UNIVERSITY OF NORTH TEXAS

CSCE 5218 & 4930
Deep Learning

Advanced Topics

Slides adapted from Adriana Kovashka

Plan for this lecture

- Alternative representations
  - I. Graph networks
- Alternative learning mechanisms
  - II. Self supervision
  - III. Reinforcement learning
- Alternative tasks
  - IV. Generation
- V. Bias and ethics (optional)

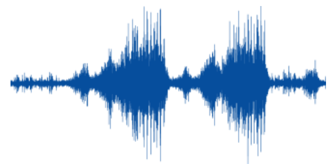
Part I: Graph Networks

- Types of graph networks
  - Graph convolutional networks
  - Graph attention networks
- Applications
  - Semi-supervised learning

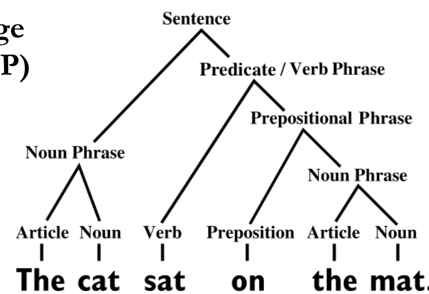
Types of data typically handled with Deep Learning

IMAGENET

Speech data

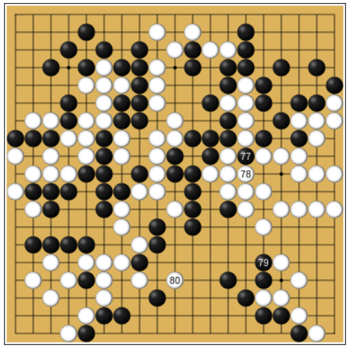


Natural language processing (NLP)



...

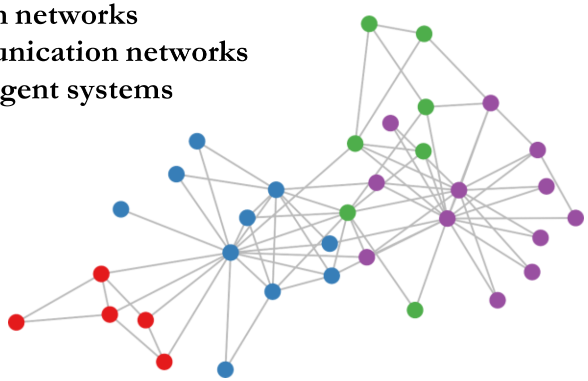
Grid games

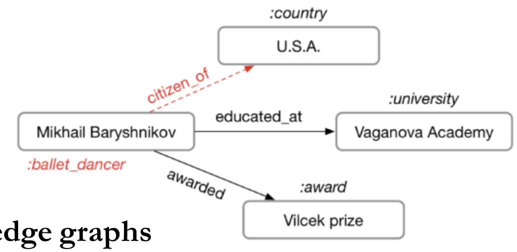


Graph-structured data

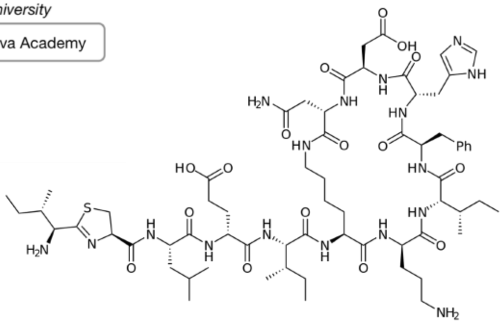
A lot of real-world data does not “live” on grids

Social networks
Citation networks
Communication networks
Multi-agent systems

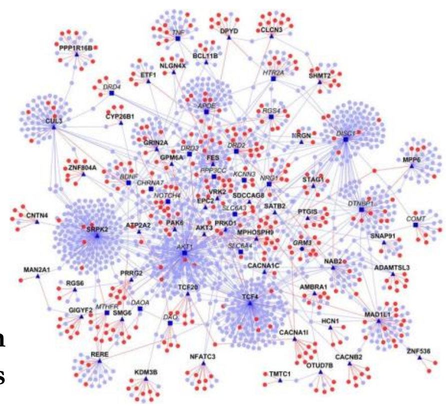




Knowledge graphs



Molecules



Protein interaction networks

Road maps

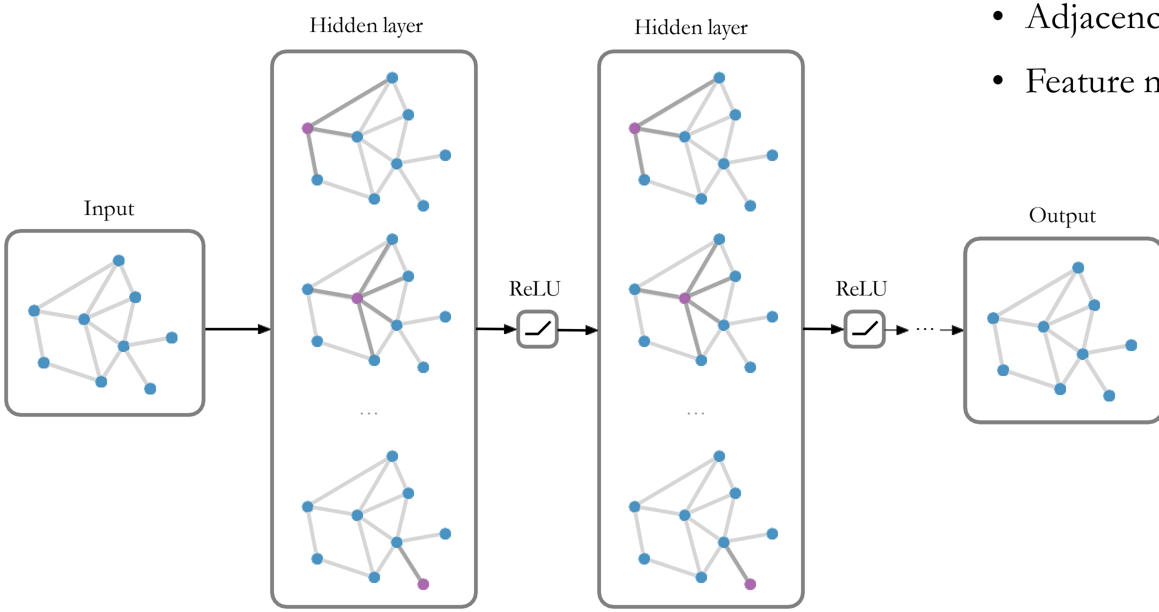
Standard deep learning architectures like CNNs and RNNs don’t work here!

Graph Neural Networks (GNNs)

The bigger picture:

Notation: $G = (\mathbf{A}, \mathbf{X})$

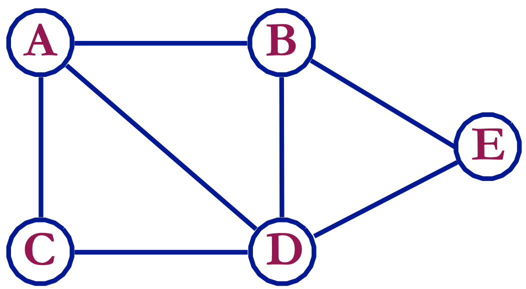
- Adjacency matrix $\mathbf{A} \quad \mathbb{R}^{N \times N}$
- Feature matrix $\mathbf{X} \quad \mathbb{R}^{N \times F}$



Main idea: Pass messages between pairs of nodes & agglomerate

Graph convolutional networks

Graph: $G = (\mathcal{V}, \mathcal{E})$



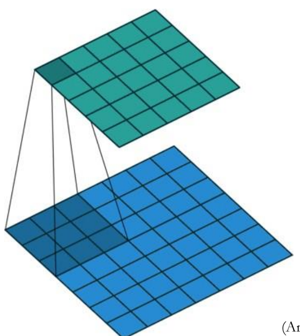
Adjacency matrix: $\mathbf{A}$

$$
\begin{array}{c c}
 & \begin{array}{ccccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} & \mathbf{E} \end{array} \\
\begin{array}{c} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \\ \mathbf{D} \\ \mathbf{E} \end{array} &
\left[\begin{array}{ccccc}
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0
\end{array}\right]
\end{array}
$$

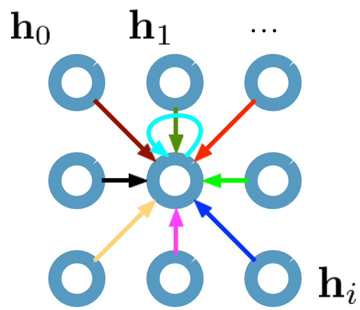
Kipf and Welling, “Semi-supervised learning with deep generative models”, ICLR 2017 (slides by Thomas Kipf)

Recap: Convolutional neural networks (on grids)

Single CNN layer with 3x3 filter:



(Animation by Vincent Dumoulin)



Update for a single pixel:

- Transform messages individually $\mathbf{W}_i \mathbf{h}_i$
- Add everything up $\sum_i \mathbf{W}_i \mathbf{h}_i$

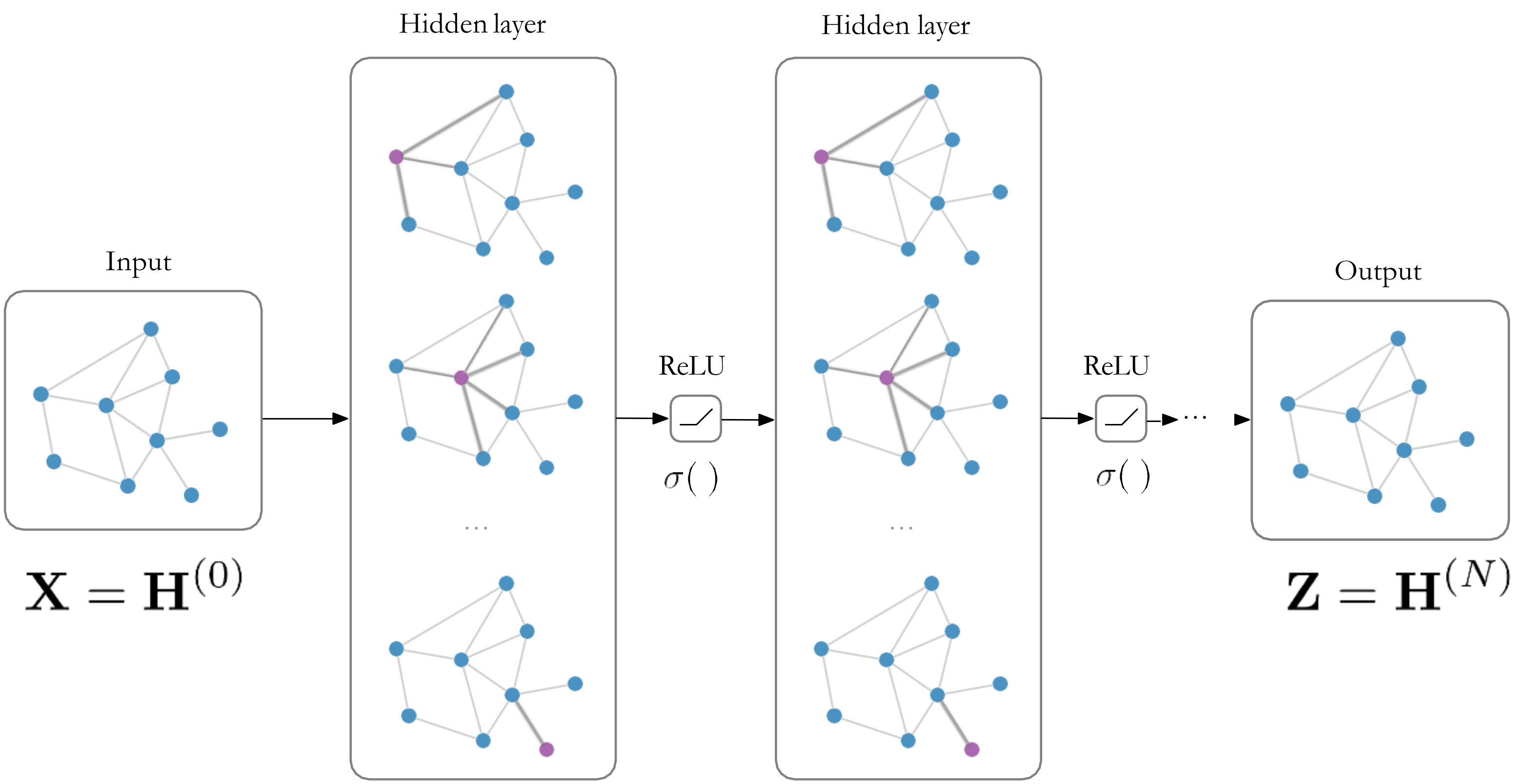
$\mathbf{h}_i$ *in* R^F are (hidden layer) activations of a pixel/node

Full update:

$$\mathbf{h}_4^{(l+1)} = \sigma\left(\mathbf{W}_0^{(l)}\mathbf{h}_0^{(l)} + \mathbf{W}_1^{(l)}\mathbf{h}_1^{(l)} + \cdots + \mathbf{W}_8^{(l)}\mathbf{h}_8^{(l)}\right)$$

Graph convolutional networks

Input: Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times E}$, preprocessed adjacency matrix $\hat{\mathbf{A}}$



$$\mathbf{X} = \mathbf{H}^{(0)} \qquad \sigma(\,) \qquad \mathbf{Z} = \mathbf{H}^{(N)}$$

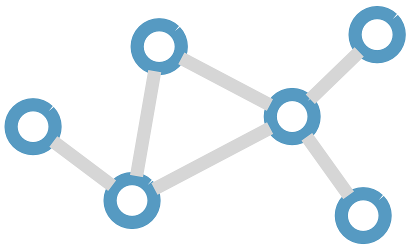
$$\mathbf{H}^{(l+1)} = \sigma\left(\hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right)$$

Kipf and Welling, "Semi-supervised learning with deep generative models", ICLR 2017 (slides by Thomas Kipf)

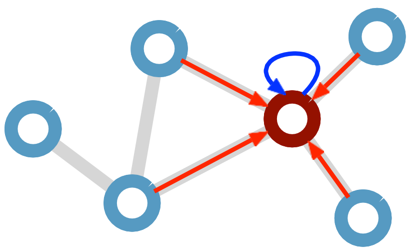
Graph convolutional networks (GCNs)

Kipf & Welling (ICLR 2017), related previous works by Duvenaud et al. (NIPS 2015) and Li et al. (ICLR 2016)

Consider this undirected graph:



Calculate update for node in red:



Update rule:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)}\mathbf{W}_0^{(l)} + \sum_{j\in\mathcal{N}_i}\frac{1}{c_{ij}}\mathbf{h}_j^{(l)}\mathbf{W}_1^{(l)}\right)$$

Scalability: subsample messages [Hamilton et al., NIPS 2017]

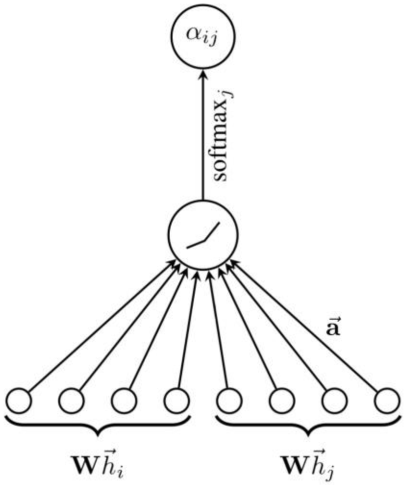
$\mathcal{N}_i$: neighbor indices

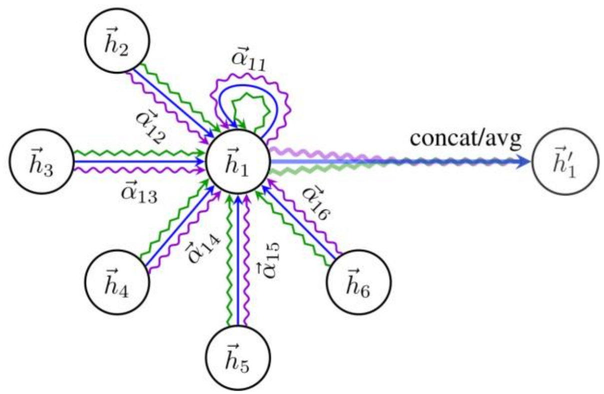
c_{ij} : norm. constant (fixed/trainable)

Graph neural networks with attention

Monti et al. (CVPR 2017), Hoshen (NIPS 2017), Veličković et al. (ICLR 2018)

https://arxiv.org/pdf/1710.10903.pdf

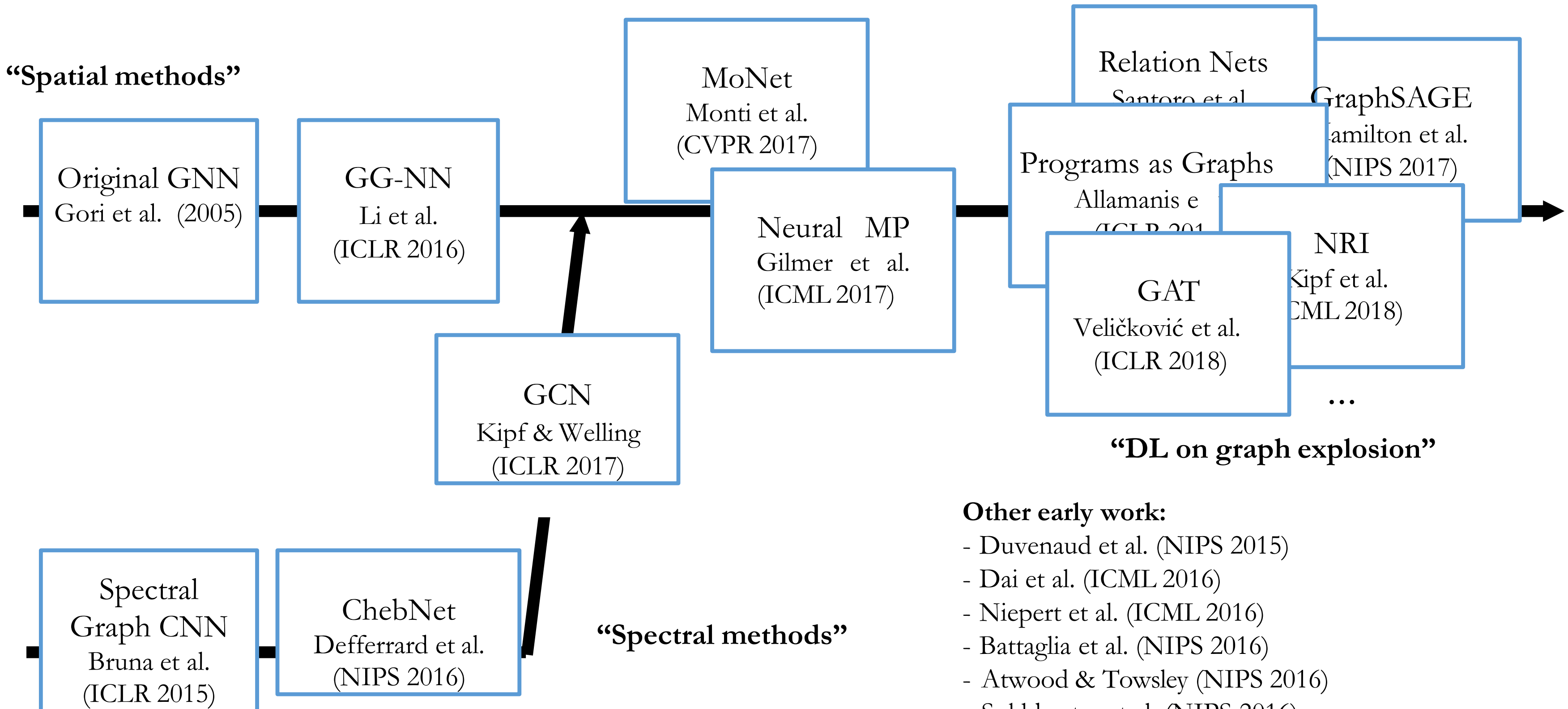




[Figure from Veličković et al. (ICLR 2018)]

$$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k \mathbf{W}^k \vec{h}_j\right) \qquad \alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

A brief history of graph neural nets



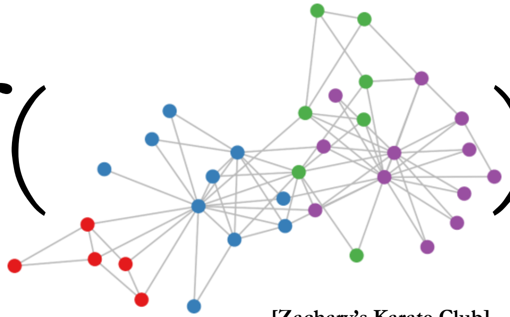
Other early work:
- Duvenaud et al. (NIPS 2015)
- Dai et al. (ICML 2016)
- Niepert et al. (ICML 2016)
- Battaglia et al. (NIPS 2016)
- Atwood & Towsley (NIPS 2016)
- Sukhbaatar et al. (NIPS 2016)

(slide inspired by Alexander Gaunt's talk on GNNs)

What do learned representations look like?

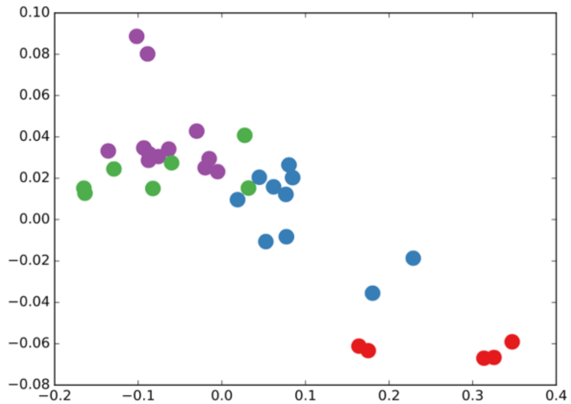
Forward pass through **untrained** 3-layer GCN model

Parameters initialized randomly

f() =

[Zachary's Karate Club]

2-dim output per node



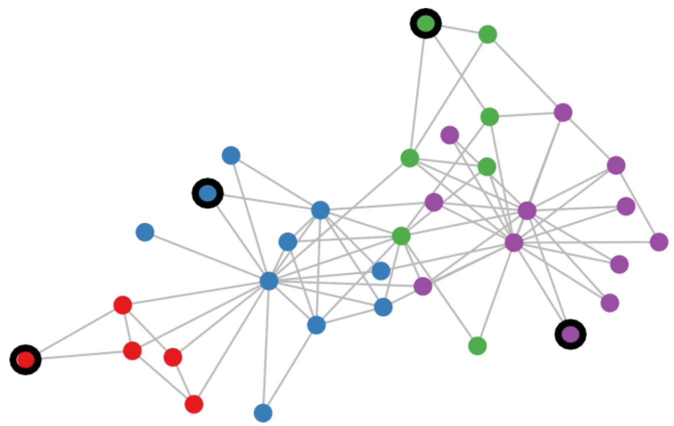
Semi-supervised classification on graphs

Setting:

Some nodes are labeled (black circle) All other nodes are unlabeled

Task:

Predict node label of unlabeled nodes



Evaluate loss on labeled nodes only:

$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$

$\mathcal{Y}_L$ set of labeled node indices

$\mathbf{Y}$ label matrix

$\mathbf{Z}$ GCN output (after softmax)

Application: Classification on citation networks

Input: Citation networks (nodes are papers, edges are citation links, optionally bag-of-words features on nodes)

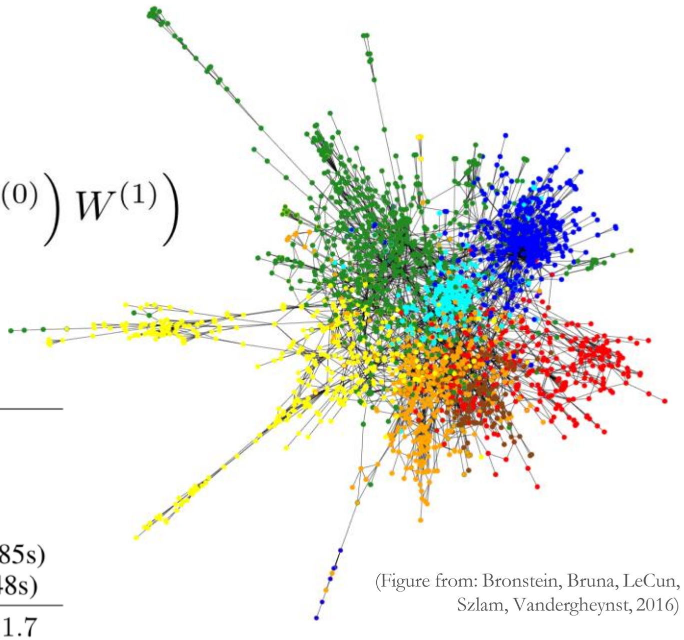
Target: Paper category (e.g. stat.ML, cs.LG, …)

Model: 2-layer GCN $$Z = f(X, A) = \mathrm{softmax}\Big(\hat{A}\ \mathrm{ReLU}\Big(\hat{A}XW^{(0)}\Big)\,W^{(1)}\Big)$$

Classification results (accuracy)

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [24]	59.6	59.0	71.1	26.7
LP [27]	45.3	68.0	63.0	26.5
DeepWalk [18]	43.2	67.2	65.3	58.1
Planetoid* [25]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

no input features



(Figure from: Bronstein, Bruna, LeCun, Szlam, Vandergheynst, 2016)

Kipf & Welling, Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Part II: Self-Supervised Learning

- Learn representations from context in raw data
- Language – predict nearby words [*already covered*]
  - Word2Vec
  - Transformers
- Vision – predict pixels from other pixels
  - Predict nearby patches in an image
  - Predict order of frames in a video
  - Predict ranking

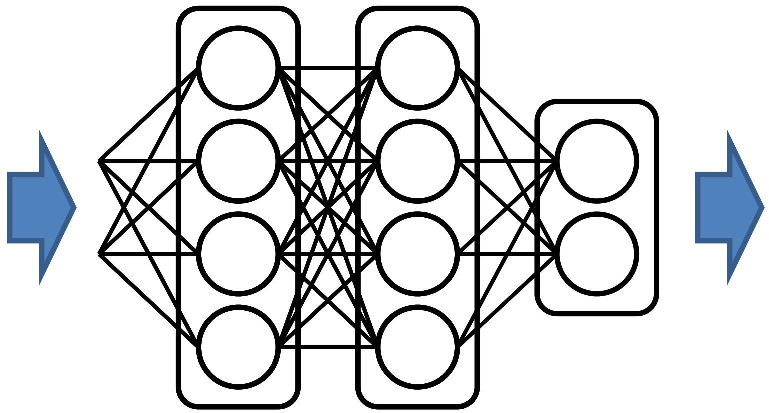
Jitendra Malik: "**Supervision** is the opium of the AI researcher"
Alyosha Efros: "The AI revolution will not be **supervised**"
Yann LeCun: "**Self-supervised** learning is the cake, **supervised** learning is the icing on the cake, **reinforcement learning** is the cherry on the cake"

Unsupervised Visual Representation Learning by Context Prediction

Carl Doersch, Alexei Efros and Abhinav Gupta

ICCV 2015

ImageNet + Deep Learning

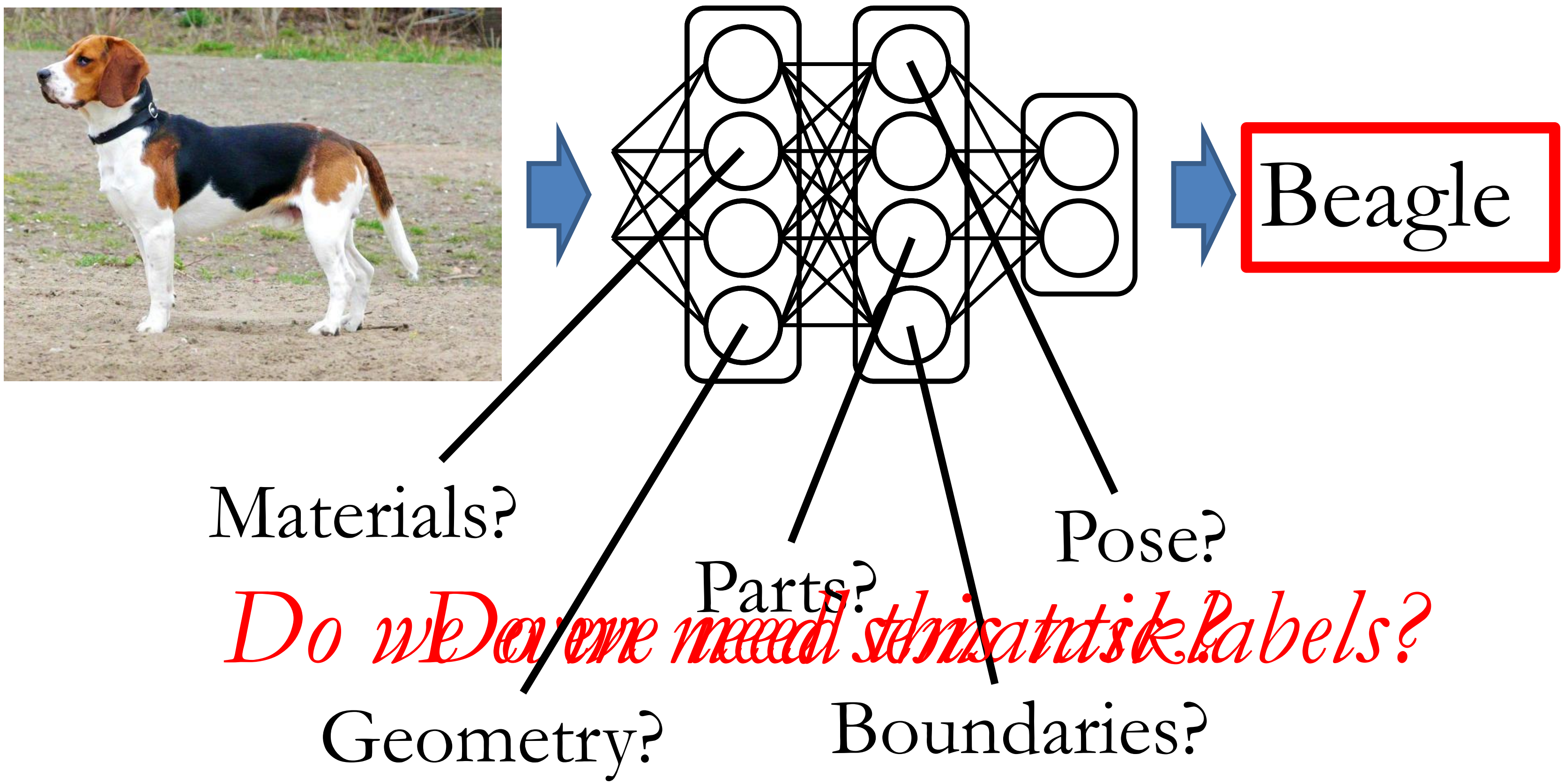


Beagle

- Image Retrieval
- Detection (RCNN)
- Segmentation (FCN)
- Depth Estimation
- …

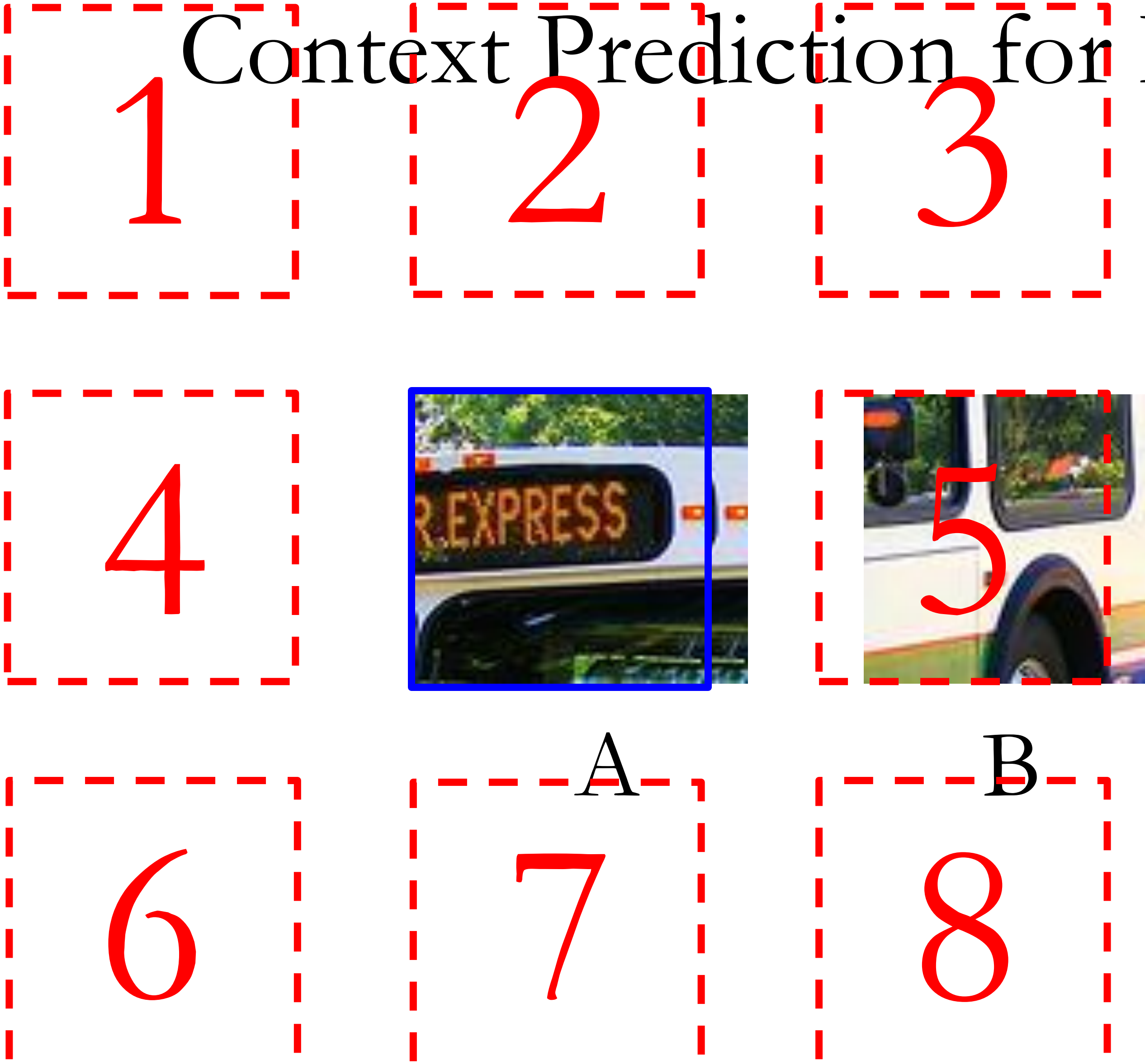
Doersch et al., "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

ImageNet + Deep Learning



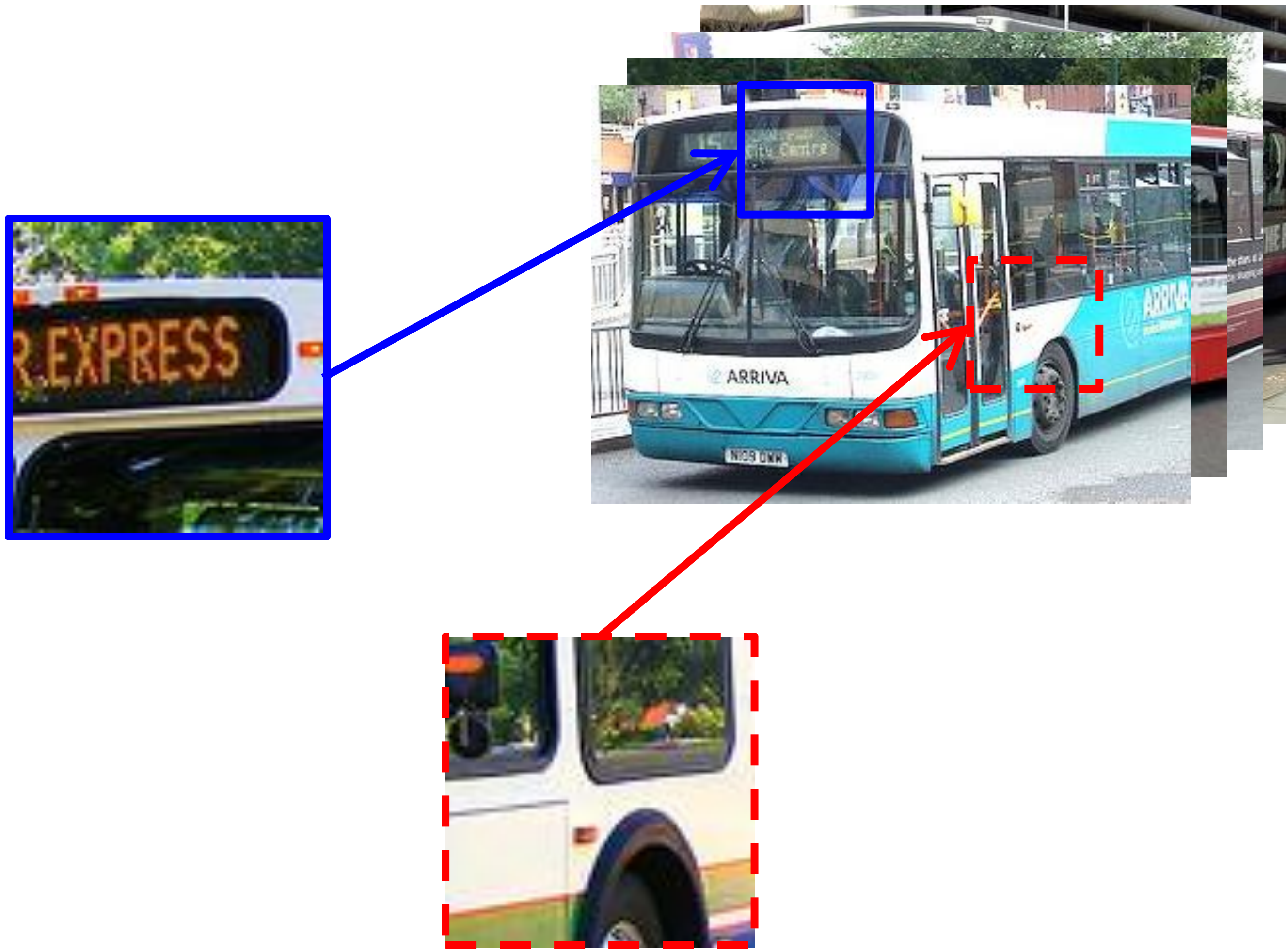
Doersch et al., “Unsupervised Visual Representation Learning by Context Prediction”, ICCV 2015

Context Prediction for Images



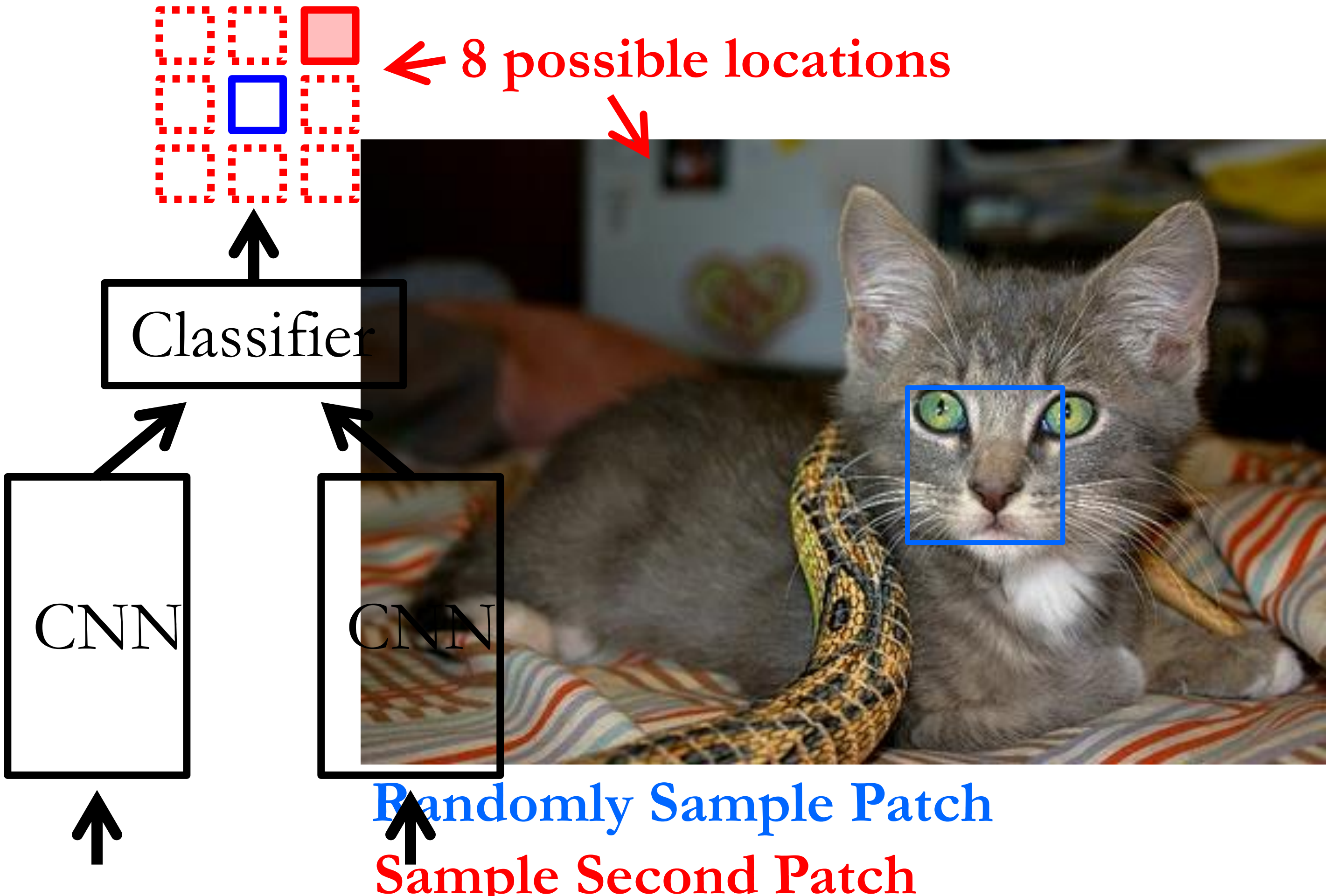
Doersch et al., "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

Semantics from a non-semantic task

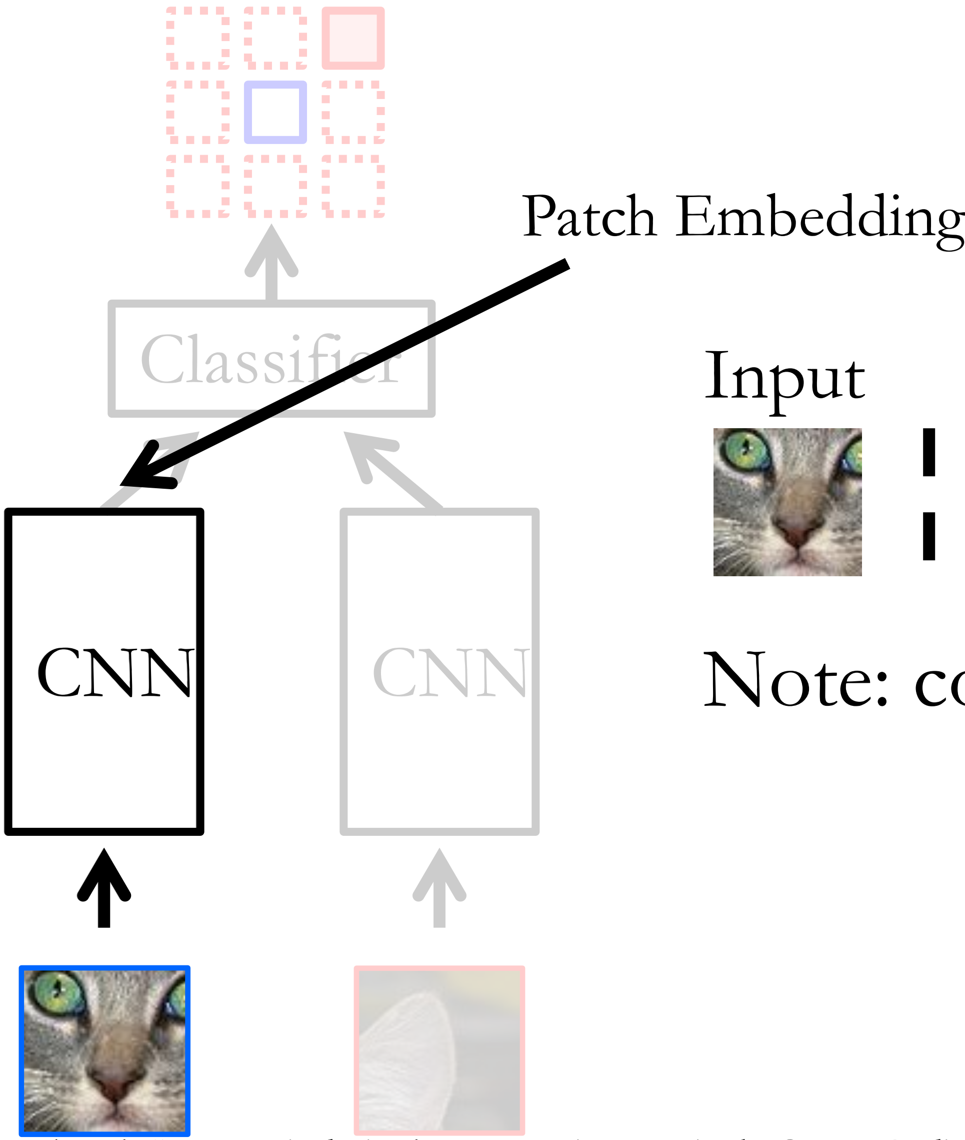


Doersch et al., "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

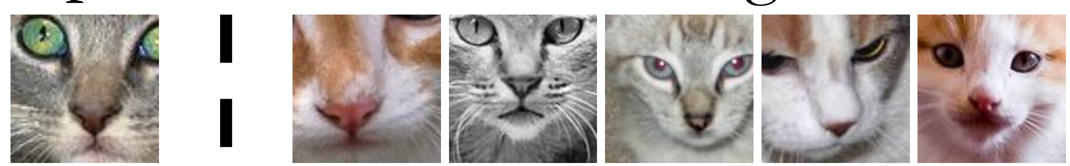
Relative Position Task



Doersch et al., "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015



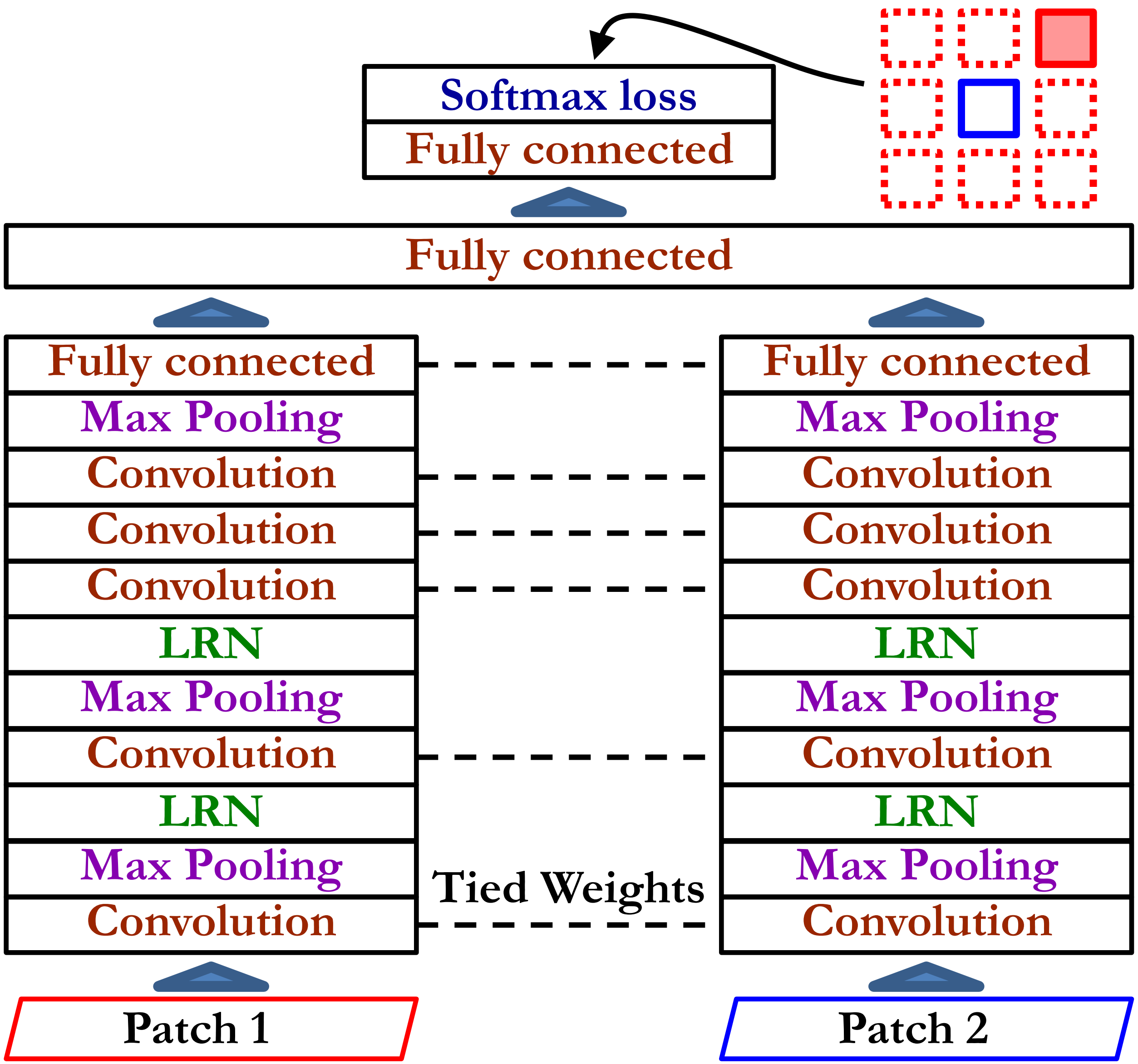
Input Nearest Neighbors



Note: connects ***across*** instances!

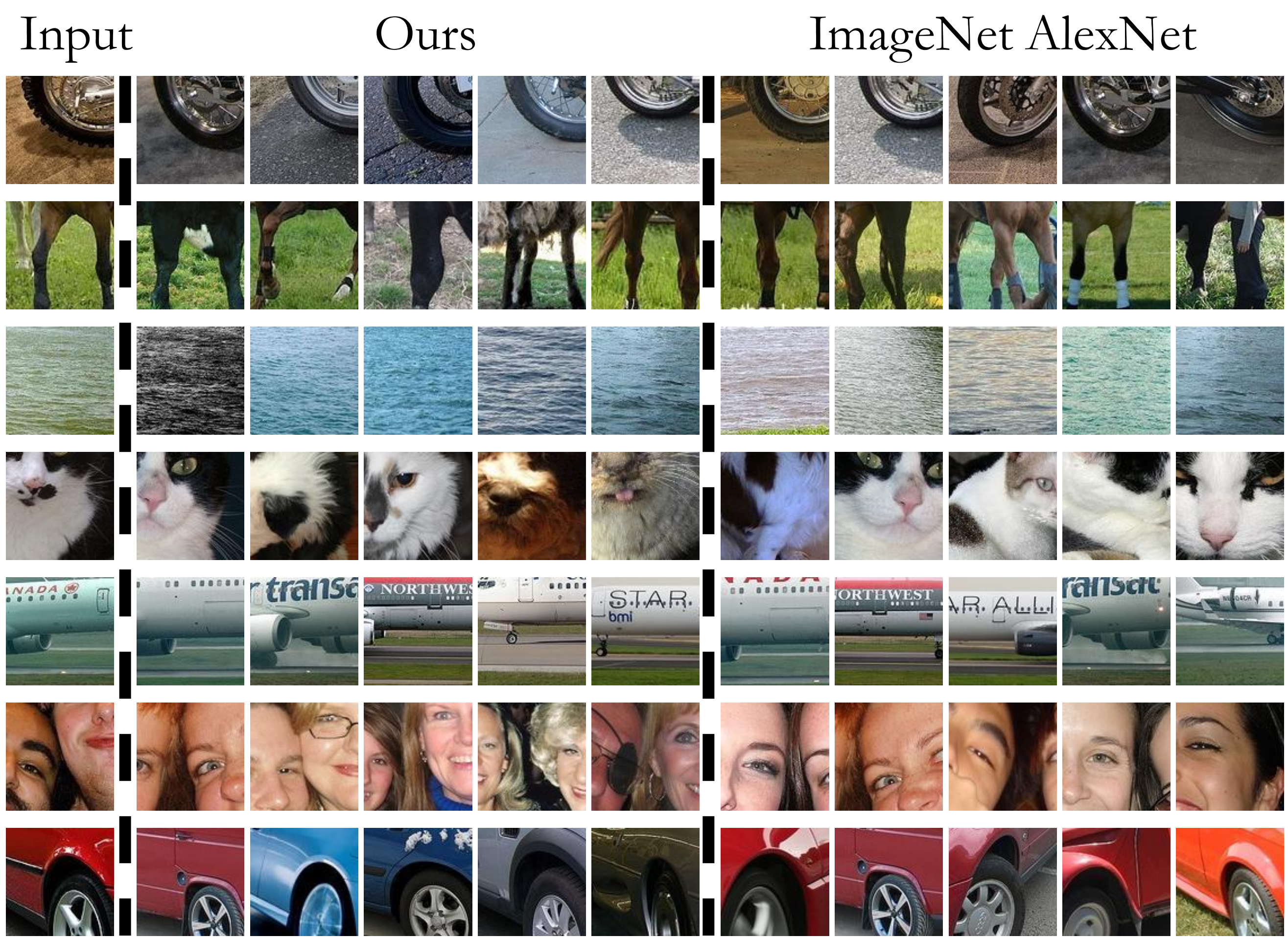
Doersch et al., "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

Architecture



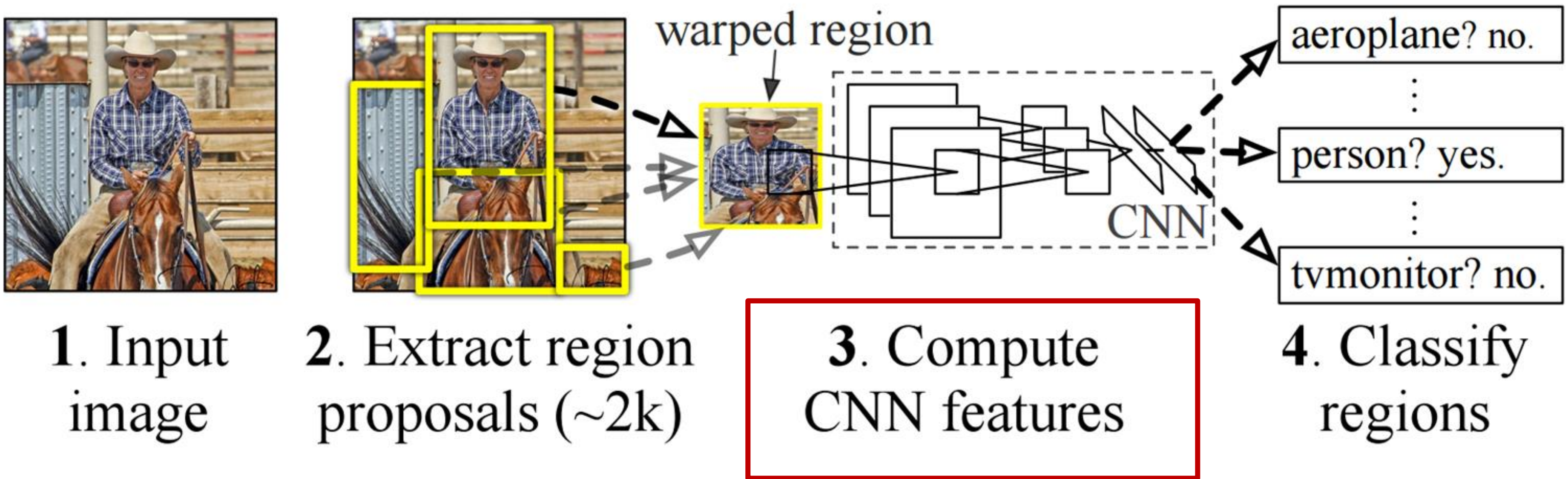
Doersch et al., "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

What is learned?



Doersch et al., "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

Pre-Training for R-CNN



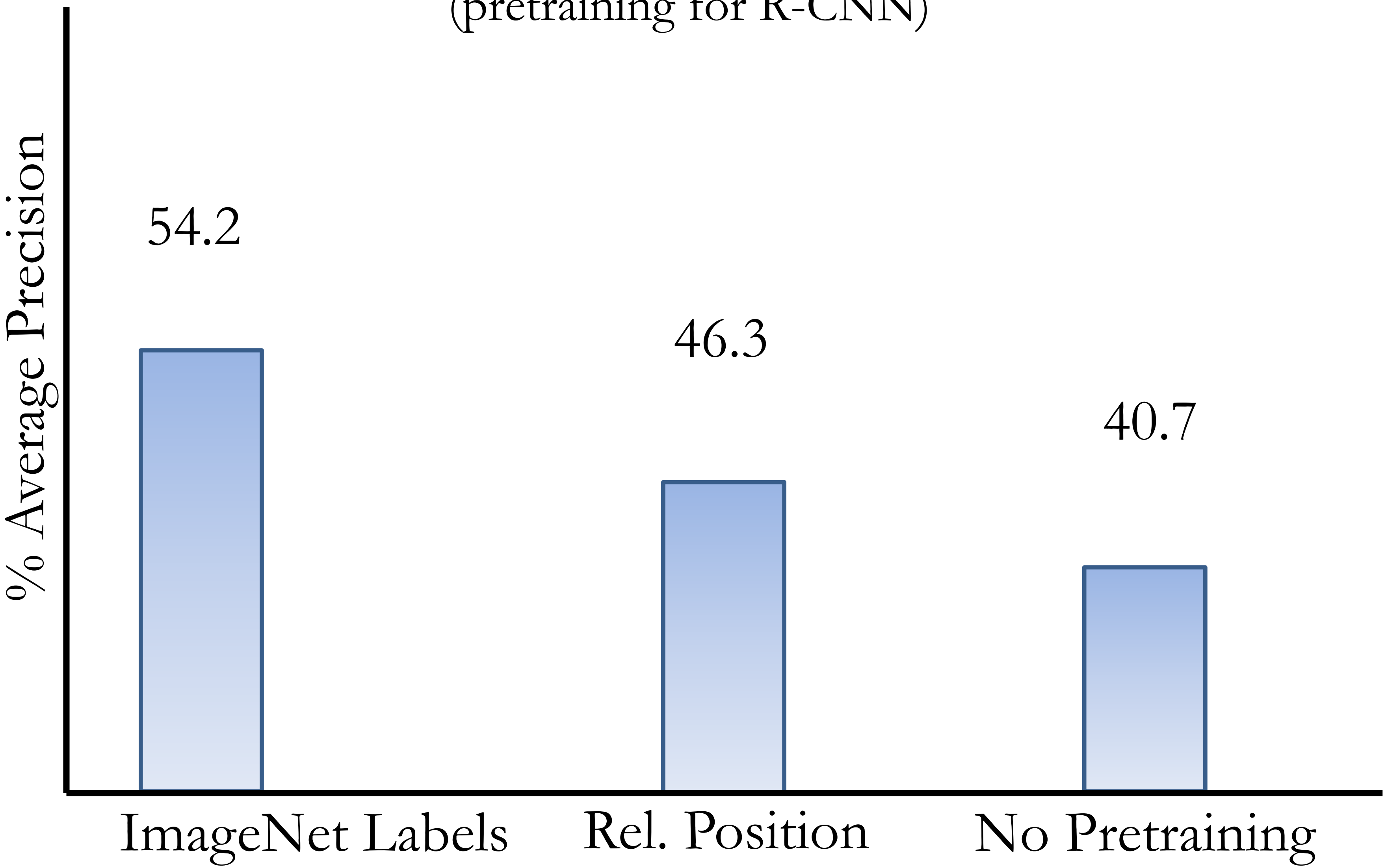
Pre-train on relative-position task, w/o labels

Doersch et al., "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

[Girshick et al. 2014]

VOC 2007 Performance

(pretraining for R-CNN)



Doersch et al., "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015

Which will be better?

- Option 1: pretrain (unsup) on dataset B
- Option 2: pretrain (sup) on dataset A
- Test on dataset B

Shuffle and Learn: Unsupervised Learning using Temporal Order Verification

Ishan Misra, C. Lawrence Zitnick, and Martial Hebert

ECCV 2016

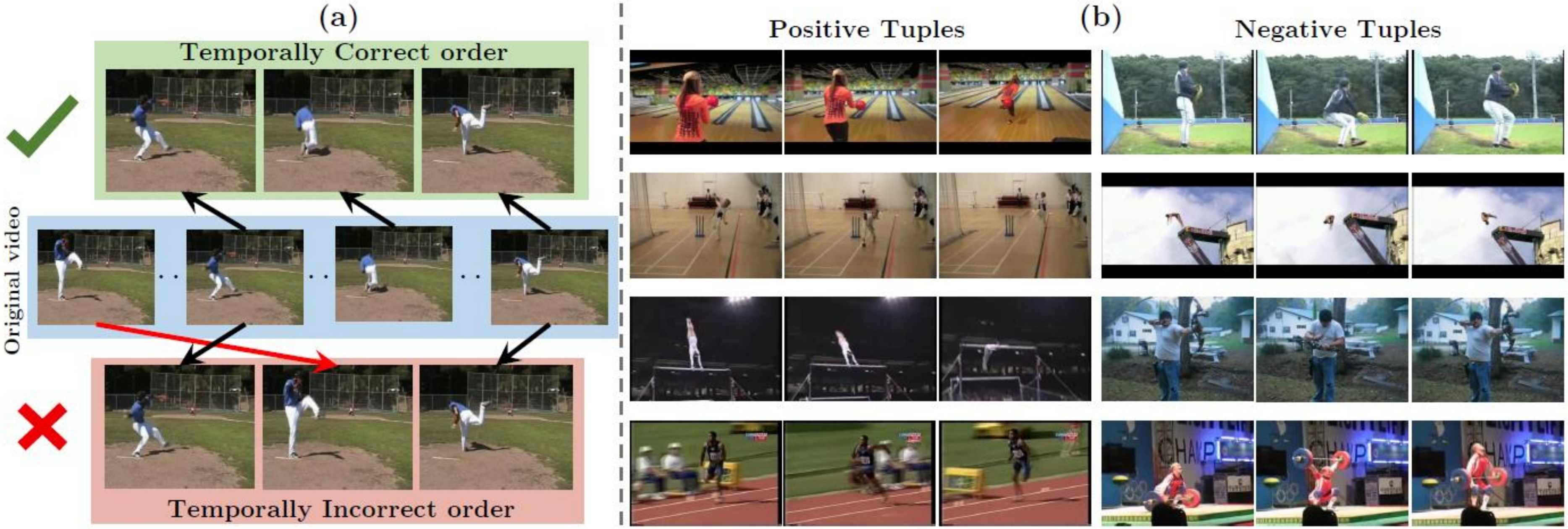


Fig. 1: **(a)** A video imposes a natural temporal structure for visual data. In many cases, one can easily verify whether frames are in the correct temporal order (shuffled or not). Such a simple sequential verification task captures important spatiotemporal signals in videos. We use this task for unsupervised pre-training of a Convolutional Neural Network (CNN). **(b)** Some examples of the automatically extracted positive and negative tuples used to formulate a classification task for a CNN.

Misra et al., "Shuffle and Learn: Unsupervised Learning using Temporal Order Verification", ECCV 2016

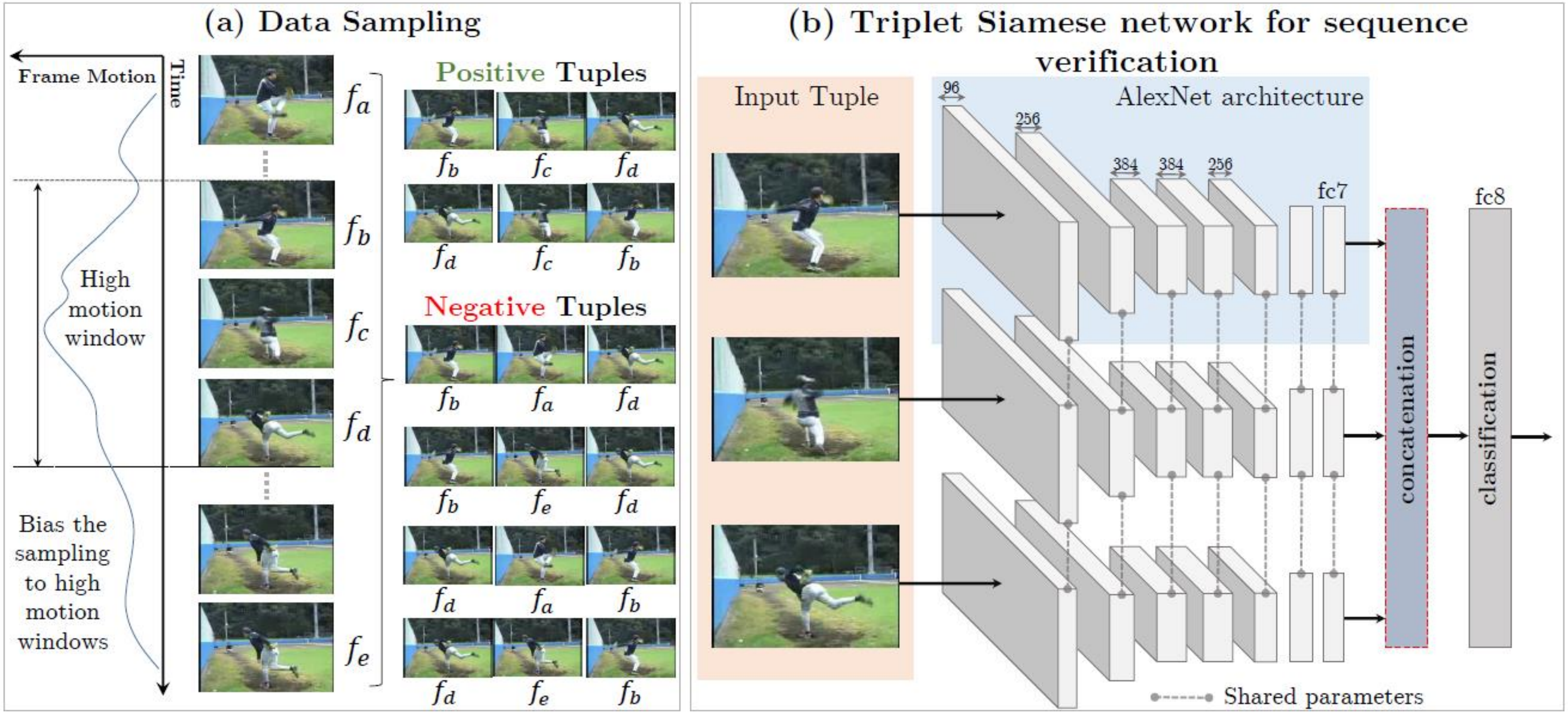


Fig. 2: **(a)** We sample tuples of frames from high motion windows in a video. We form positive and negative tuples based on whether the three input frames are in the correct temporal order. **(b)** Our triplet Siamese network architecture has three parallel network stacks with shared weights upto the `fc7` layer. Each stack takes a frame as input, and produces a representation at the `fc7` layer. The concatenated `fc7` representations are used to predict whether the input tuple is in the correct temporal order.

Misra et al., "Shuffle and Learn: Unsupervised Learning using Temporal Order Verification", ECCV 2016

Table 2: Mean classification accuracies over the 3 splits of UCF101 and HMDB51 datasets. We compare different initializations and finetune them for action recognition.

Dataset	Initialization	Mean Accuracy
UCF101	Random	38.6
	(Ours) Tuple verification	**50.2**
HMDB51	Random	13.3
	UCF Supervised	15.2
	(Ours) Tuple verification	**18.1**

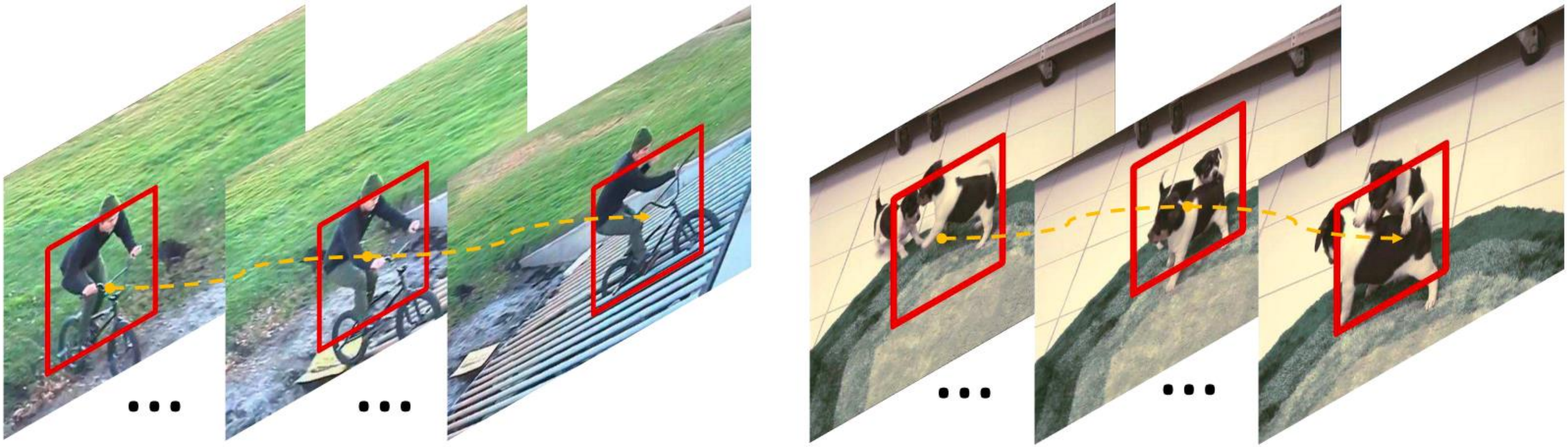
Misra et al., "Shuffle and Learn: Unsupervised Learning using Temporal Order Verification", ECCV 2016

Unsupervised Learning of Visual Representations using Videos

Xiaolong Wang, and Abhinav Gupta

ICCV 2016

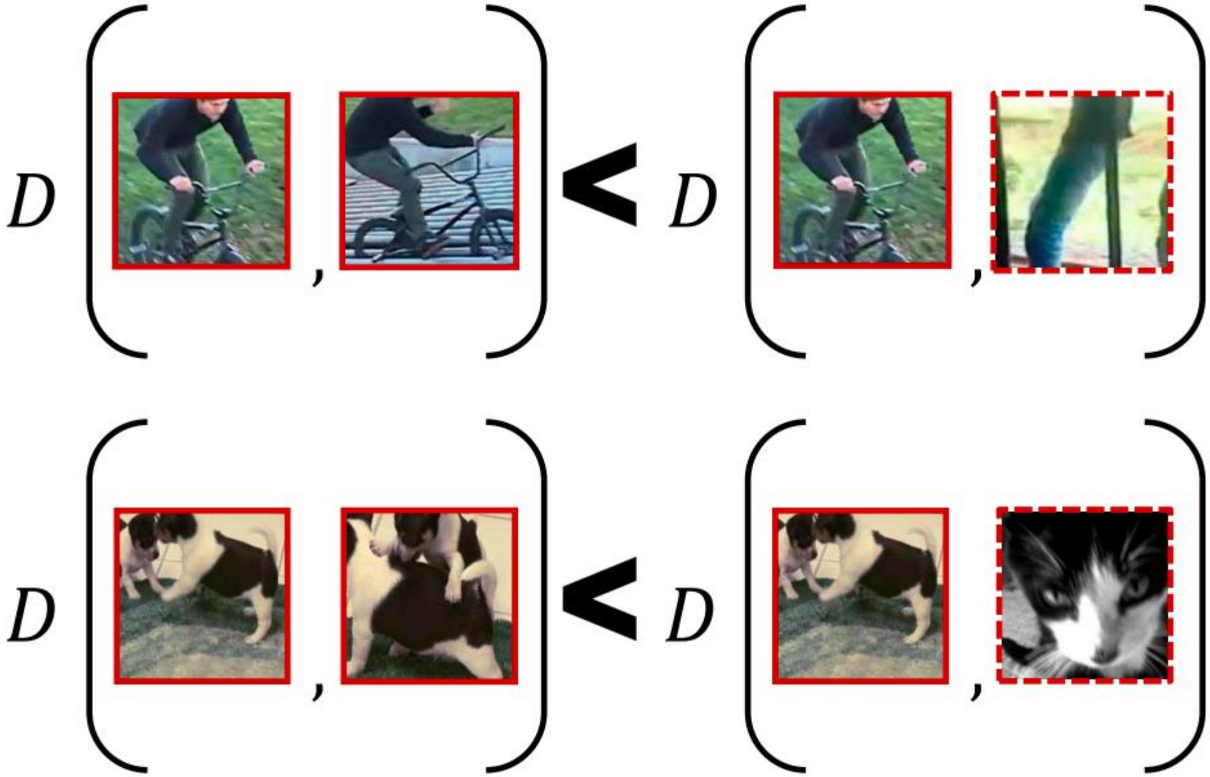
Visual Supervision from Tracking



The tracking target is the same instance in the video.

Wang and Gupta, "Unsupervised Learning of Visual Representations using Videos", ICCV 2015.

Distance in Deep Feature Space

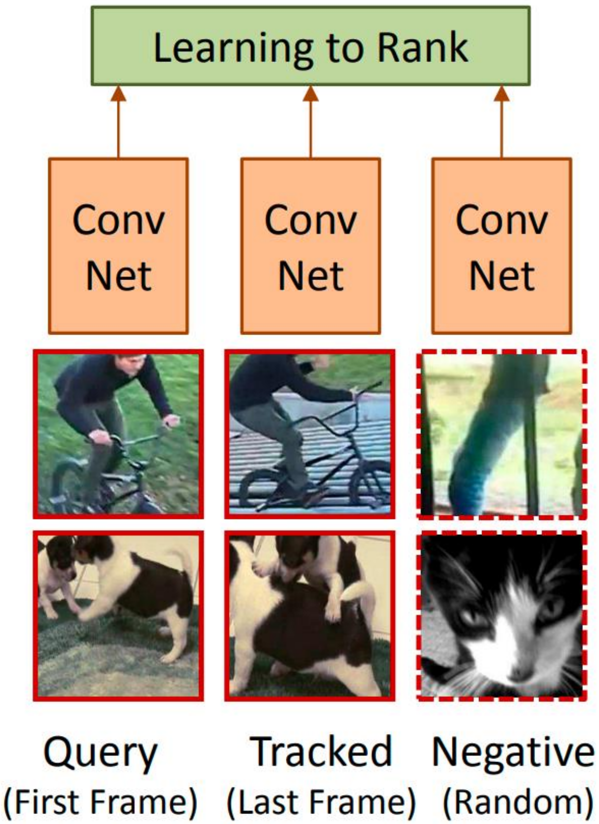


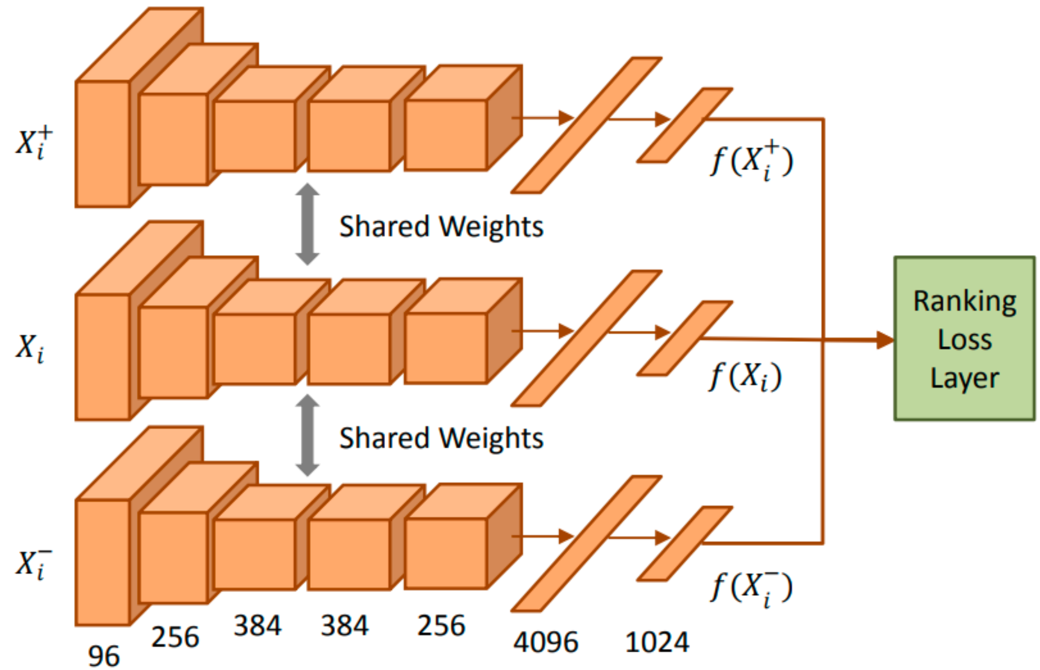
Same target

Target and background

Wang and Gupta, “Unsupervised Learning of Visual Representations using Videos”, ICCV 2015.

Learning to Predict Ranking





$$L(X_i, X_i^+, X_i^-) = \max\{0, D(X_i, X_i^+) - D(X_i, X_i^-) + M\}$$

$$D(X_1, X_2) = 1 - \frac{f(X_1) \cdot f(X_2)}{\|f(X_1)\| \|f(X_2)\|}$$

Wang and Gupta, "Unsupervised Learning of Visual Representations using Videos", ICCV 2015.

Results

Wang and Gupta, "Unsupervised Learning of Visual Representations using Videos", ICCV 2015.