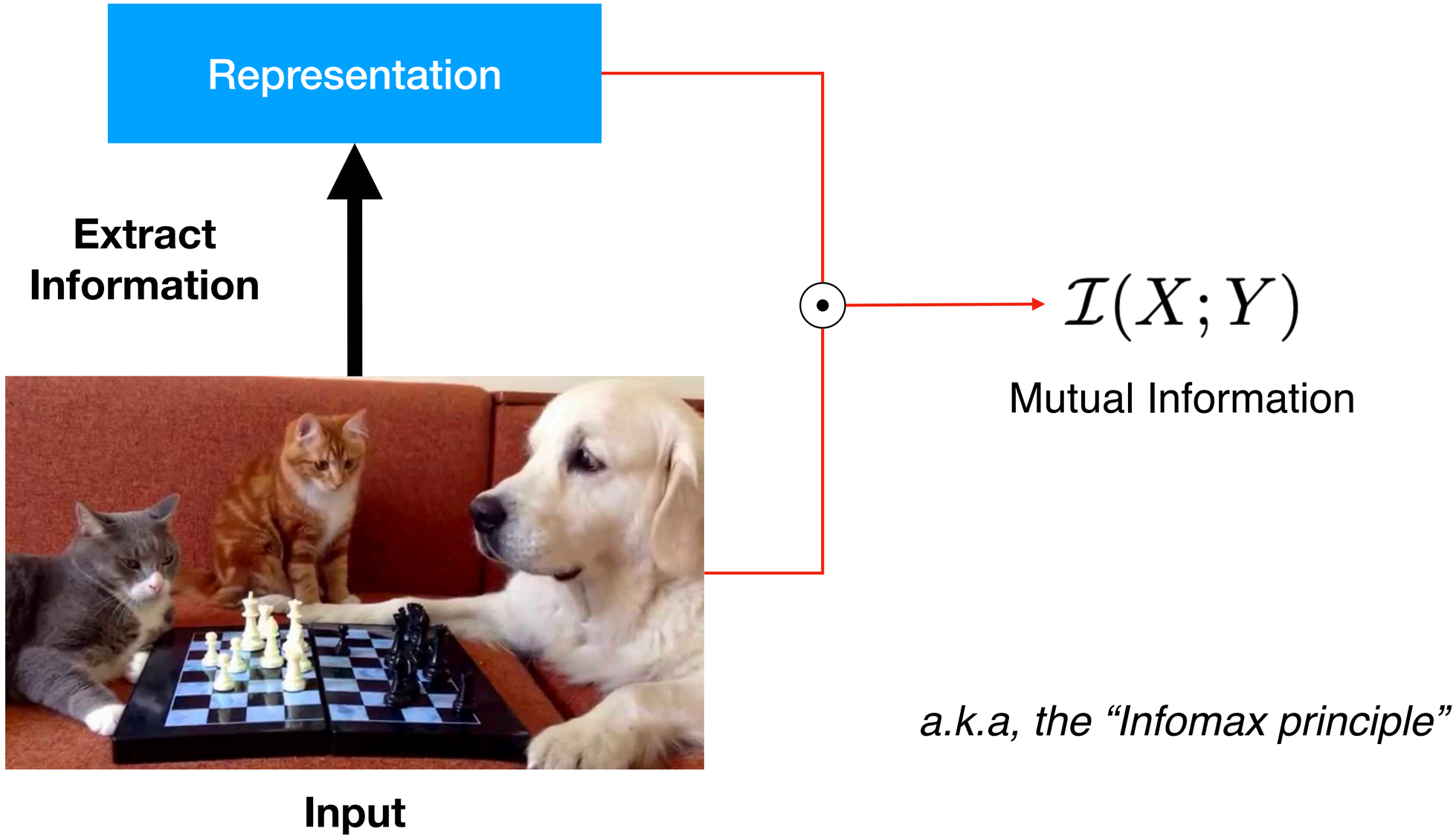
Mutual information

$$\mathcal{I}(X;Y) := \mathcal{D}_{KL}(\mathbb{P}_{X,Y} || \mathbb{P}_X \otimes \mathbb{P}_Y)$$

(Joint distribution II Product of marginals)

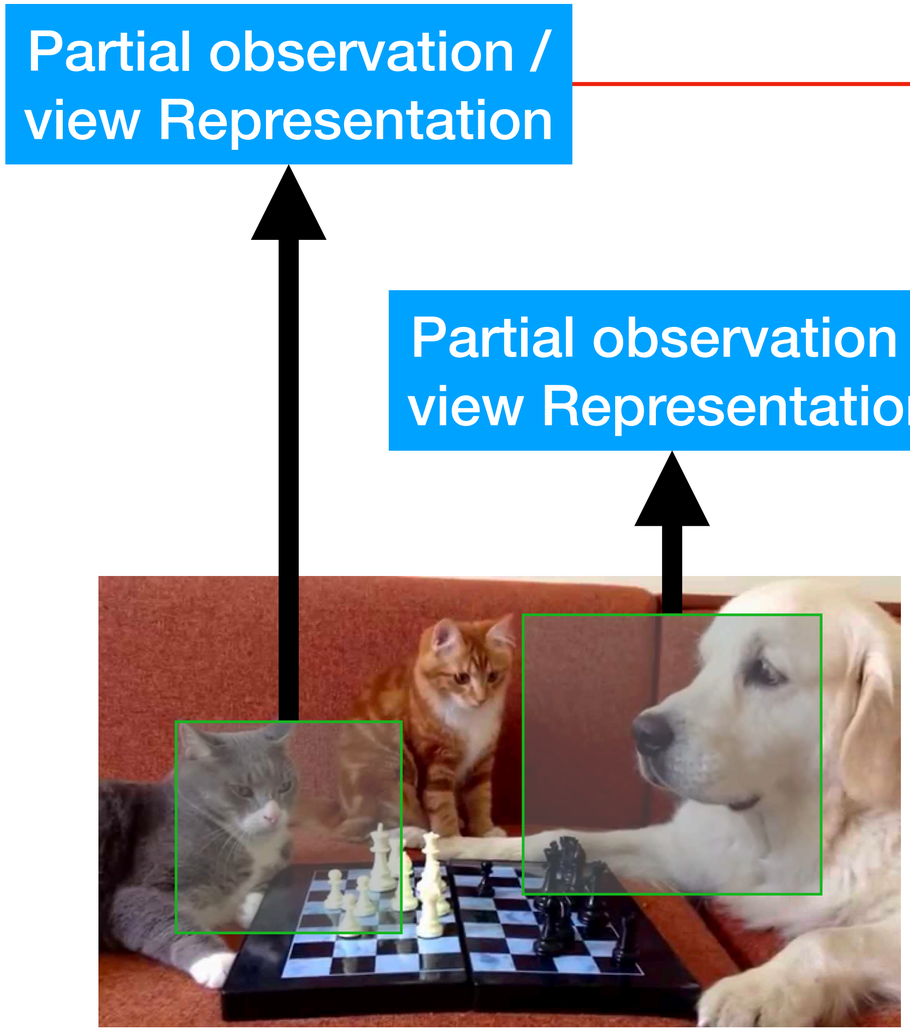


Mutual Information for Representation Learning



a.k.a, the "Infomax principle"

MI for Self-Supervised Representation Learning



$\mathcal{I}(X;Y)$

Mutual Information

Ask questions about the data at the representation / feature level

Mutual Information Neural Estimation (MINE)

But, mutual information is difficult to compute, no?

Mutual information estimator $\widehat{\mathcal{I}}_\omega(X;Y)$

Neural network discriminator $T_\omega : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

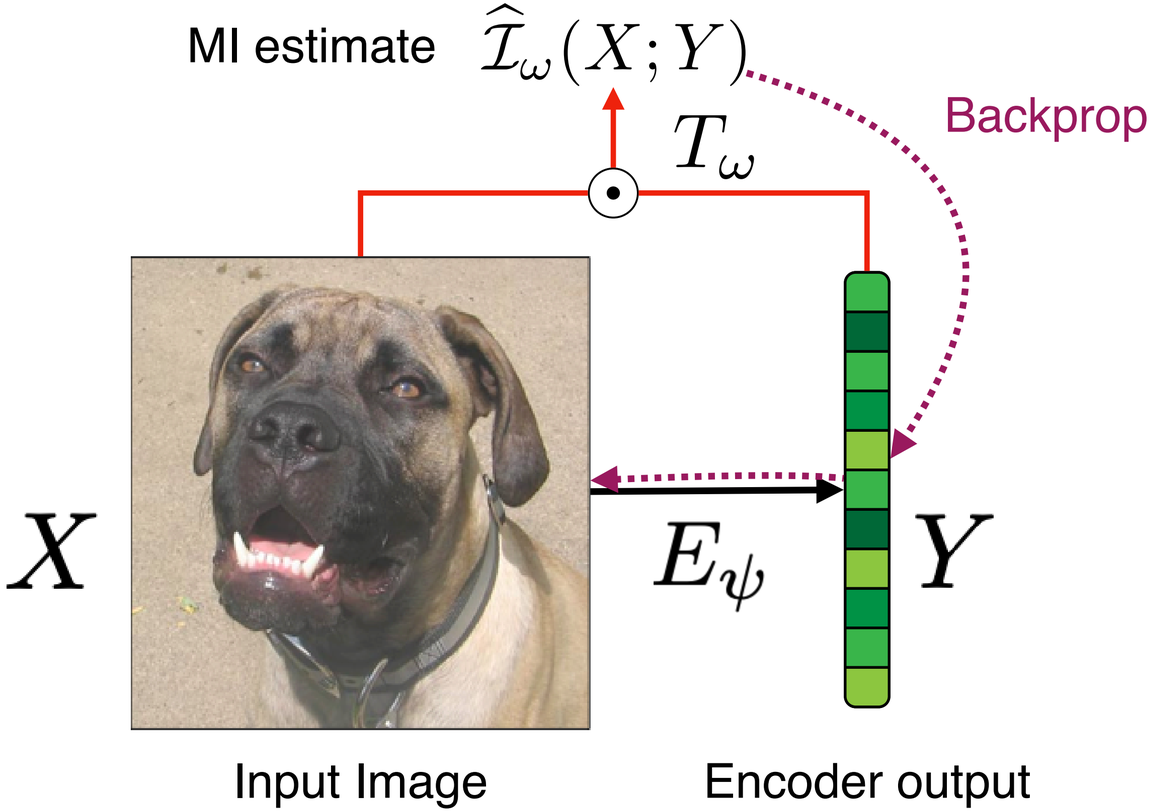
Lower bound to mutual information (Donsker-Varadhan representation)

$$\mathcal{D}_{KL}(\mathbb{P}_{X,Y}||\mathbb{P}_X \otimes \mathbb{P}_Y) \geq \widehat{\mathcal{I}}_\omega(X;Y) := \mathbb{E}_{\mathbb{P}_{X,Y}}[T_\omega] - \log \mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Y}[e^{T_\omega}]$$

In order to estimate MI, we only need to train a neural network to maximize this.

Belghazi et. al., ICML 2018

Maximizing mutual information in encoders

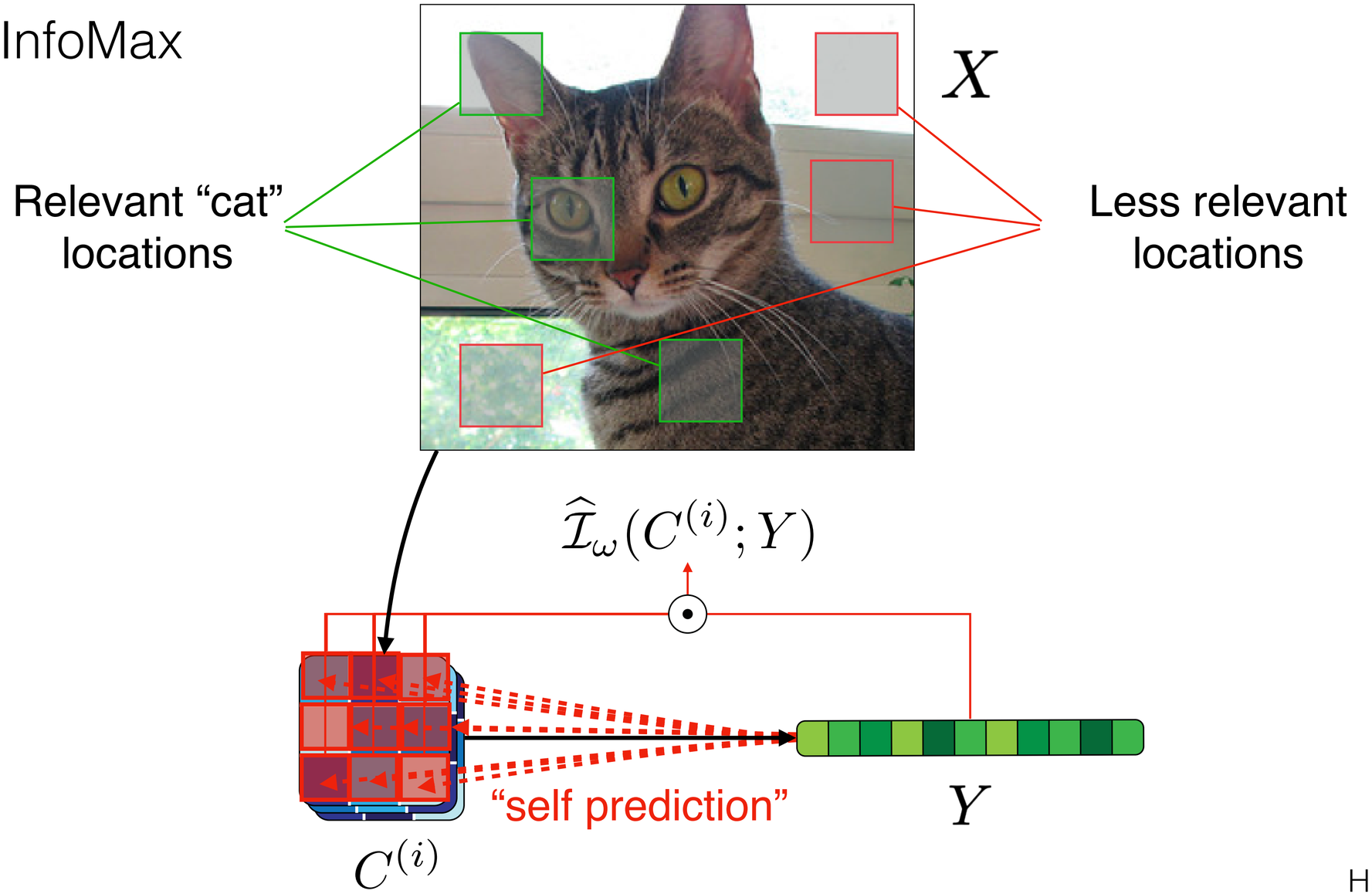


- Network encodes the input
- The discriminator estimates mutual information (batch-wise)
- Estimate is used to maximize the mutual information between encoder input and output

Hjelm et. al., ICLR 2019

DIM: Using local structure is crucial

DIM: Deep InfoMax



Hjelm et. al., ICLR 2019

DIM: 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

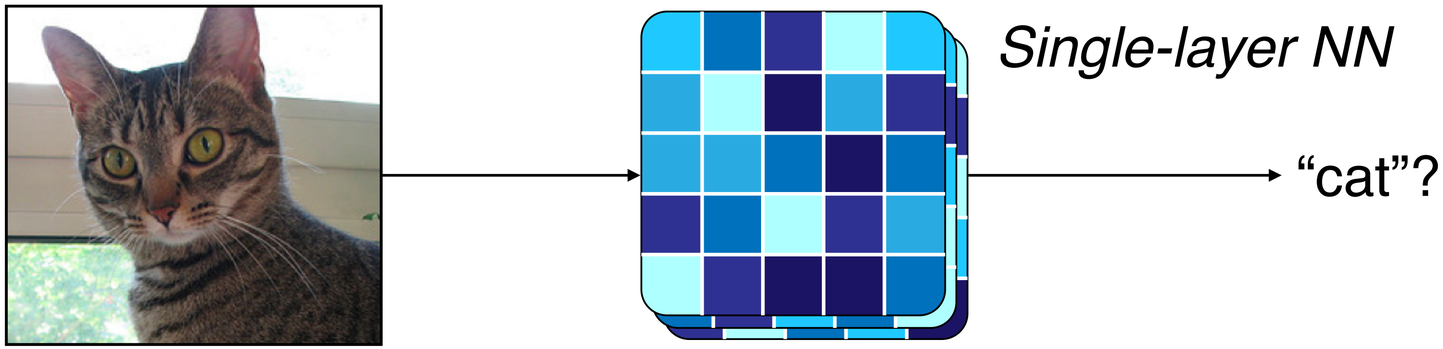
Global Feature Vector

Cat	Striped	Fur	Ear	White	Window

Order of importance

Hjelm et. al., ICLR 2019

DIM: 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
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**

Hjelm et. al., ICLR 2019

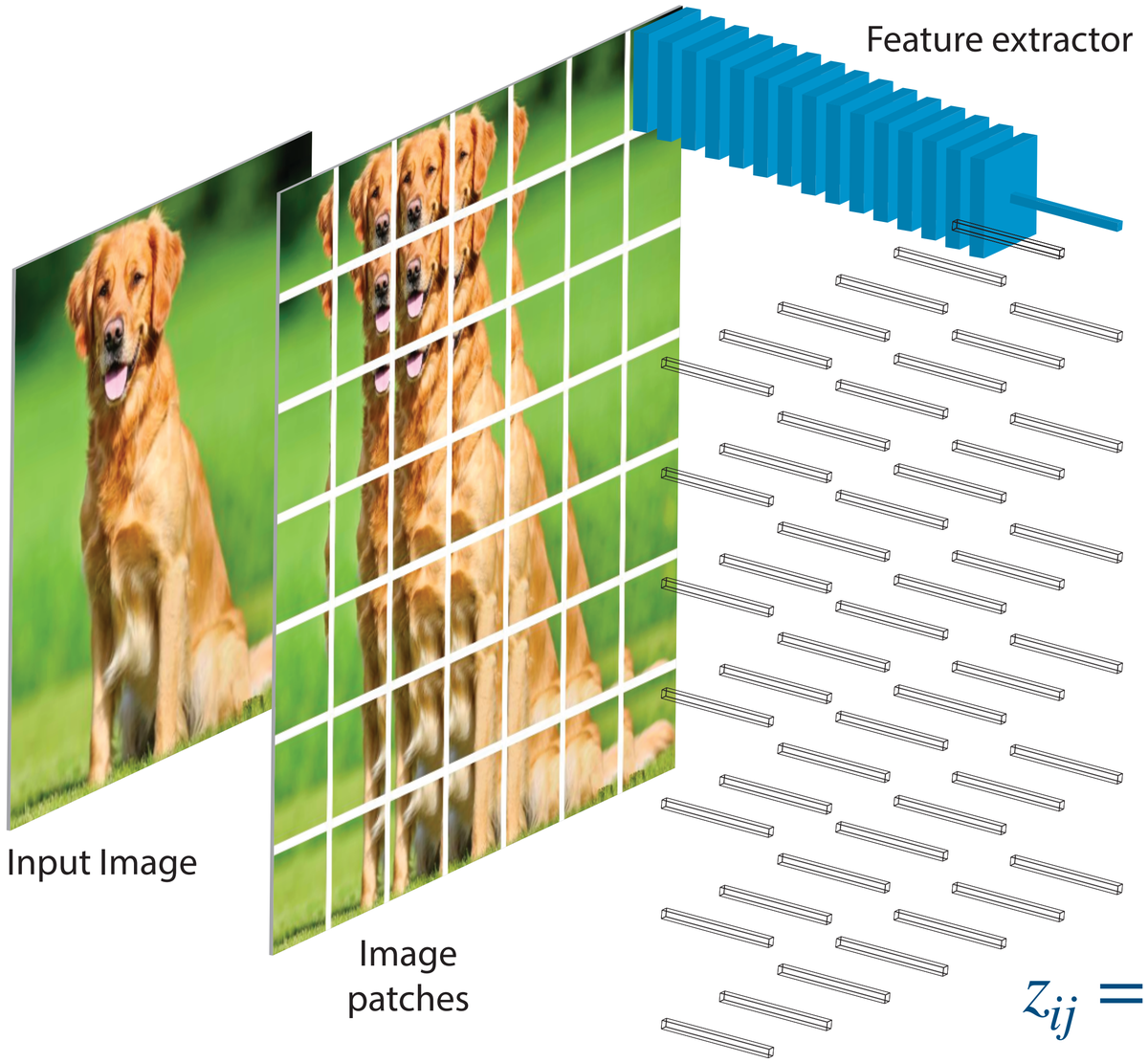
Contrastive predictive coding (CPC)

van den Oord, Li, and Vinyals. Representation learning with contrastive predictive coding. arXiv, 2018

- Predict encodings of related patches (predictive coding)
- To avoid trivial encodings, score negative encodings lower (contrastive)
- Negative patches sampled from different images (easy negatives) and different locations from the same image (hard negatives)

Contrastive predictive coding (CPC)

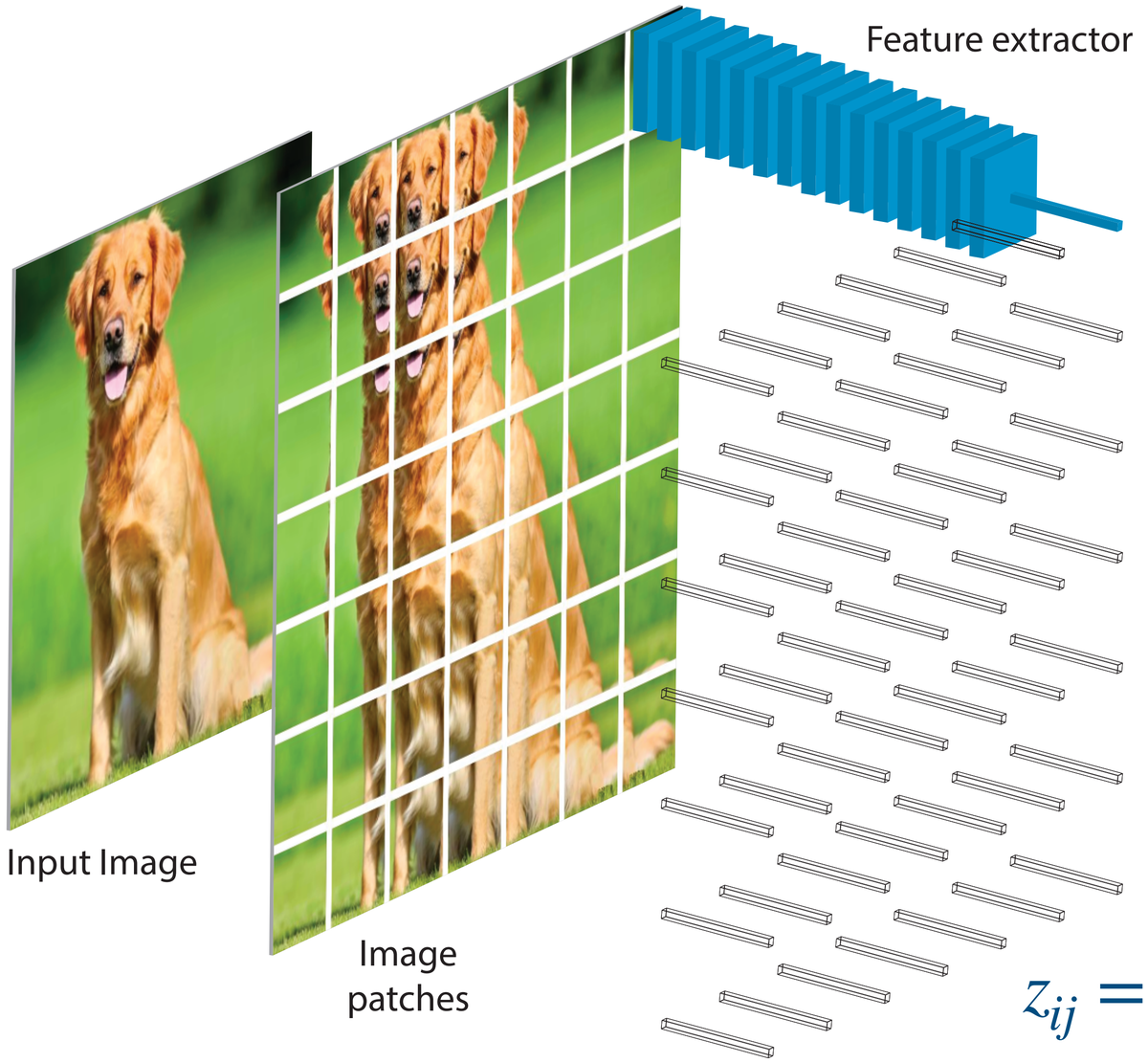
van den Oord, Li, and Vinyals. Representation learning with contrastive predictive coding. arXiv, 2018



$$z_{ij} = g_{enc}(x_{ij})$$

Contrastive predictive coding (CPC)

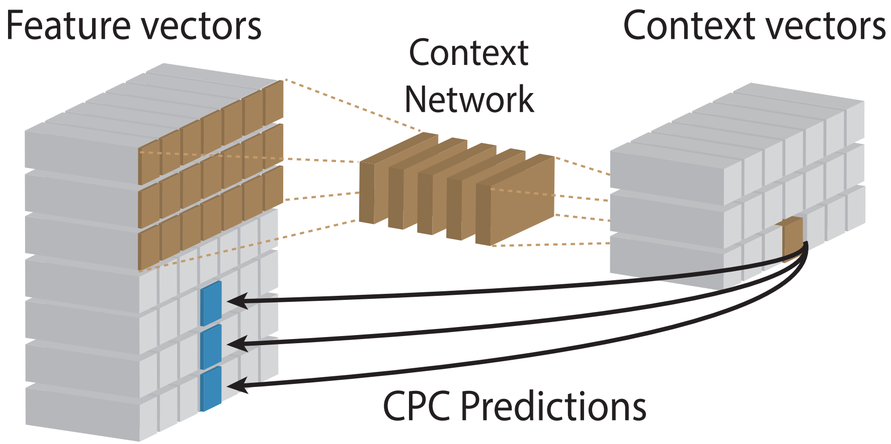
van den Oord, Li, and Vinyals. Representation learning with contrastive predictive coding. arXiv, 2018



$$z_{ij} = g_{enc}(x_{ij})$$

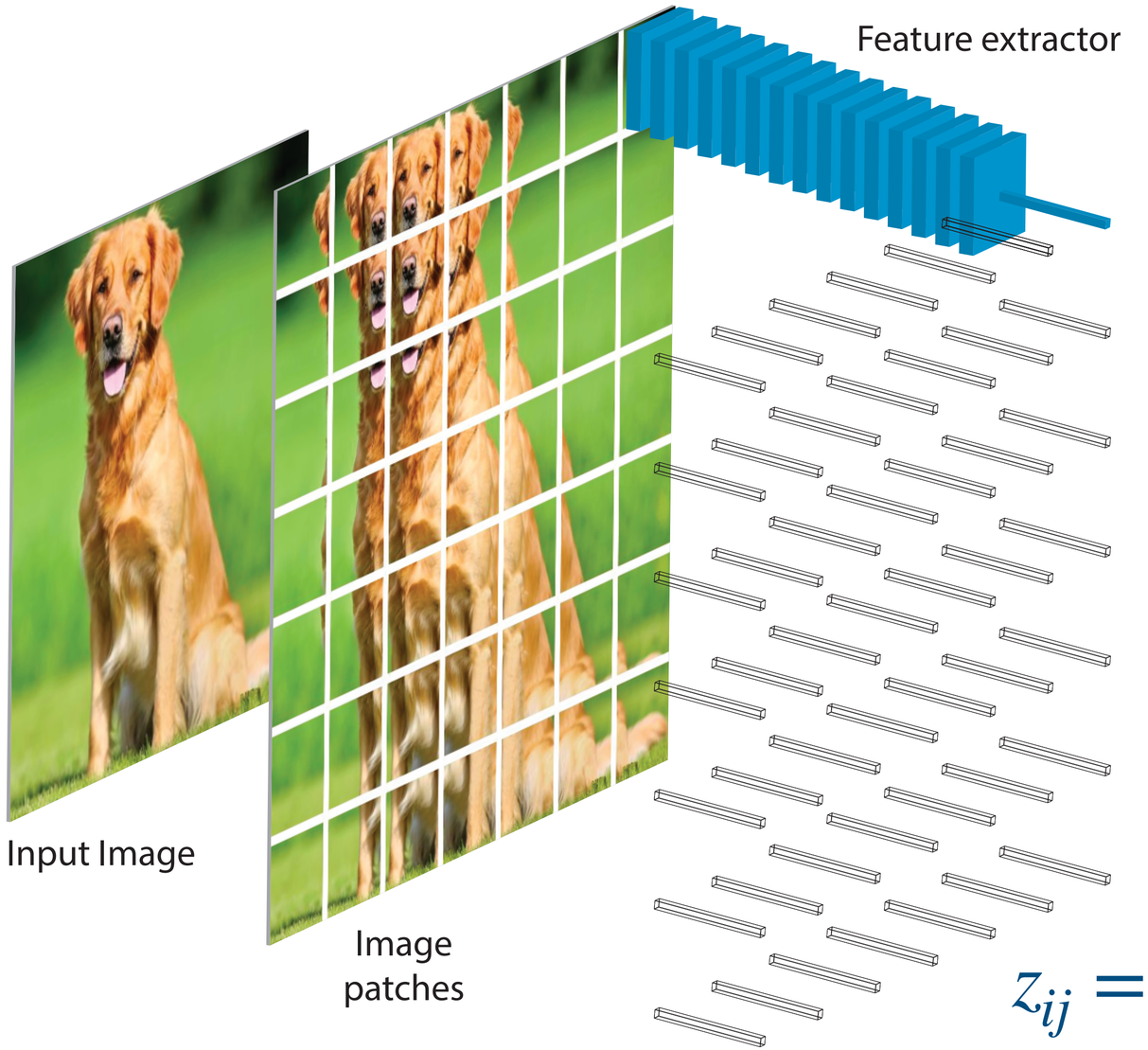
Merge

$$c_{ij} = g_{context}(z_{ij})$$

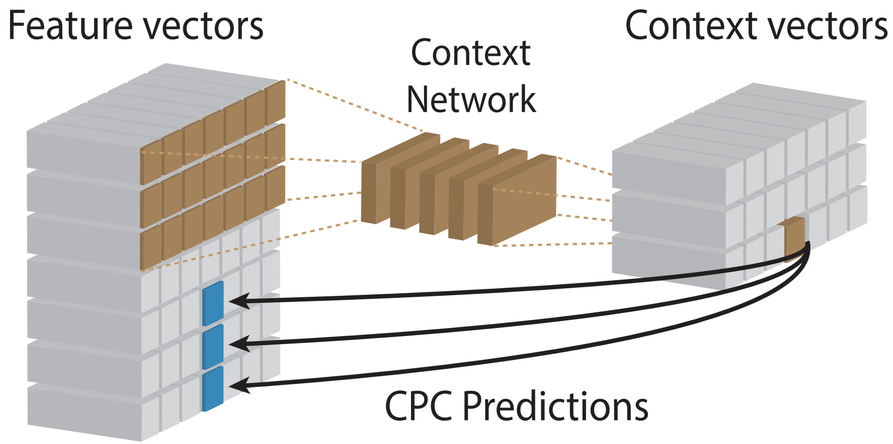


Contrastive predictive coding (CPC)

van den Oord, Li, and Vinyals. Representation learning with contrastive predictive coding. arXiv, 2018



Merge



$$c_{ij} = g_{context}(z_{ij})$$

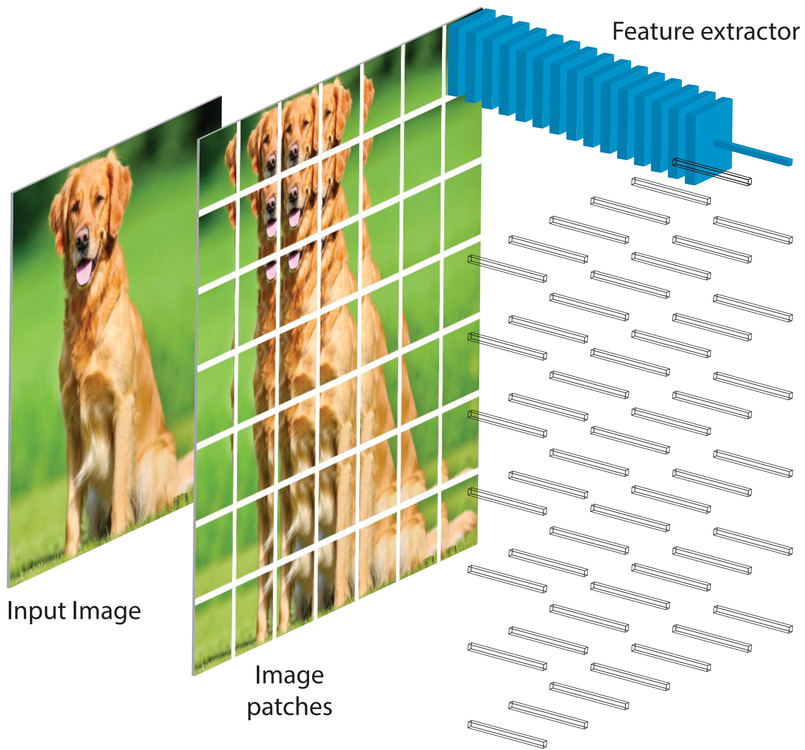
$$z_{ij} = g_{enc}(x_{ij}) \longleftrightarrow \hat{z}_{i+k,j} = W_k^\top c_{ij}$$

Contrastive predictive coding (CPC)

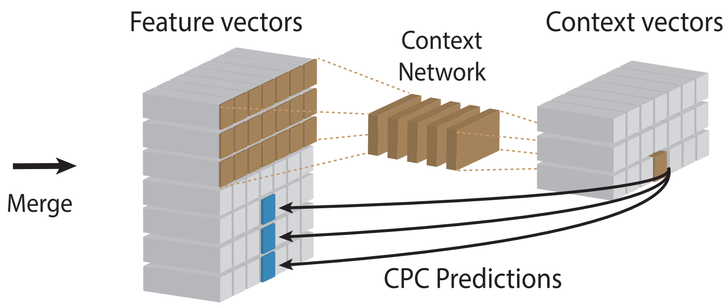
van den Oord, Li, and Vinyals. Representation learning with contrastive predictive coding. arXiv, 2018

Train a classifier to correctly classify the positive sample:

$$\mathscr{L} = -\sum_{ijk} \log \frac{\exp(\hat{z}_{i+k,j}^{\top} z_{i+k,j})}{\exp(\hat{z}_{i+k,j}^{\top} z_{i+k,j}) + \sum_{l} \exp(\hat{z}_{i+k,j}^{\top} z_{l})}$$



$$c_{ij} = g_{context}(z_{ij})$$



$$\hat{z}_{i+k,j} = W_k^{\top} c_{ij}$$

$$z_{ij} = g_{enc}(x_{ij})$$

Contrastive predictive coding (CPC)

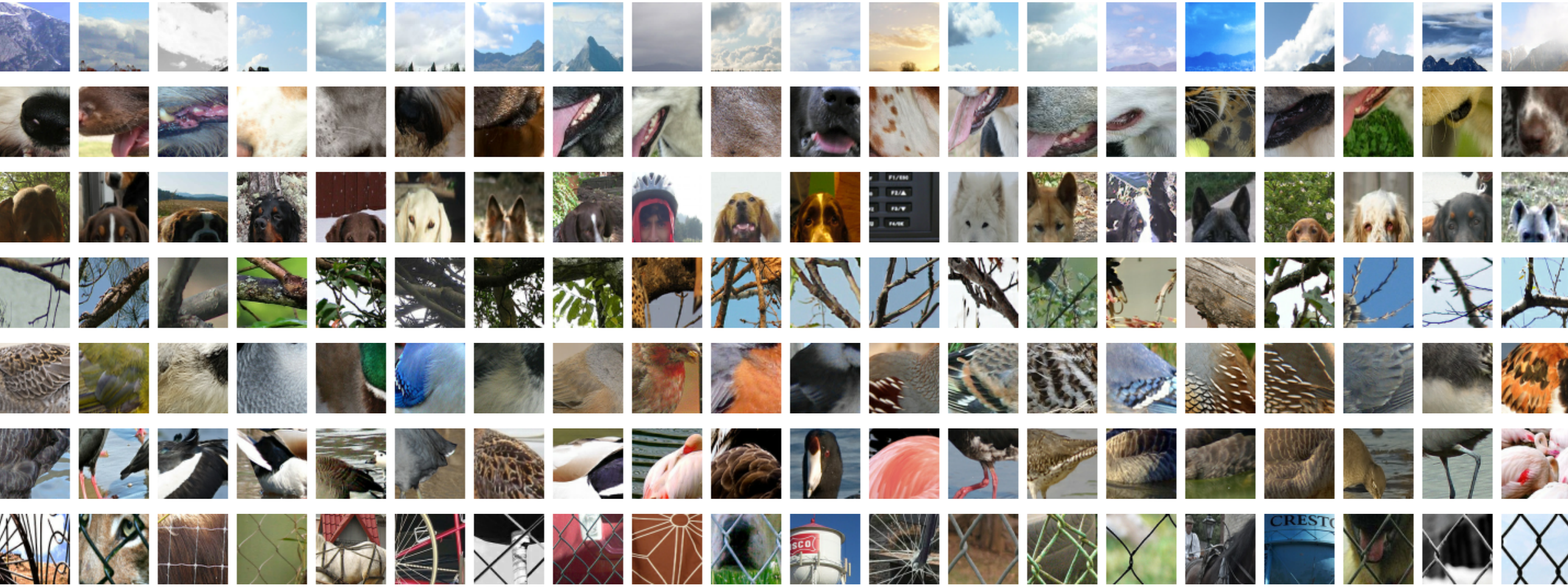
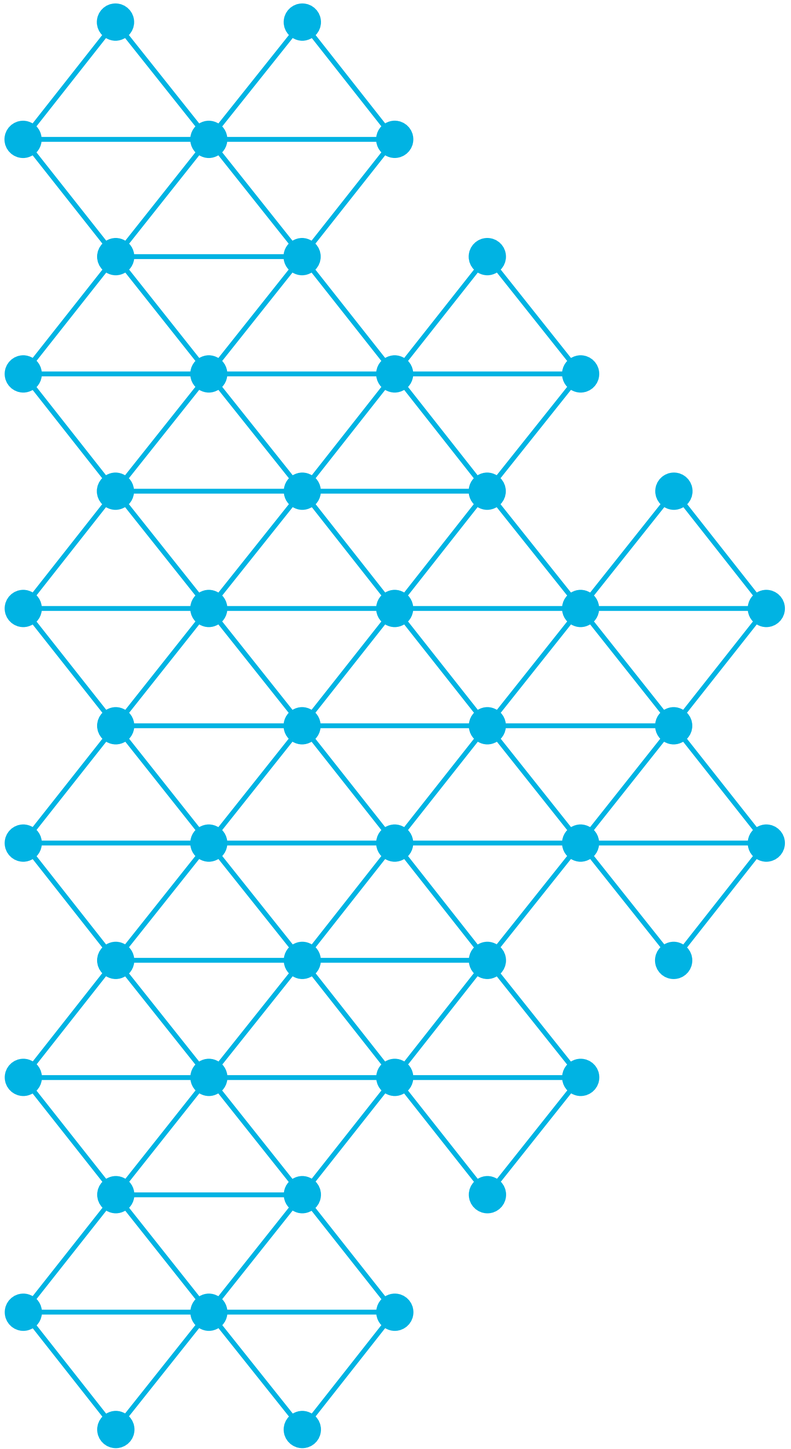


Figure 5: Every row shows image patches that activate a certain neuron in the CPC architecture.

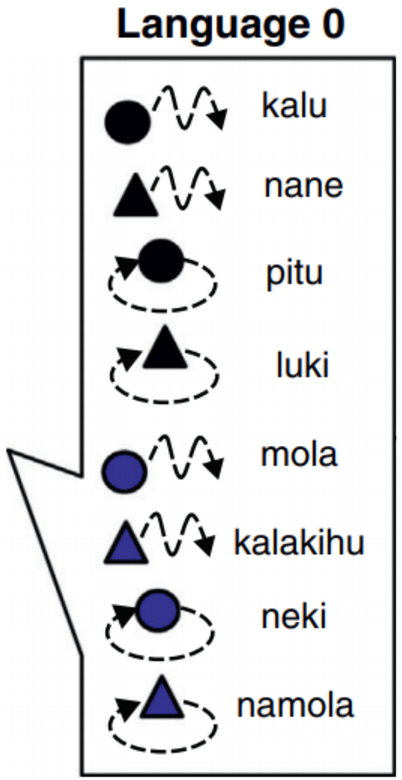
Image credit: van den Oord, Li, and Vinyals. Representation learning with contrastive predictive coding. arXiv, 2018



Iterated Learning

Iterated Learning: human language emergence

Iterated Learning: the compositionally structure of human language emerged through the successive re-learning of language from generation to generation.

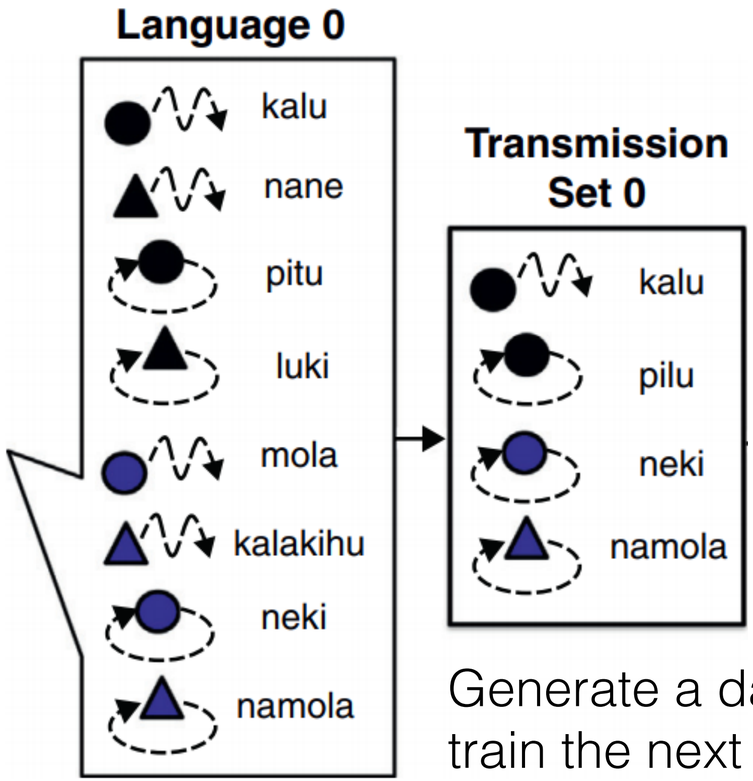


Start with a set of shape/actions

Simon Kirby, Tom Griffiths, and Kenny Smith. “Iterated learning and the evolution of language.” In: *Current opinion in neurobiology* 28 (2014),

Iterated Learning: human language emergence

Iterated Learning: the compositionally structure of human language emerged through the successive re-learning of language from generation to generation.

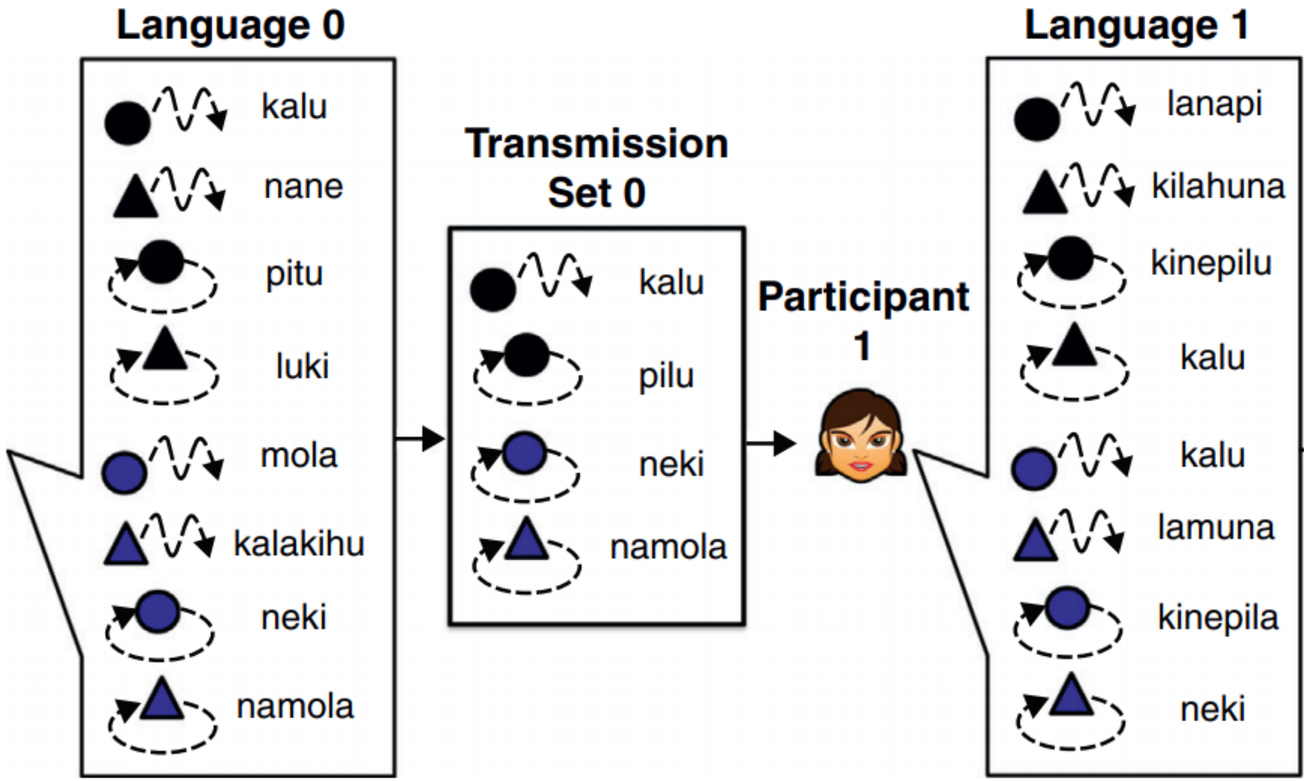


Generate a dataset to train the next generation

Simon Kirby, Tom Griffiths, and Kenny Smith. “Iterated learning and the evolution of language.” In: *Current opinion in neurobiology* 28 (2014),

Iterated Learning: human language emergence

Iterated Learning: the compositionally structure of human language emerged through the successive re-learning of language from generation to generation.

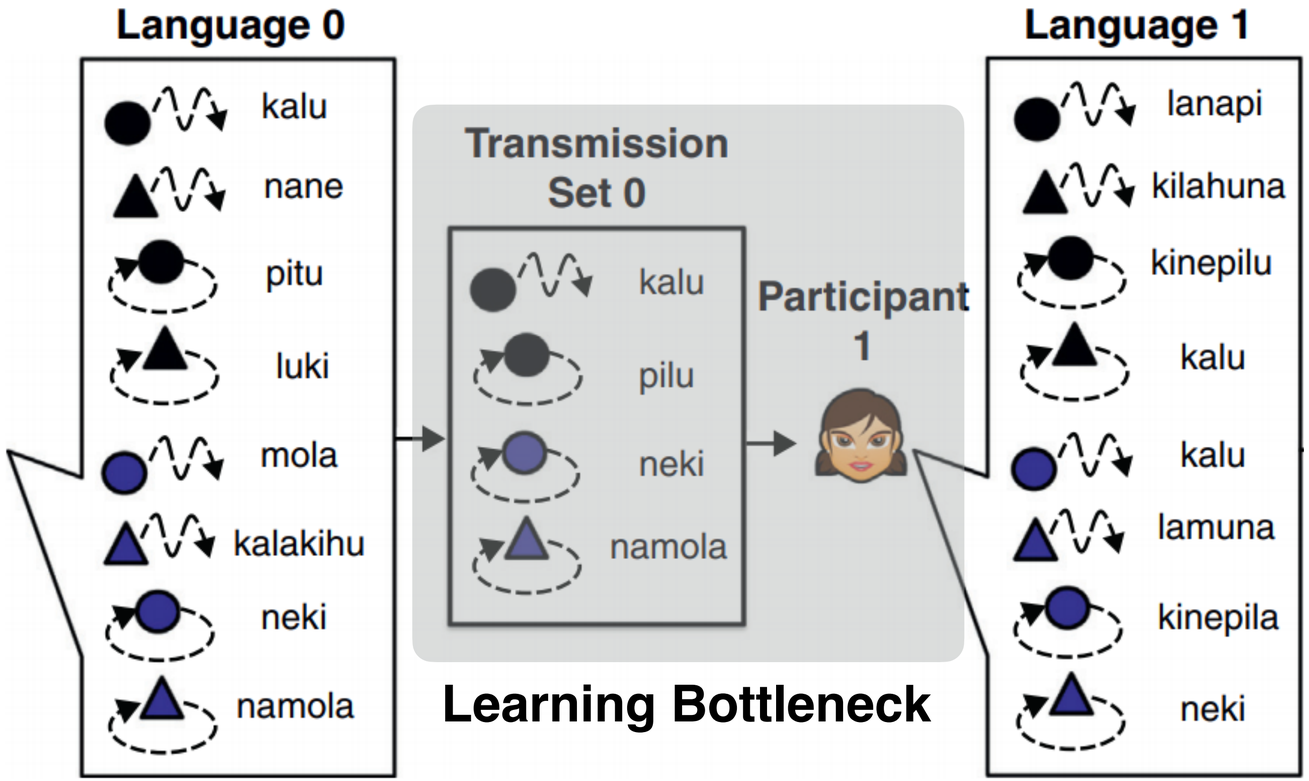


Training the next generation from the limited data

Simon Kirby, Tom Griffiths, and Kenny Smith. “Iterated learning and the evolution of language.” In: *Current opinion in neurobiology* 28 (2014),

Iterated Learning: human language emergence

Iterated Learning: the compositionally structure of human language emerged through the successive re-learning of language from generation to generation.

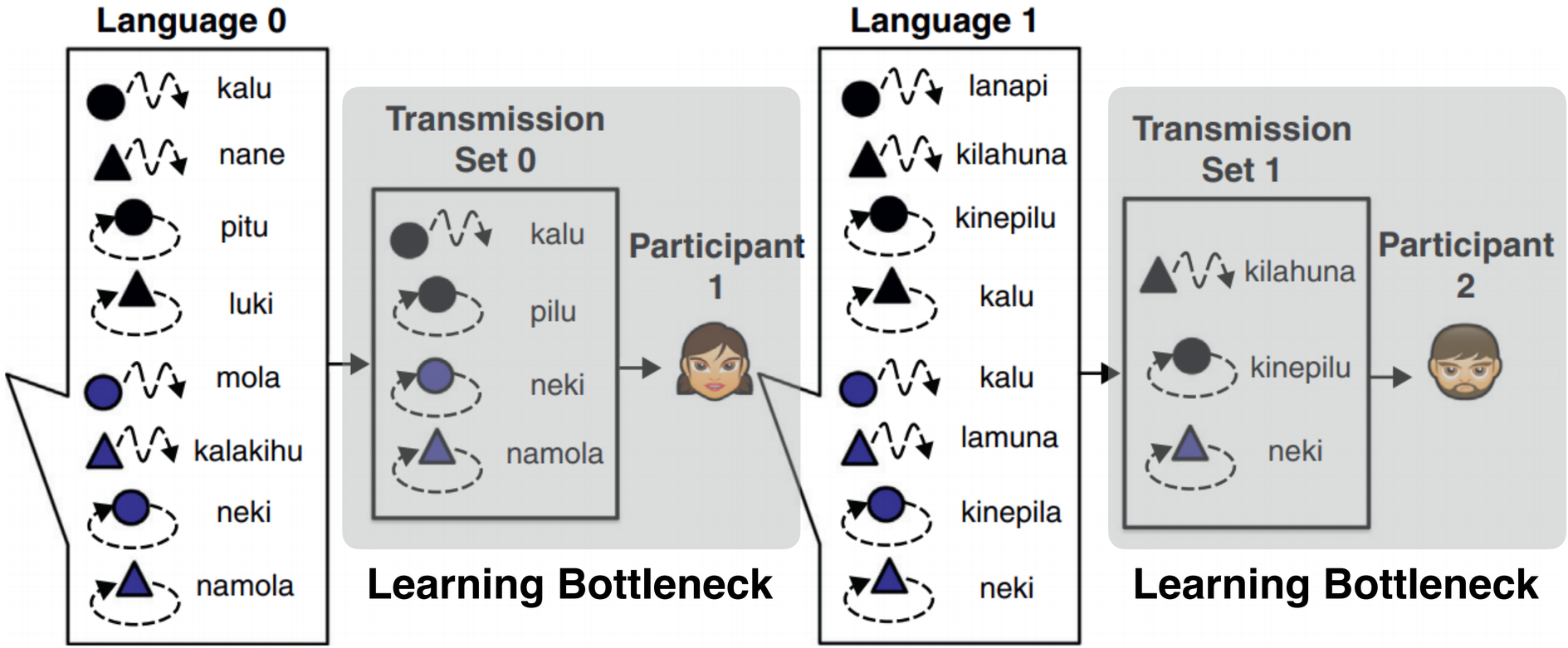


Training the next generation from the limited data

Simon Kirby, Tom Griffiths, and Kenny Smith. "Iterated learning and the evolution of language." In: *Current opinion in neurobiology* 28 (2014),

Iterated Learning: human language emergence

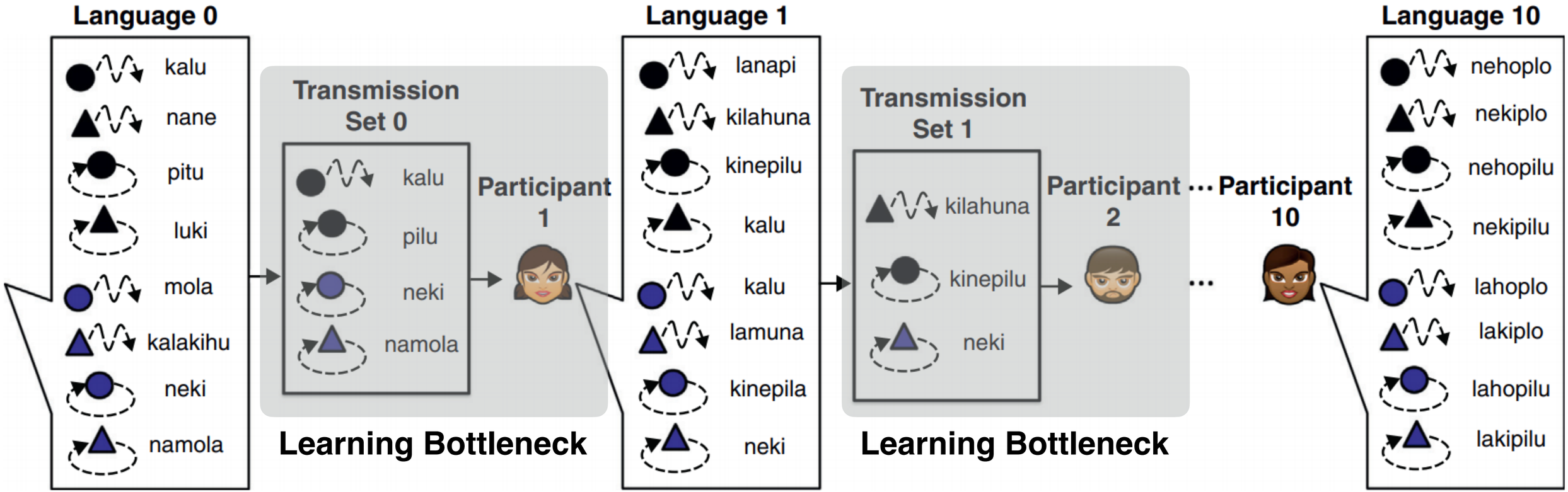
Iterated Learning: the compositionally structure of human language emerged through the successive re-learning of language from generation to generation.



Simon Kirby, Tom Griffiths, and Kenny Smith. "Iterated learning and the evolution of language." In: *Current opinion in neurobiology* 28 (2014),

Iterated Learning: human language emergence

Iterated Learning: the compositionally structure of human language emerged through the successive re-learning of language from generation to generation.



Simon Kirby, Tom Griffiths, and Kenny Smith. "Iterated learning and the evolution of language." In: *Current opinion in neurobiology* 28 (2014),

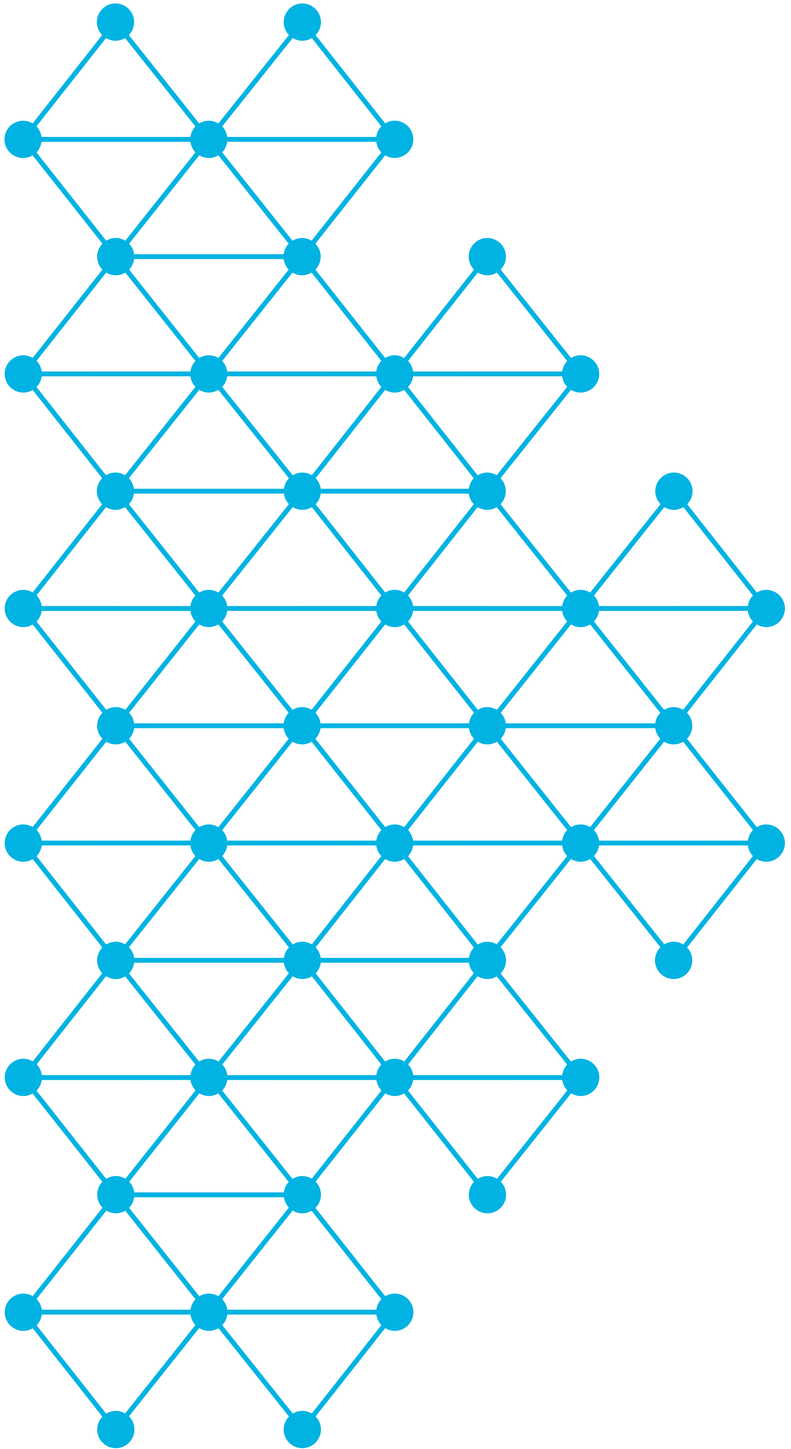
Iterated Learning applications to AI

Questions:

- Can we exploit this mechanism to product representations that are compositionally structured in our machine learning methods?
- If so, will this lead to improved systematic generalization?
- Is this working on the same or similar mechanism to self-training
  (as in Xie et al (2020) Self-training with Noisy Student improves ImageNet classification.)

Iterated Learning applications to AI

- Can we exploit this mechanism to product representations that are compositionally structured in our machine learning methods?
  - Help with the emergence of compositional language in AI agents.
    - Ren et al. (2020) Compositional languages emerge in a neural iterated learning model. ICLR 2020
    - Guo et al. (2020) The emergence of compositional languages for numeric concepts through iterated learning in neural agents.
    - Cogswell et al. (2020) Emergence of Compositional Language with Deep Generational Transmission
  - Also helps avoid language drift in dialogue agent self-play.
    - Lu et al (2020) Countering Language Drift with Seeded Iterated Learning
  - Early evidence on the promotion of systematic generalization.



Self-Training

Self-Training with Noisy Students

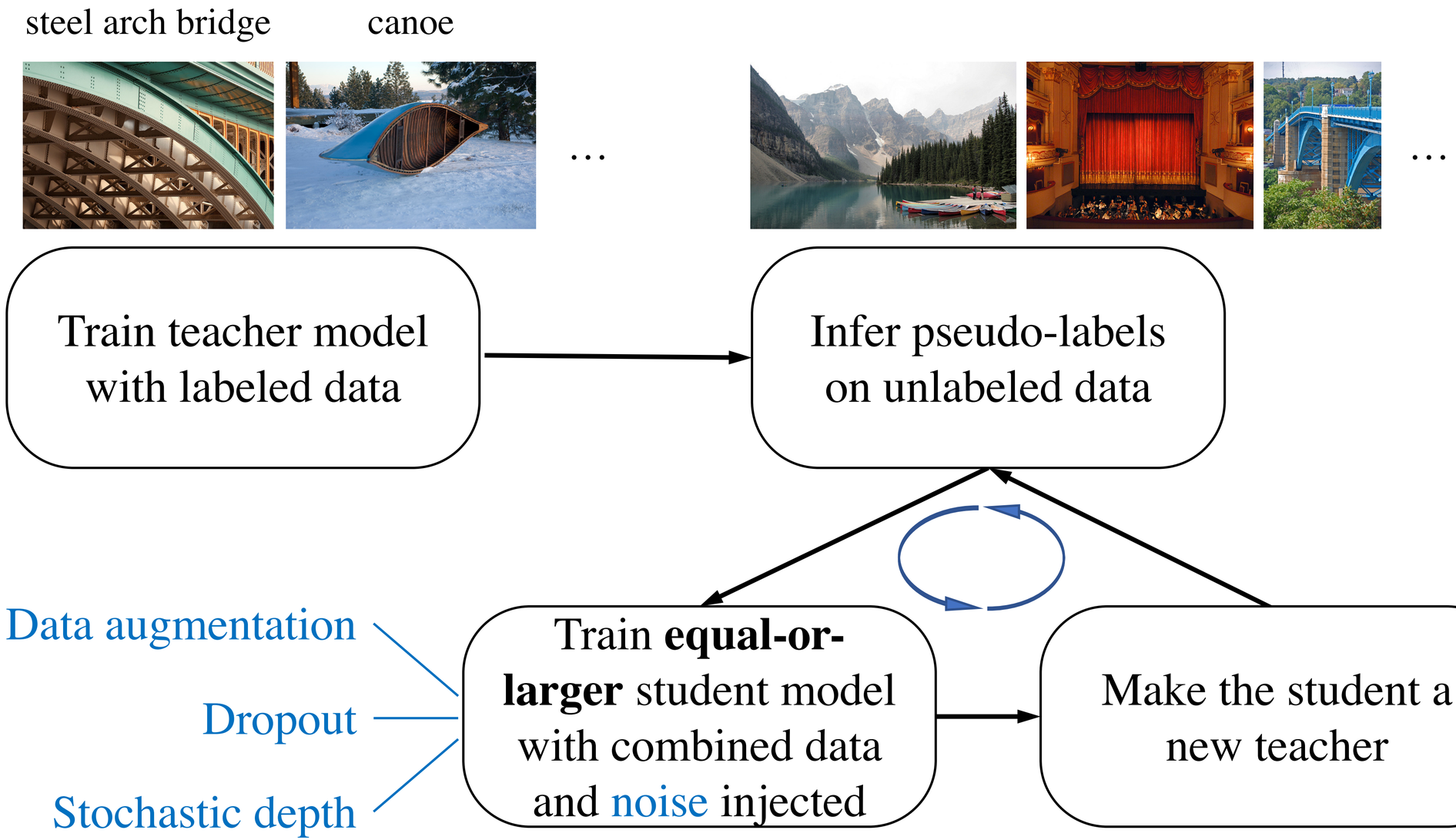
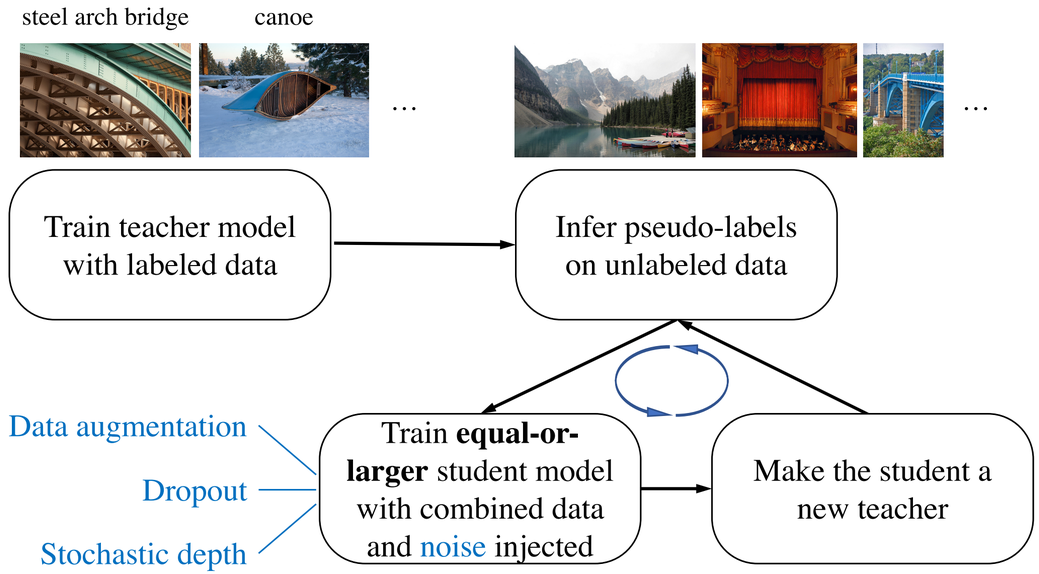


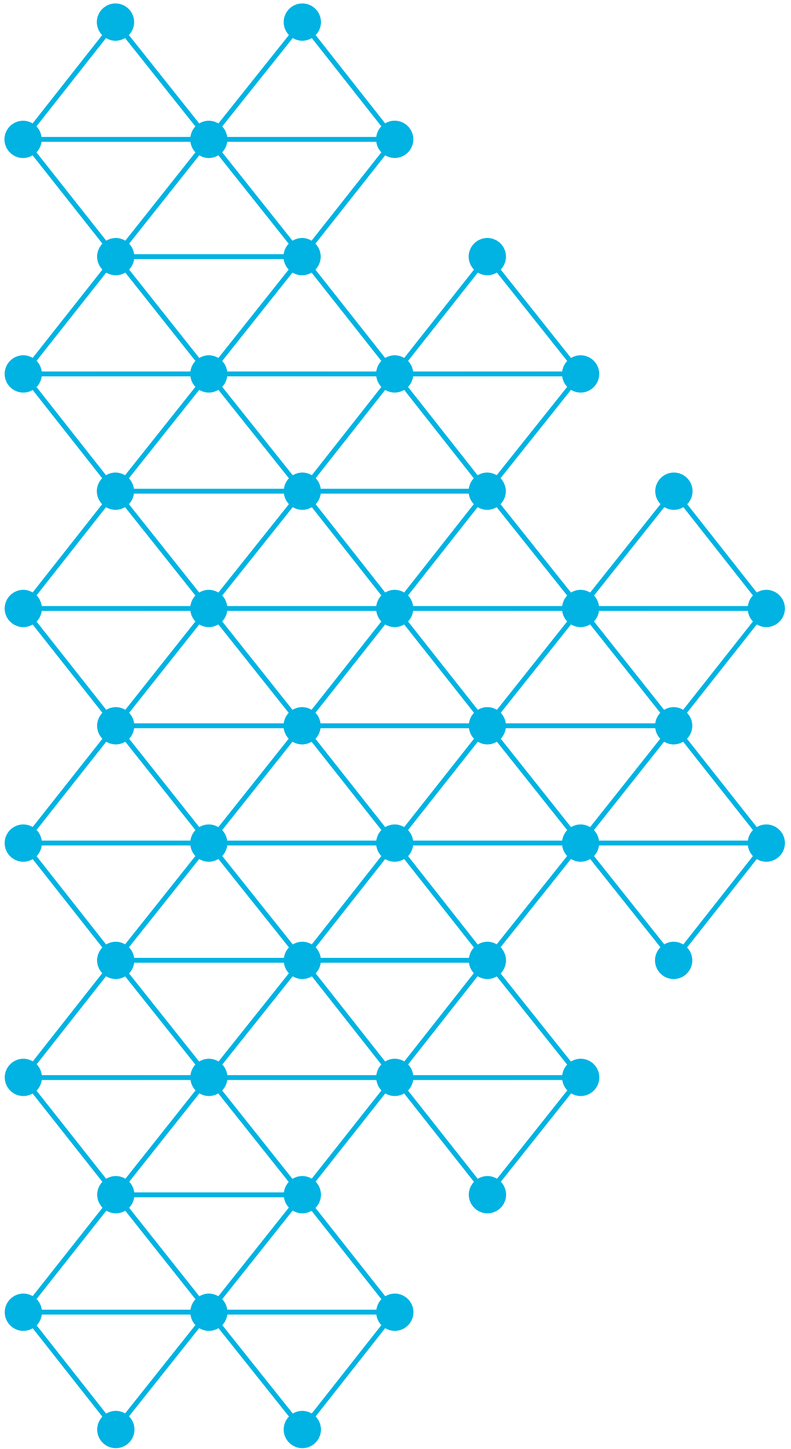
Image credit: Xie et al. (2020) “Self-training with Noisy Student improves ImageNet classification “

Self-Training with Noisy Students



	ImageNet top-1 acc.	ImageNet-A top-1 acc.	ImageNet-C mCE	ImageNet-P mFR
Prev. SOTA	86.4%	61.0%	45.7	27.8
Ours	**88.4%**	**83.7%**	**28.3**	**12.2**

Image credit: Xie et al. (2020) “Self-training with Noisy Student improves ImageNet classification “



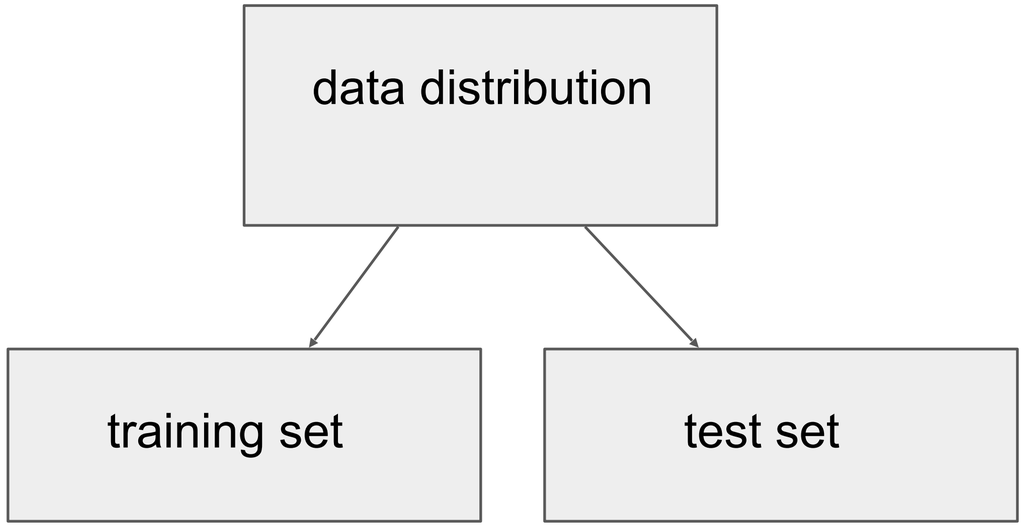
Systematic Generalization

Can Self-Supervised Methods Help?

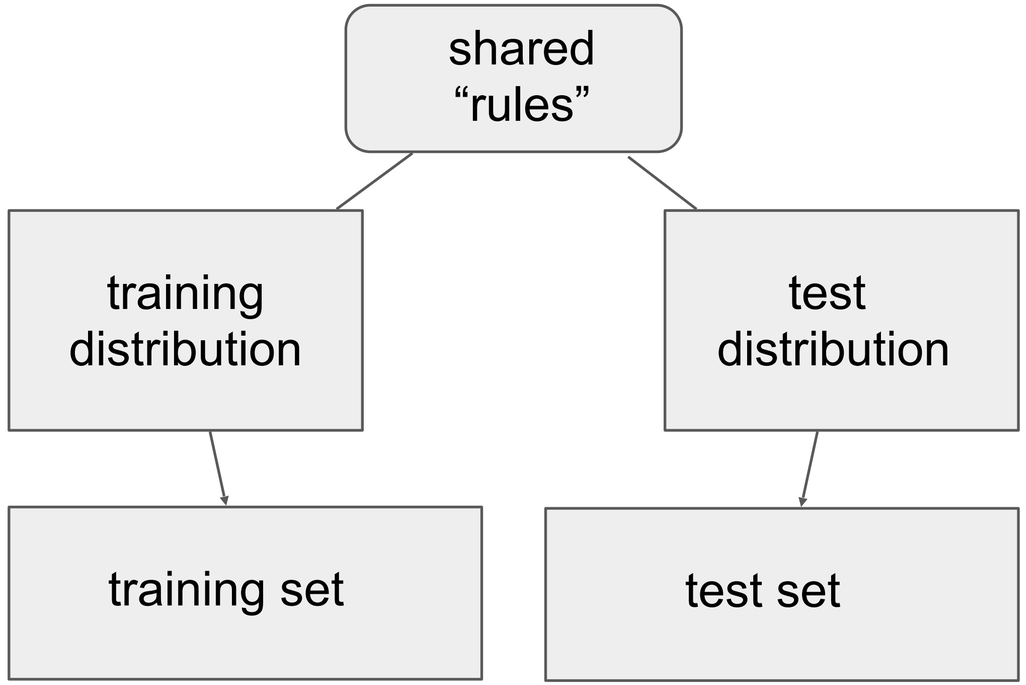
Systematic Generalization

- **Systematic Generalization:** Generalization to examples that may not be drawn from the same distribution as the training data, but that obey the same basic rules of production.

Generalization:



Systematic Generalization:



clarifai ABOUT PRICING DEVELO

cow milk agriculture farm cattle livestock dairy
beef hayfield field grass mammal pasture calf
farmland rural animal pastoral bull grassland

cow beef agriculture cattle milk pasture mammal
livestock farmland grass farm hayfield rural herd
dairy pastoral grassland field calf bull

From Bernhard Schölkopf's slides (2017) - who got it from Pietro Perona (2017)

clarifai ABOUT PRICING DEVELO

cow milk agriculture farm cattle livestock dairy
beef hayfield field grass mammal pasture calf
farmland rural animal pastoral bull grassland

cow beef agriculture cattle milk pasture mammal
livestock farmland grass farm hayfield rural herd
dairy pastoral grassland field calf bull

From Bernhard Schölkopf's slides (2017) - who got it from Pietro Perona (2017)

clarifai ABOUT PRICING DEVELOPE

beach sand travel no person water sea seashore
summer sky outdoors ocean nature

water no person beach seashore sea sand mammal
outdoors travel ocean surf sky

From Bernhard Schölkopf's slides (2017) - who got it from Pietro Perona (2017)

clarifai ABOUT PRICING DEVELOPE

beach sand travel no person water sea seashore
summer sky outdoors ocean nature

water no person beach seashore sea sand mammal
outdoors travel ocean surf sky

From Bernhard Schölkopf's slides (2017) - who got it from Pietro Perona (2017)

Systematic Generalization in Language

Lake and Baroni (2018): Generalization without Systematicity

- **Systematic Compositionality**: The capacity to understand and produce a potentially infinite number of novel combinations from known components.
  - **Example**: (from Lake and Baroni, 2018)
    Consider you've just learned meaning of a new verb "**dax**"
    You understand the meaning of "**dax twice and then dax again**"
- Modern neural networks display impressive generalization.
  Question: Do they demonstrate systematicity?

Systematic Generalization in Language

Lake and Baroni (2018): Generalization without Systematicity

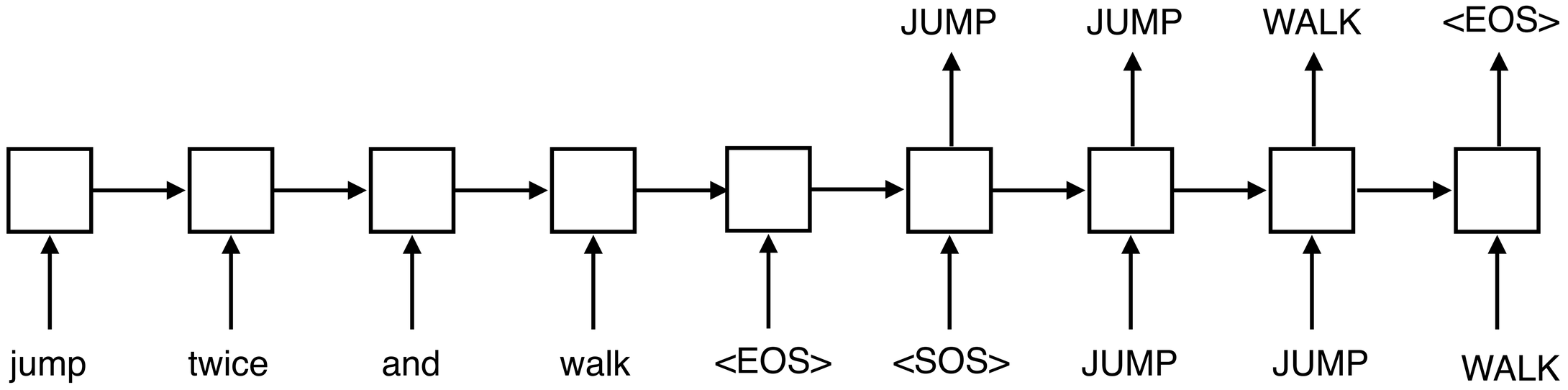
- Develop the **SCAN** dataset and task.

jump	⇒	JUMP
jump left	⇒	LTURN JUMP
jump around right	⇒	RTURN JUMP RTURN JUMP RTURN JUMP RTURN JUMP
turn left twice	⇒	LTURN LTURN
jump thrice	⇒	JUMP JUMP JUMP
jump opposite left and walk thrice	⇒	LTURN LTURN JUMP WALK WALK WALK
jump opposite left after walk around left	⇒	LTURN WALK LTURN WALK LTURN WALK LTURN WALK LTURN LTURN JUMP

Systematic Generalization in Language

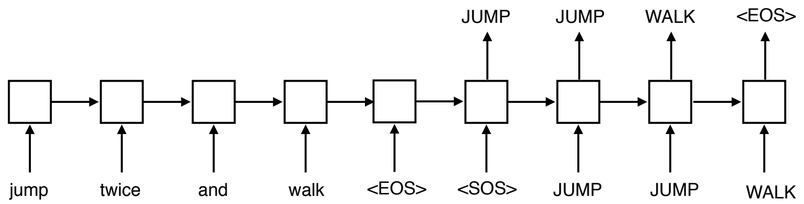
Lake and Baroni (2018): Generalization without Systematicity

- Develop the **SCAN** dataset and task.
- Trained Seq2Seq models (RNNs, actually LSTM) on this “translation” task.



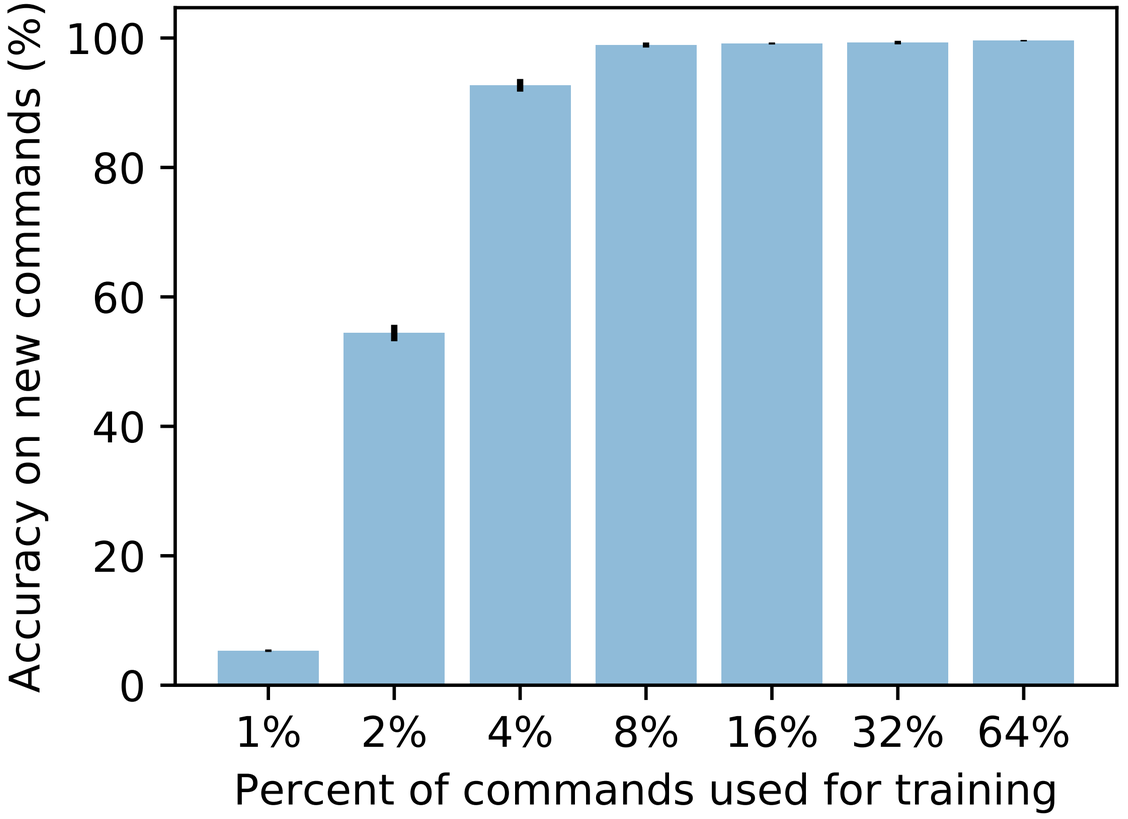
Systematic Generalization in Language

Lake and Baroni (2018): Generalization without Systematicity



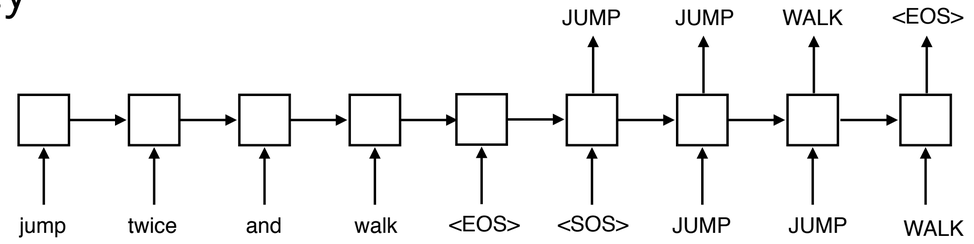
Baseline Experiment (i.e. Standard Generalization):

- Randomly sample a training set of commands, test on the rest.



Systematic Generalization in Language

Lake and Baroni (2018): Generalization without Systematicity



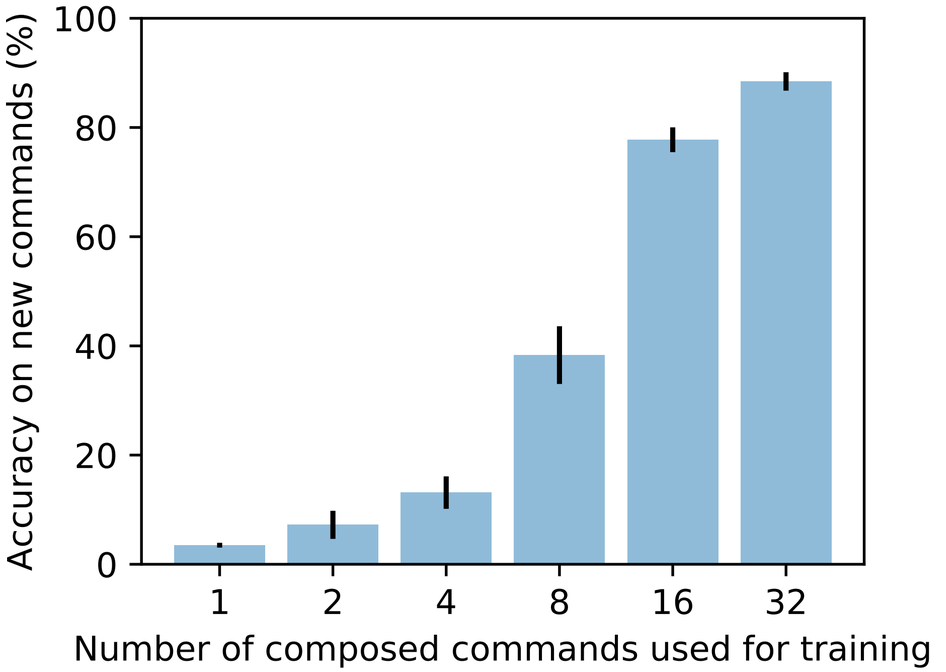
Training phase:

- For one action (e.g., “jump”), model exposed to only primitive commands and a few composed commands.
- Also exposed to all primitive and composed commands for all other actions (e.g., “run”, “run twice”, “walk”, “walk opposite left and run twice”, etc.).

Test Phase:

- Execute all other composed commands for the primitive action (e.g., “jump twice”, “jump opposite left and run twice”, etc.).

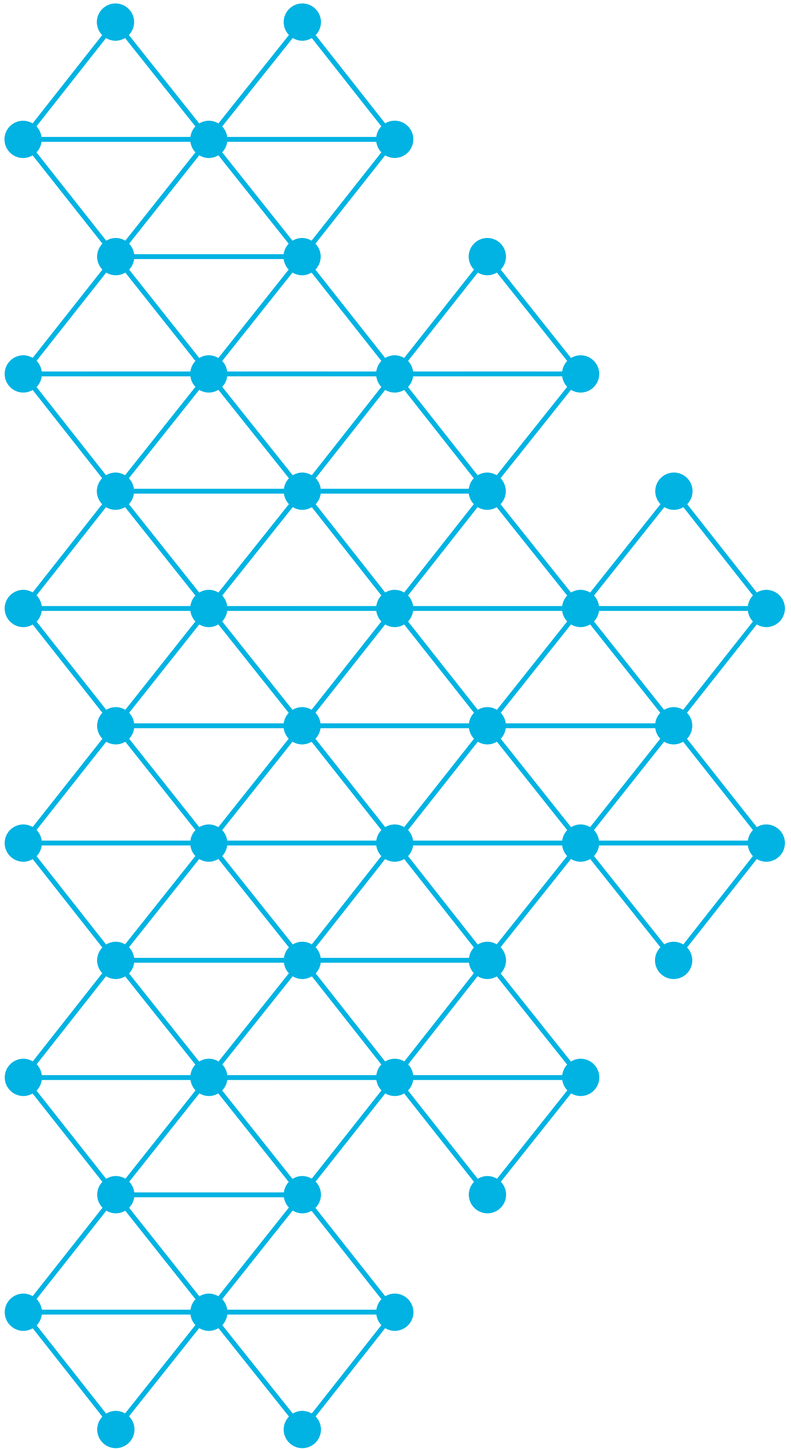
Compositionality: if you know the meaning of “run”, “jump” and “run twice”, you should also understand what “jump twice” means.



Systematic Generalization — Solution Strategies

An active area of current research, no clear answers yet.

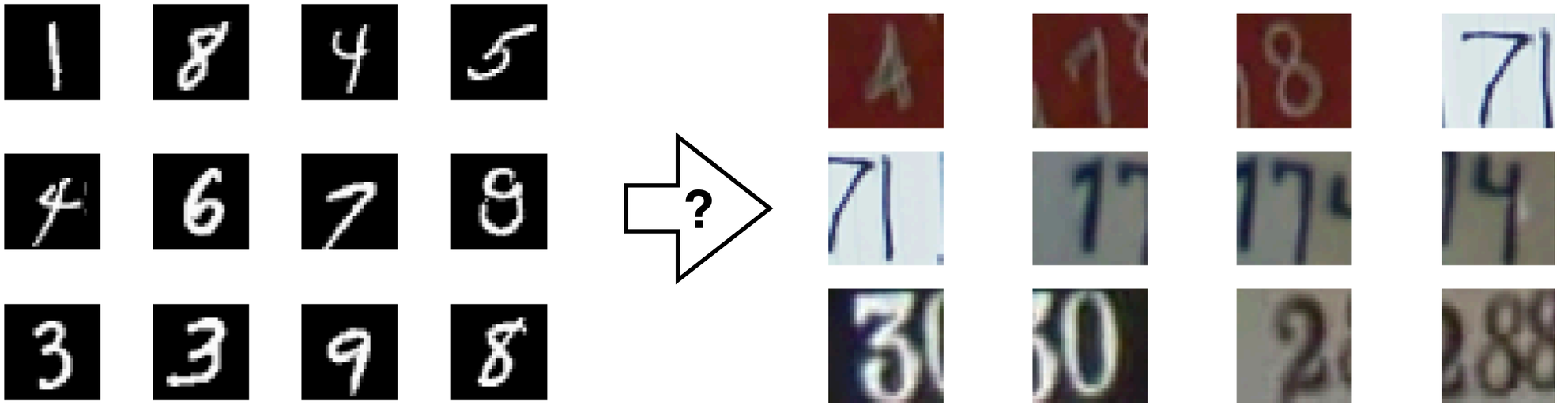
- Modularity -> Compositionality -> Systematic Generalization?
  - Maybe Neural Modular Networks (NMNs) offer a solution?
- Top-down solutions: strong priors to enforce compositional structure of the representation.
- Bottom-up solutions? (really really speculative): Language emergence theory from cog. sci. such as Iterated Learning (Kirby et al) might have promise.



Domain Adaptation / Domain Translation

Exploiting other data domains or tasks

- Self-supervised learning methods exploit our domain knowledge to design supervised learning tasks to learn representations that lead to good discriminative performance **for your target task**.
- An alternative is to use other data sources (labeled or unlabeled) to help learn representations that are useful **for your target task**.



Learning the representation h for unsupervised DA

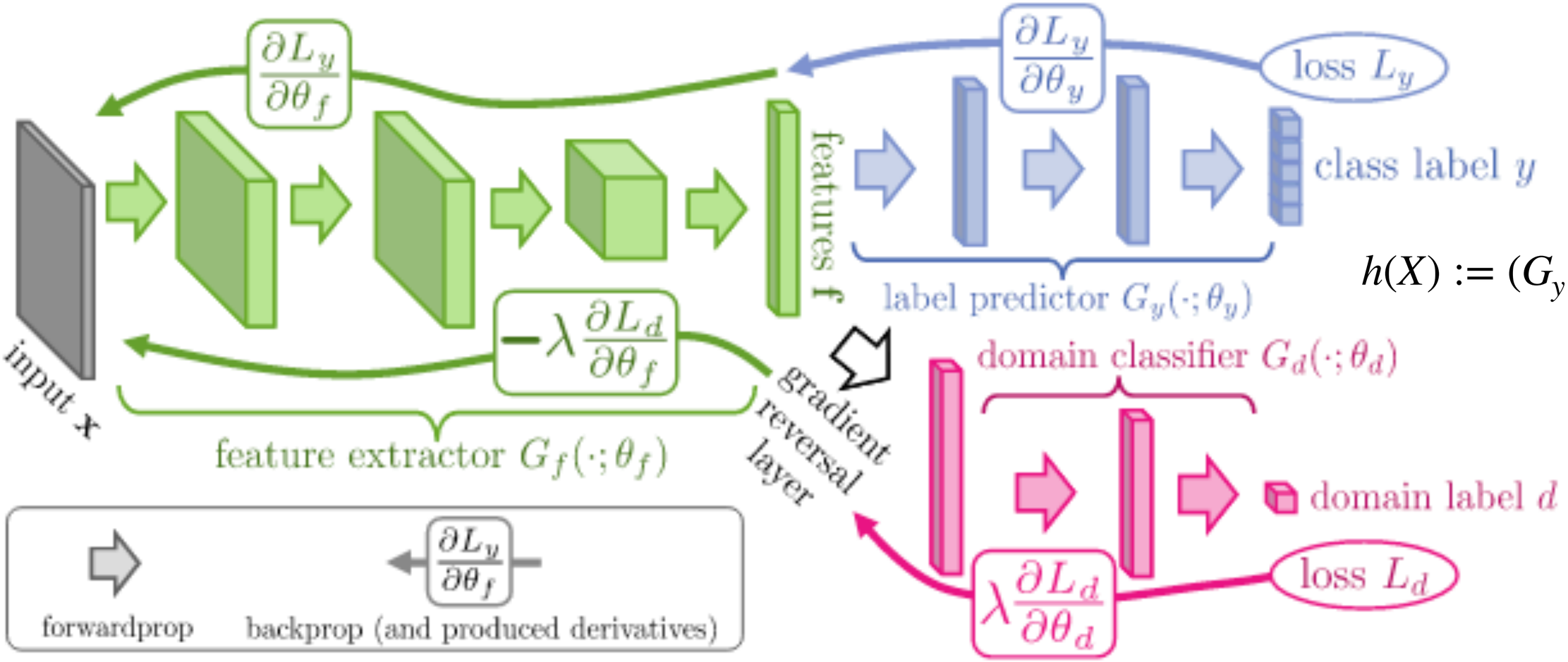
MILA

Considerations

- h has to be robust to changes of dist. from $\mathbb{P}_{x_1}$ (domain 1) and $\mathbb{P}_{x_2}$ (domain 2).
- h learned in the unsupervised domain adaptation setting (S+ T-)
- Currently SOTA: a collection of “tricks” that seems to work very well.

Trick 1: Domain Adversarial Neural Network (DANN)

Image credits: Yaroslav Ganin, Hana Ajakan, Hugo Larochelle, François Laviolette, Mario Marchand, Victor Lempitsky. "Domain-Adversarial Training of Neural Networks", *The Journal of Machine Learning Research*, 17(1), 2016.



$$h(X) := (G_y \circ G_f)(X).$$

Equivalent to a GAN on the features space

Trick 2: Cluster Assumption

- Classifier predictions should have a minimum conditional entropy for the source and target domains.

$$\min_h H(h(X) \mid X) = E_{x \sim \mathbb{P}_X} \min_h \overbrace{\sum_{i=1}}^{\text{N classes}} - h(x)_i \log h(x)_i.$$

- Done by default when learning a classifier with labels.

References: https://papers.nips.cc/paper/2740-semi-supervised-learning-by-entropy-minimization.pdf and https://arxiv.org/abs/1802.08735

Trick 3: Virtual Adversarial Training (VAT)

- Enforce a robust classification around an ϵ-ball of every samples.

$$\min_{h} E_{x \sim \mathbb{P}_X} \max_{||r|| < \epsilon} \mathsf{D}_{\mathsf{KL}}(h(x) \,||\, h(x+r)).$$

References: https://arxiv.org/abs/1704.03976 and https://arxiv.org/abs/1802.08735

Trick 4: Virtual mixup training (VMT)

- Further regularize the predictions of the linear interpolation of samples.
- The prediction of the interpolation of two samples x and x' should be an interpolation of the prediction of x and x'.

$$\tilde{x} = \alpha x + (1-\alpha)x' \text{ and } \tilde{y} = \alpha h(x) + (1-\alpha)h(x')$$

$$\min_h E_{x \sim \mathbb{P}_X} - \tilde{y}^\top \log h(\tilde{x})$$

References: https://arxiv.org/abs/1710.09412 and https://arxiv.org/abs/1905.04215

Putting it all together (S+ T-)

- In a nutshell: There is no free lunch! For unsupervised domain adaptation:
  1. Determine which assumptions you can make.
  2. For each assumption, define an objective to enforce it.

- Put together, the objective functions yield the following optimization:

$$\min_h \mathscr{L}_{\text{classifier}}(h) + \lambda_1 \mathscr{L}_{\text{cluster}}(h) + \lambda_2 \mathscr{L}_{\text{DANN}}(h) + \lambda_3 \mathscr{L}_{\text{VAT}}(h) + \lambda_4 \mathscr{L}_{VMT}(h).$$

Unsupervised domain adaptation (S+ T-) Results

DIRT-T: fine-tuning the objective function on the target domain

Source Target	MNIST SVHN	SVHN MNIST	MNIST MNIST-M	SYN SVHN	CIFAR STL	STL CIFAR
MMD (Long et al. 2015)	-	71.1	76.9	88.0	-	-
DANN (Ganin et al. 2016)	35.7	71.1	81.5	90.3	-	-
DRCN (Ghifary et al. 2016)	40.1	82.0	-	-	66.4	58.7
DSN (Bousmalis et al. 2016b)	-	82.7	83.2	91.2	-	-
kNN-Ad (Sener et al. 2016)	40.3	78.8	86.7	-	-	-
PixelDA (Bousmalis et al. 2017)	-	-	98.2	-	-	-
ATT (Saito, Ushiku, and Harada 2017)	52.8	86.2	94.2	92.9	-	-
Π-model (aug) (French, Mackiewicz, and Fisher 2018)	71.4	92.0	-	94.2	76.3	64.2
Without Instance-Normalized Input:						
Source-Only	27.9	77.0	58.5	86.9	76.3	63.6
VADA (Shu et al. 2018)	47.5	97.9	97.7	94.8	80.0	73.5
Co-DA (Kumar et al. 2018)	55.3	**98.8**	**99.0**	96.1	81.4	76.4
VMT (ours)	**59.3**	**98.8**	**99.0**	**96.2**	**82.0**	**80.2**
VADA + DIRT-T (Shu et al. 2018)	54.5	99.4	98.9	96.1	-	75.3
Co-DA + DIRT-T (Kumar et al. 2018)	63.0	99.4	**99.1**	**96.5**	-	77.6
VMT + DIRT-T (ours)	**88.1**	**99.5**	**99.1**	**96.5**	-	**80.6**
With Instance-Normalized Input:						
Source-Only	40.9	82.4	59.9	88.6	77.0	62.6
VADA (Shu et al. 2018)	73.3	94.5	95.7	94.9	78.3	71.4
Co-DA (Kumar et al. 2018)	81.7	98.7	**98.0**	96.0	80.6	74.7
VMT (ours)	**85.2**	**98.9**	**98.0**	**96.4**	**81.3**	**79.5**
VADA + DIRT-T (Shu et al. 2018)	76.5	**99.4**	98.7	96.2	-	73.3
Co-DA + DIRT-T (Kumar et al. 2018)	88.0	**99.4**	98.8	96.5	-	75.9
VMT + DIRT-T (ours)	**95.1**	**99.4**	**98.9**	**96.6**	-	**80.2**

Achieve 95% accuracy on SVHN without any labelled SVHN samples!

Table credits: Mao, X., Ma, Y., Yang, Z., Chen, Y., and Li, Q. Virtual mixup training for unsupervised domain adaptation. https://arxiv.org/pdf/1905.04215.pdf