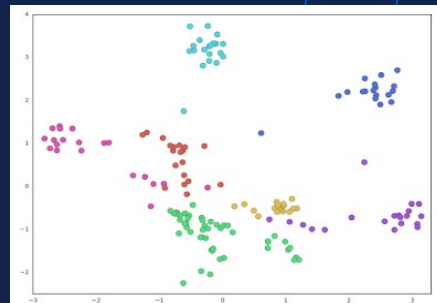


Predictive and non-contrastive self-supervised representation learning for graphs



Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, Michal Valko:
Bootstrapped representation learning on graphs

Why we are after self-supervised learning?



“For AGI we want agents to generalise significantly beyond the specific tasks that they were trained on. “



Reality check = **very limited supervision**

... but supervised learning is what ML is good at!



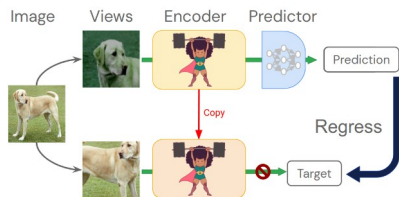
Mastering SSL we equip agents with stronger generalization capabilities.



1

SSL

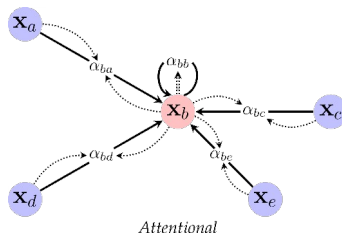
- (Modern) SSL
- How BYOL works
- ResNet as *encoder*
- ImageNet as *data*



2

Graph Nets

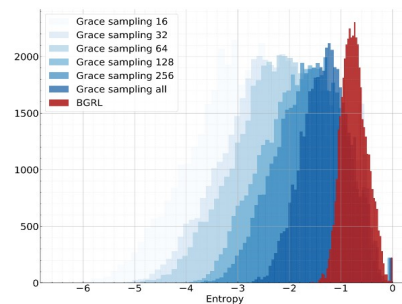
Graph Nets as the *encoders* for graph data



3

BGRL

Self-Supervised Learning on Graphs



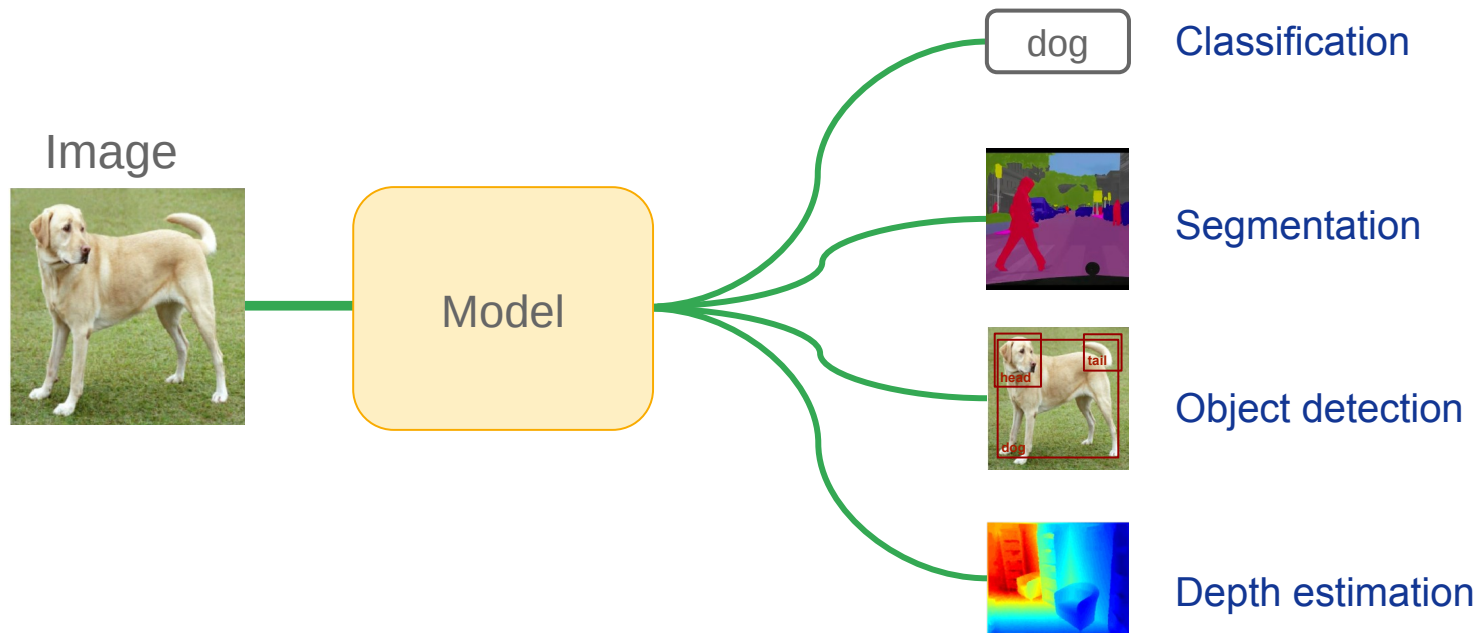


1

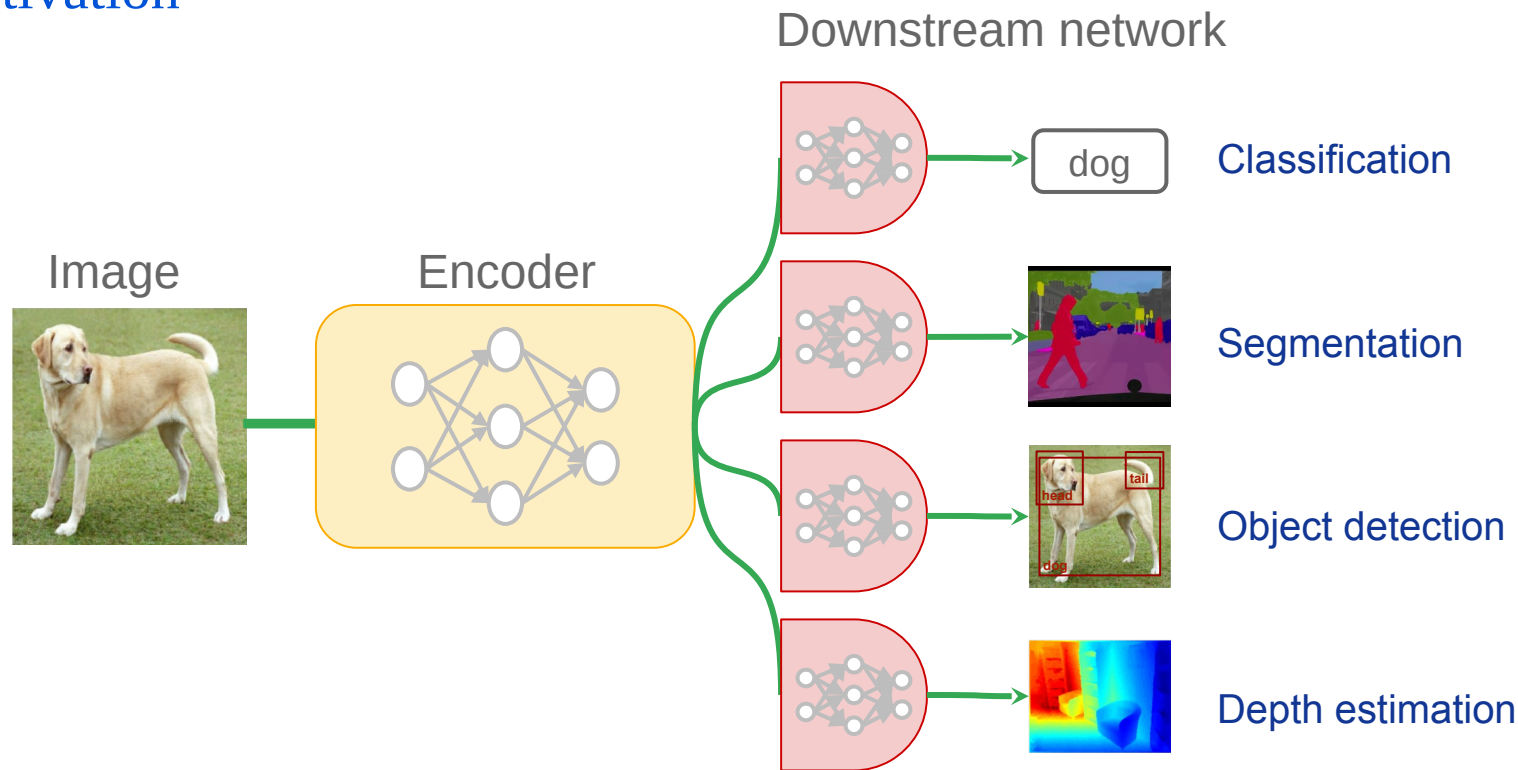
Self-Supervised Learning



Computer Vision Goal



Motivation

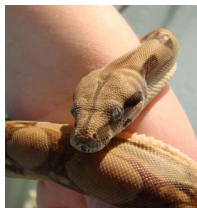


How to train the encoder?

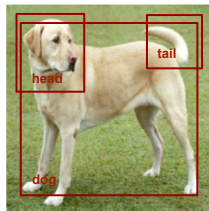
Motivation



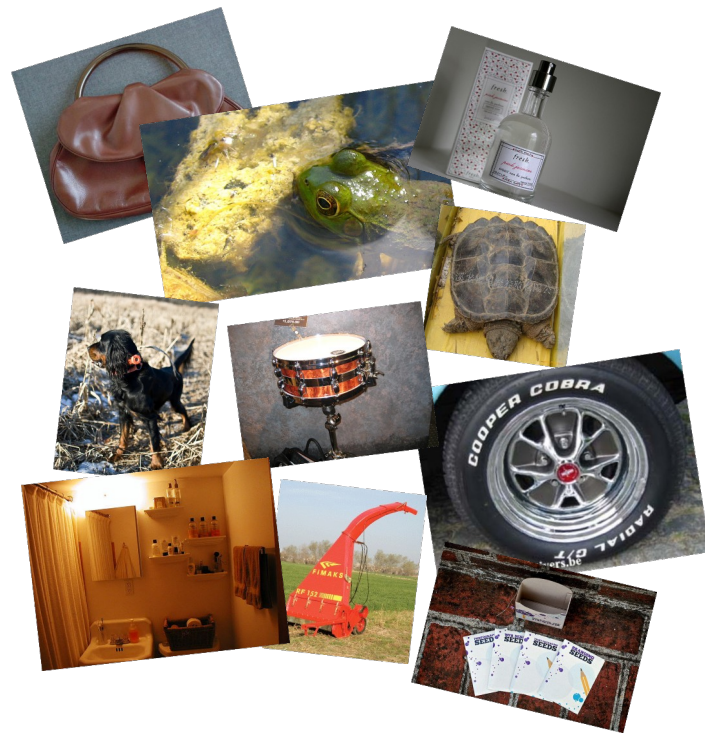
Dog



Snake

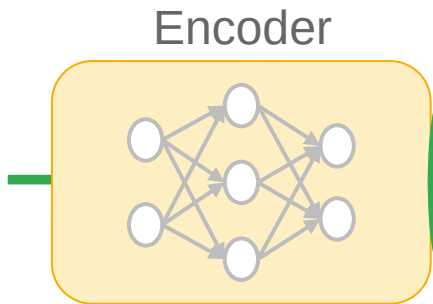


Labelled, but costly/few data

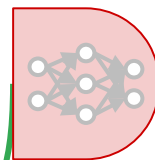


Unlabelled, **free** data!

Motivation

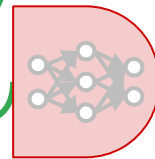


Downstream network

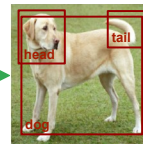
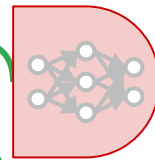


dog

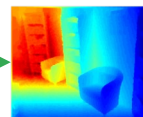
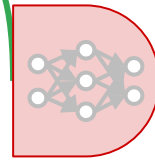
Classification



Segmentation



Object detection



Depth estimation

BYO



Self-supervised
Free unlabeled data

Supervised
Few labeled data

2

BYOL

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond
Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheslaghi
Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, Michal Valko

Intuition: Two different views (augmentations) of the same picture should be predictive of each other.

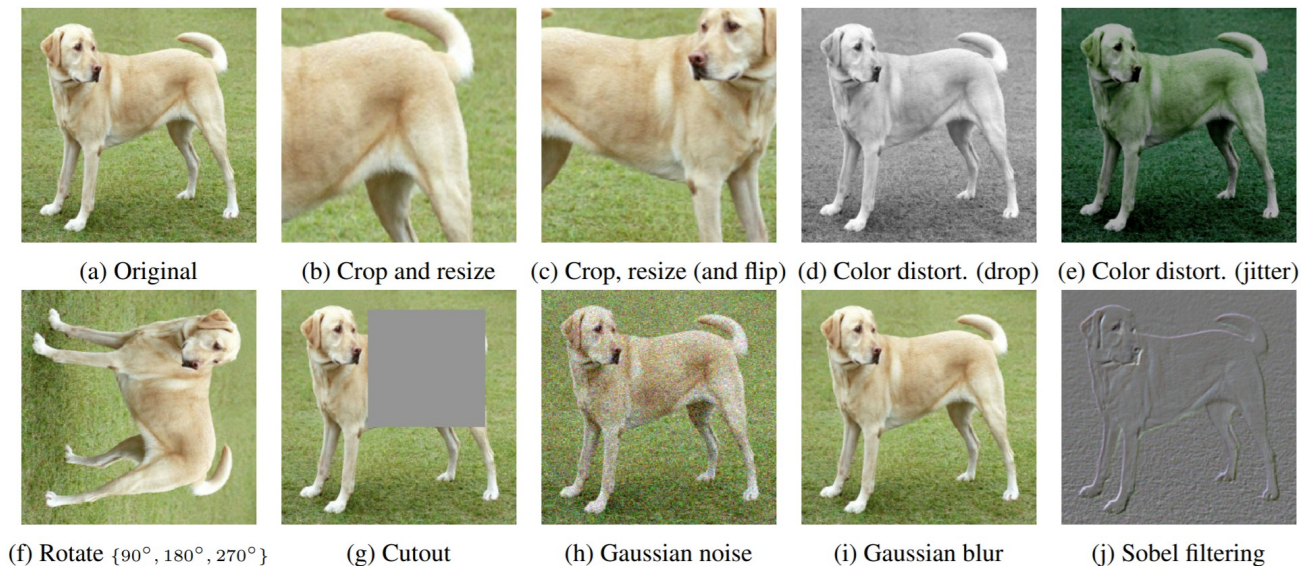


Figure from SimCLR¹

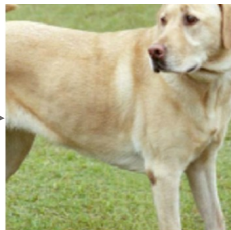
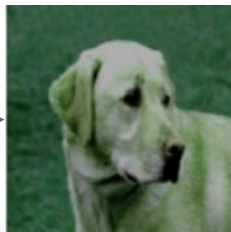
A view of a dog is still a dog, i.e. semantic information is **invariant** to transformations.

¹ SimCLR: Chen et al., A simple framework for contrastive learning of visual representations. ICML. 2020

BYOL main intuition

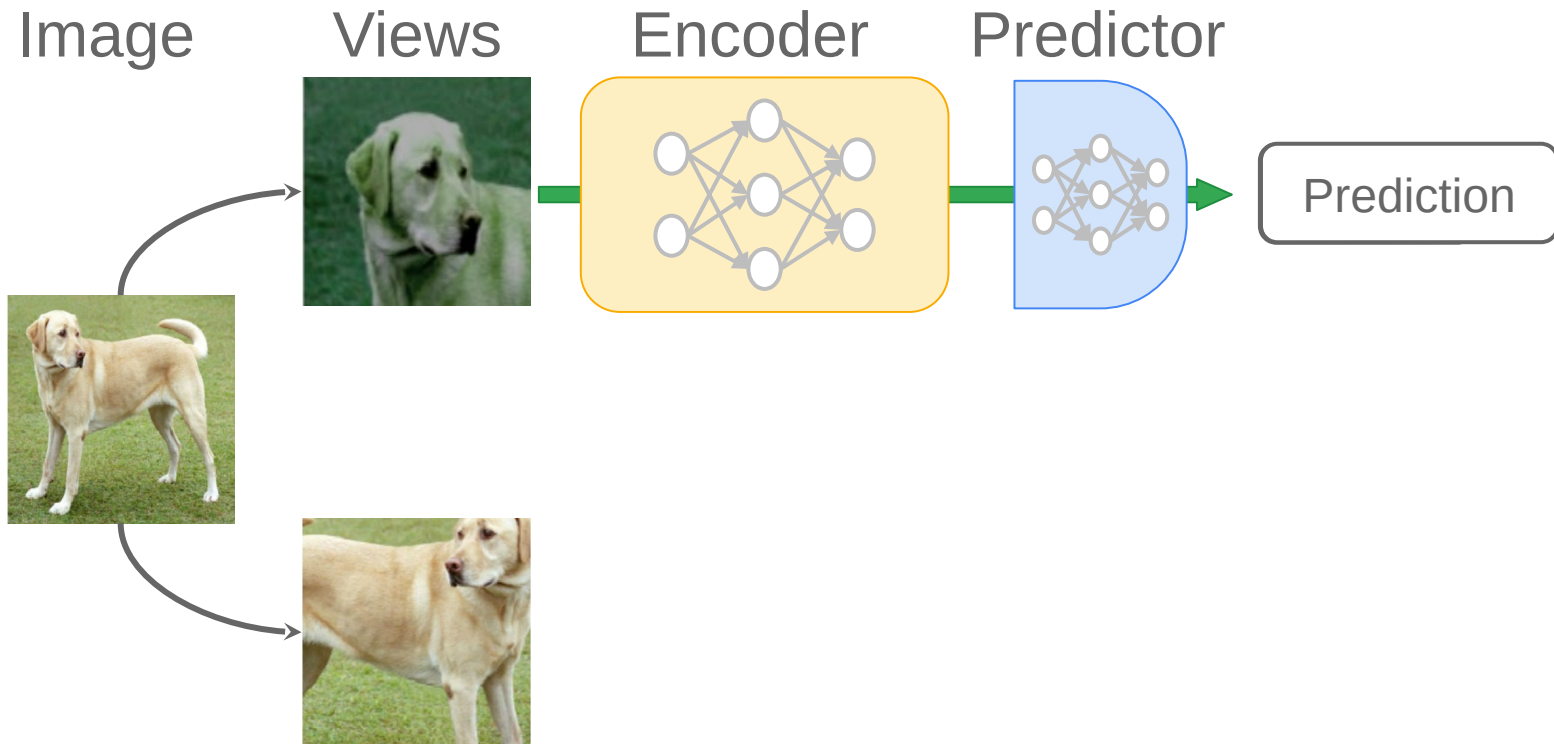
Image

Views

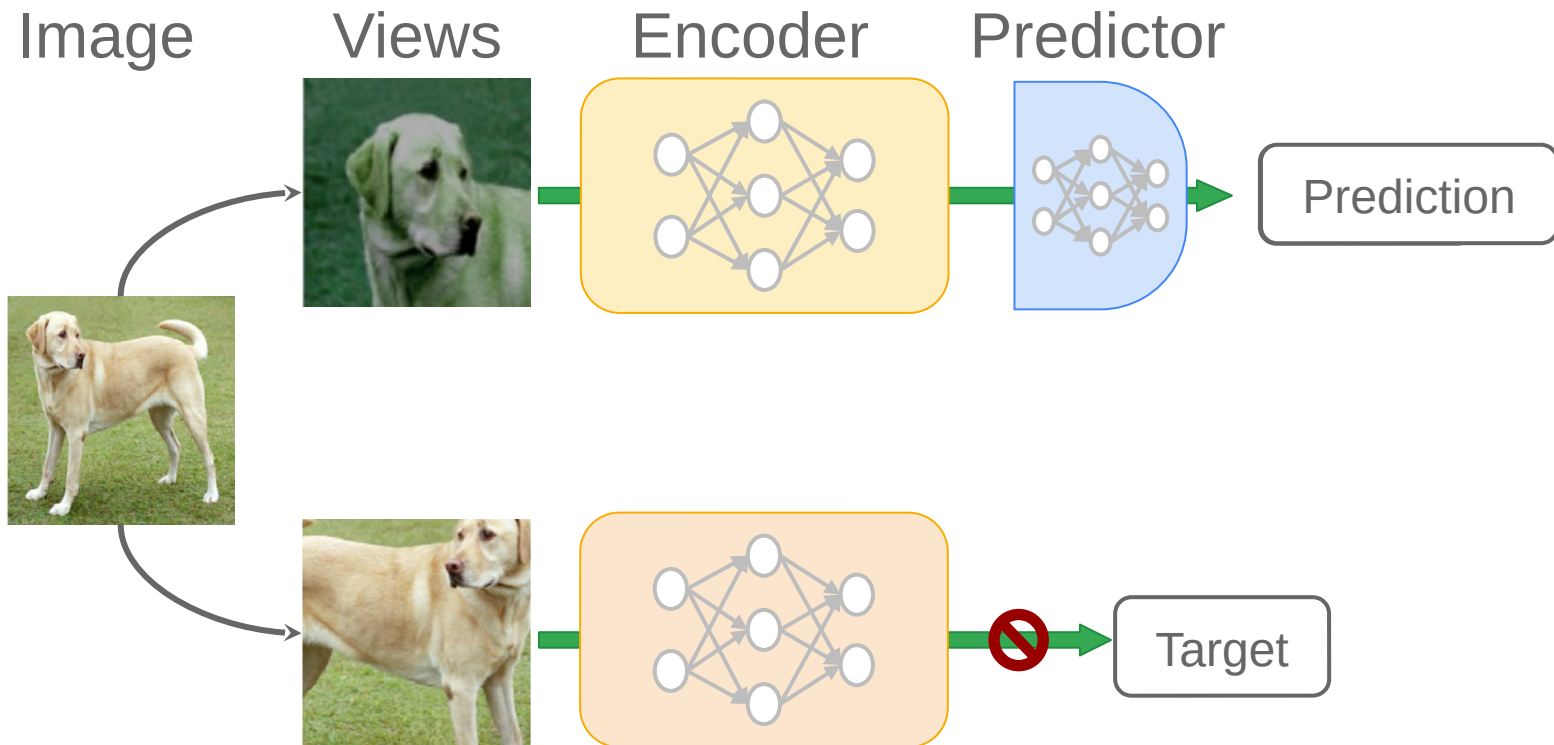


Predict?

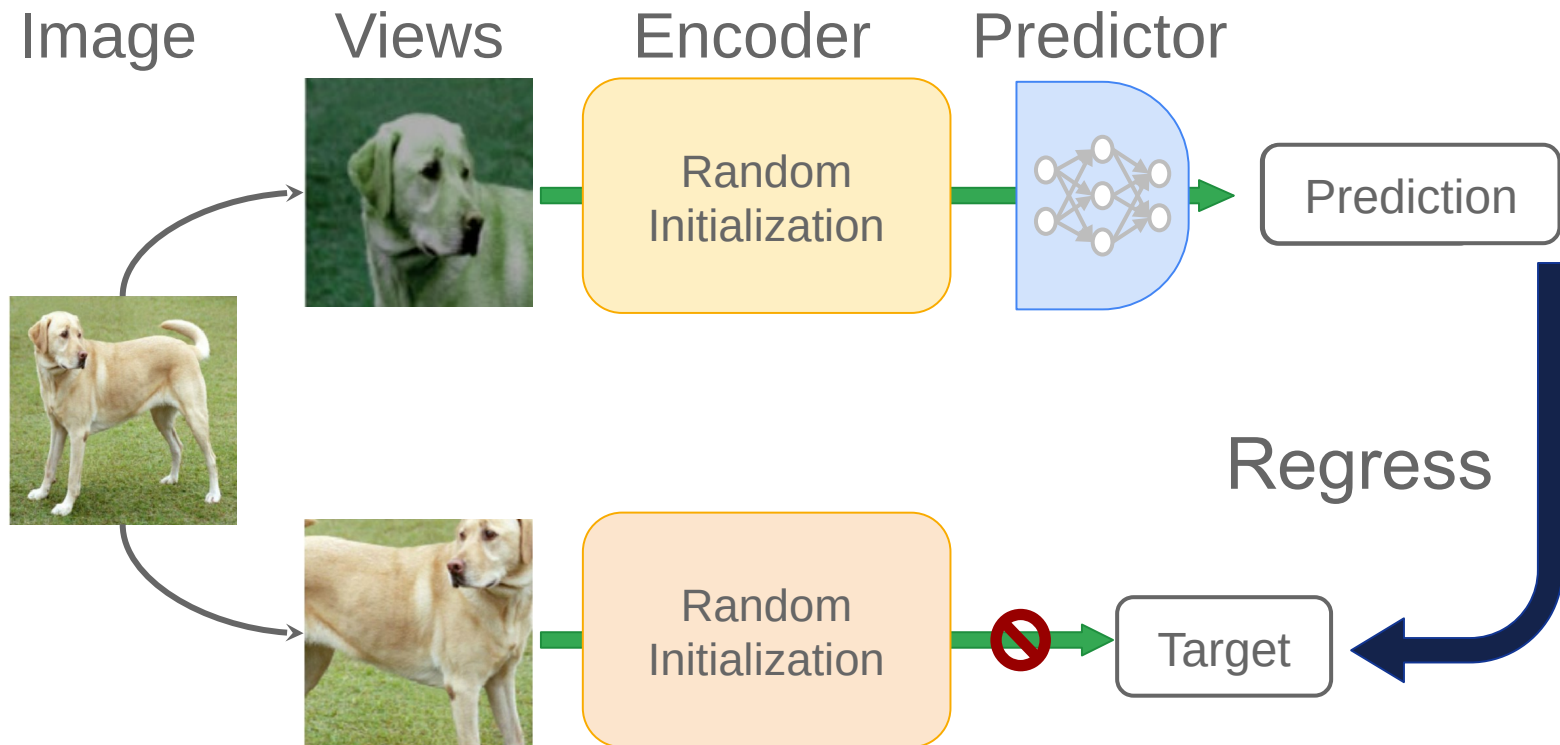
BYOL main intuition



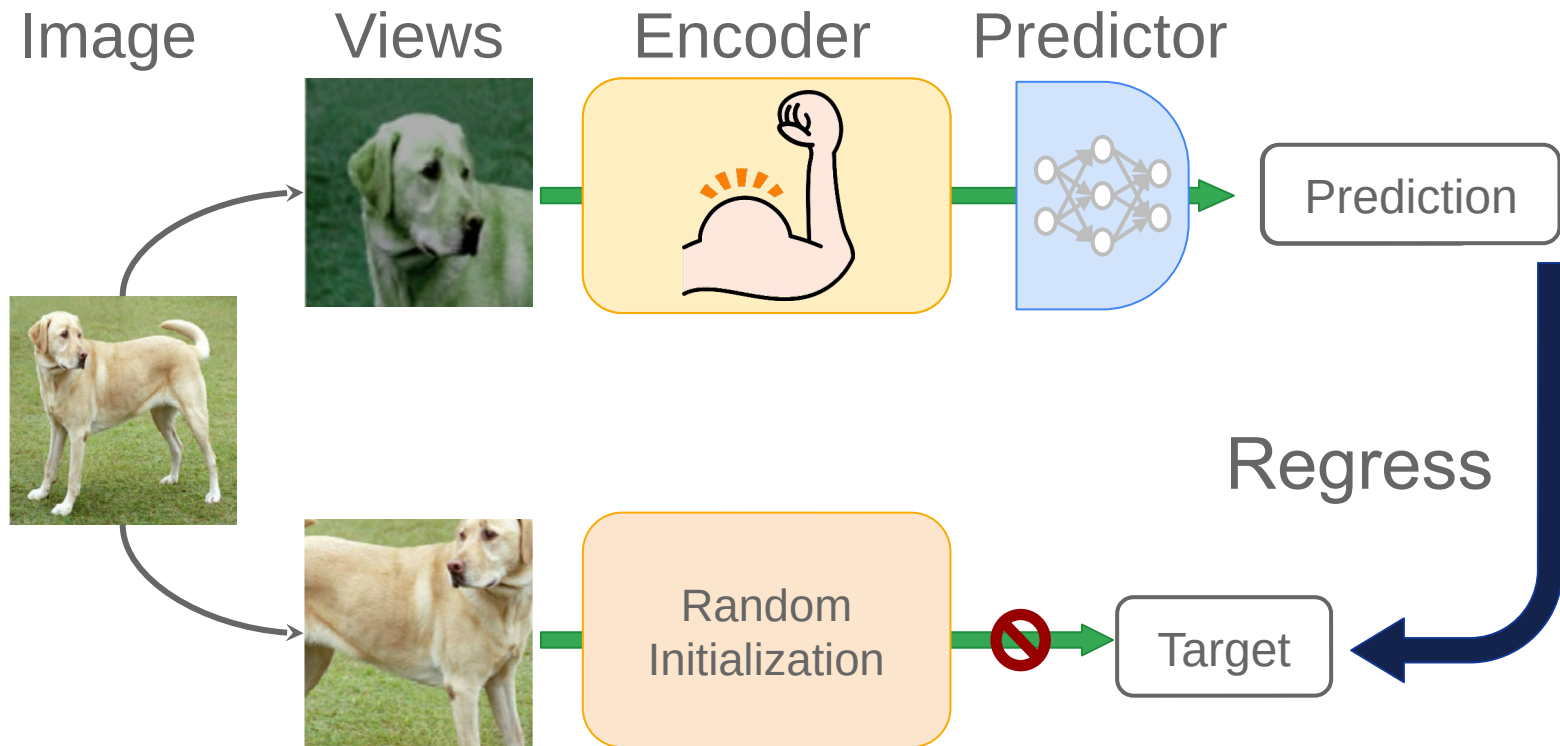
BYOL main intuition



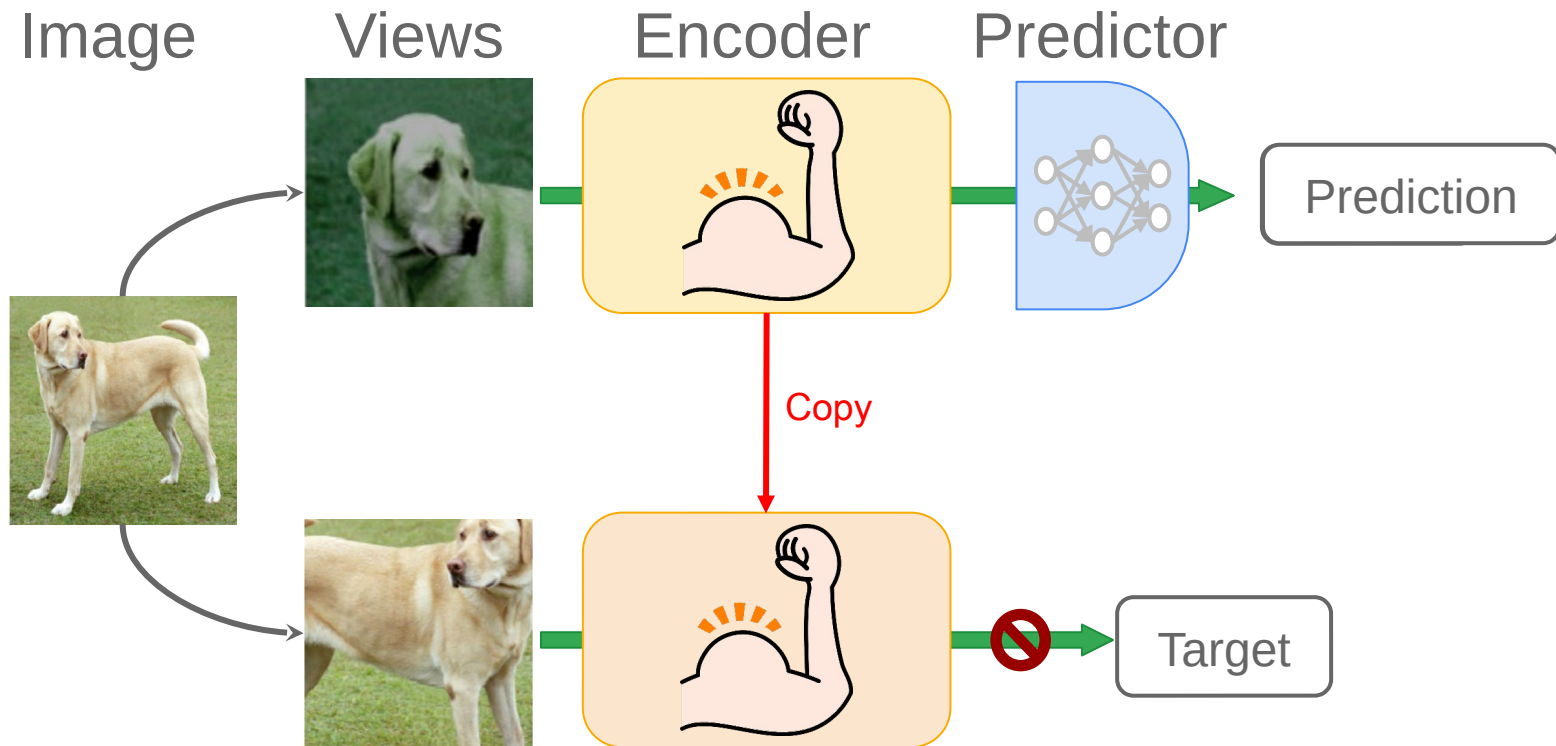
BYOL main intuition



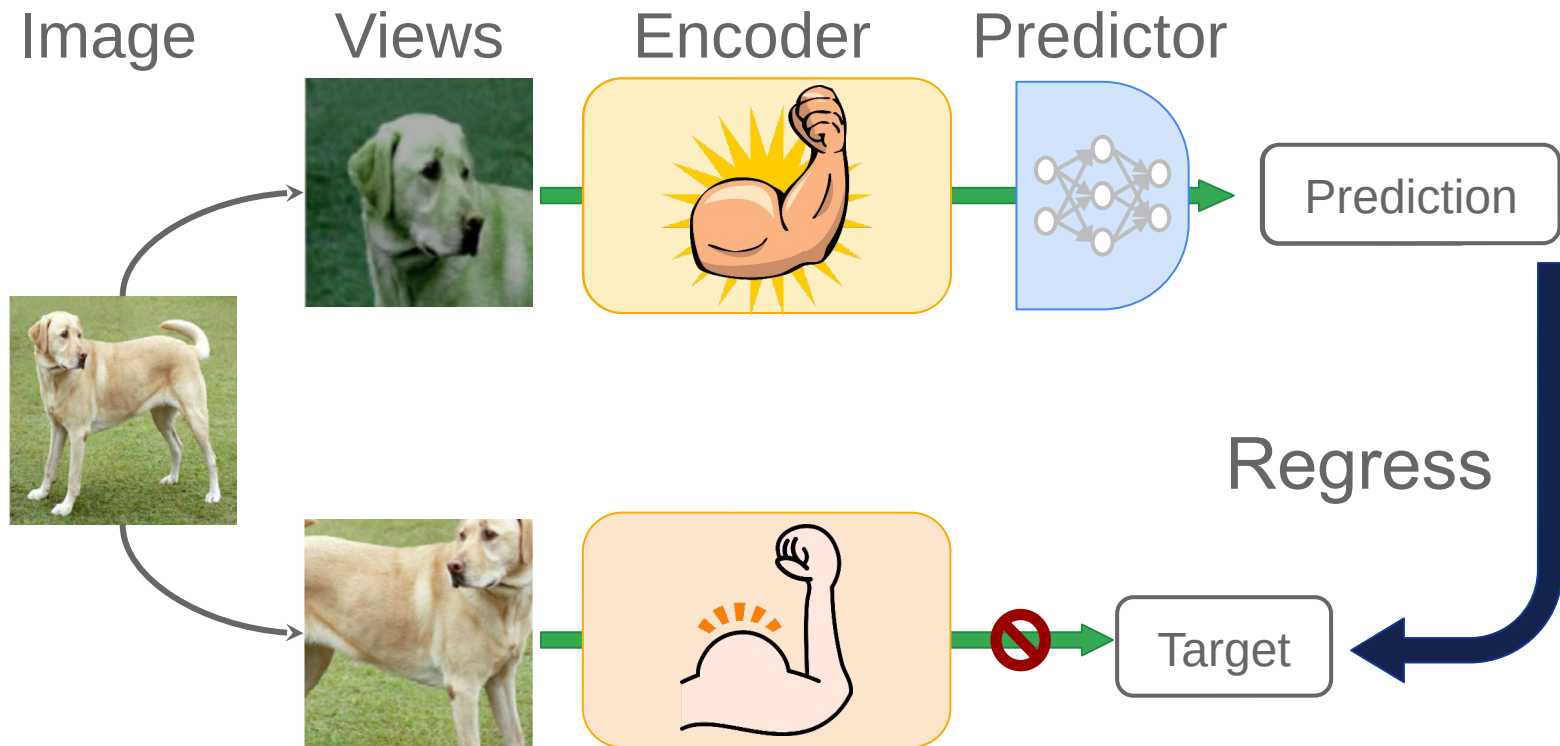
BYOL main intuition



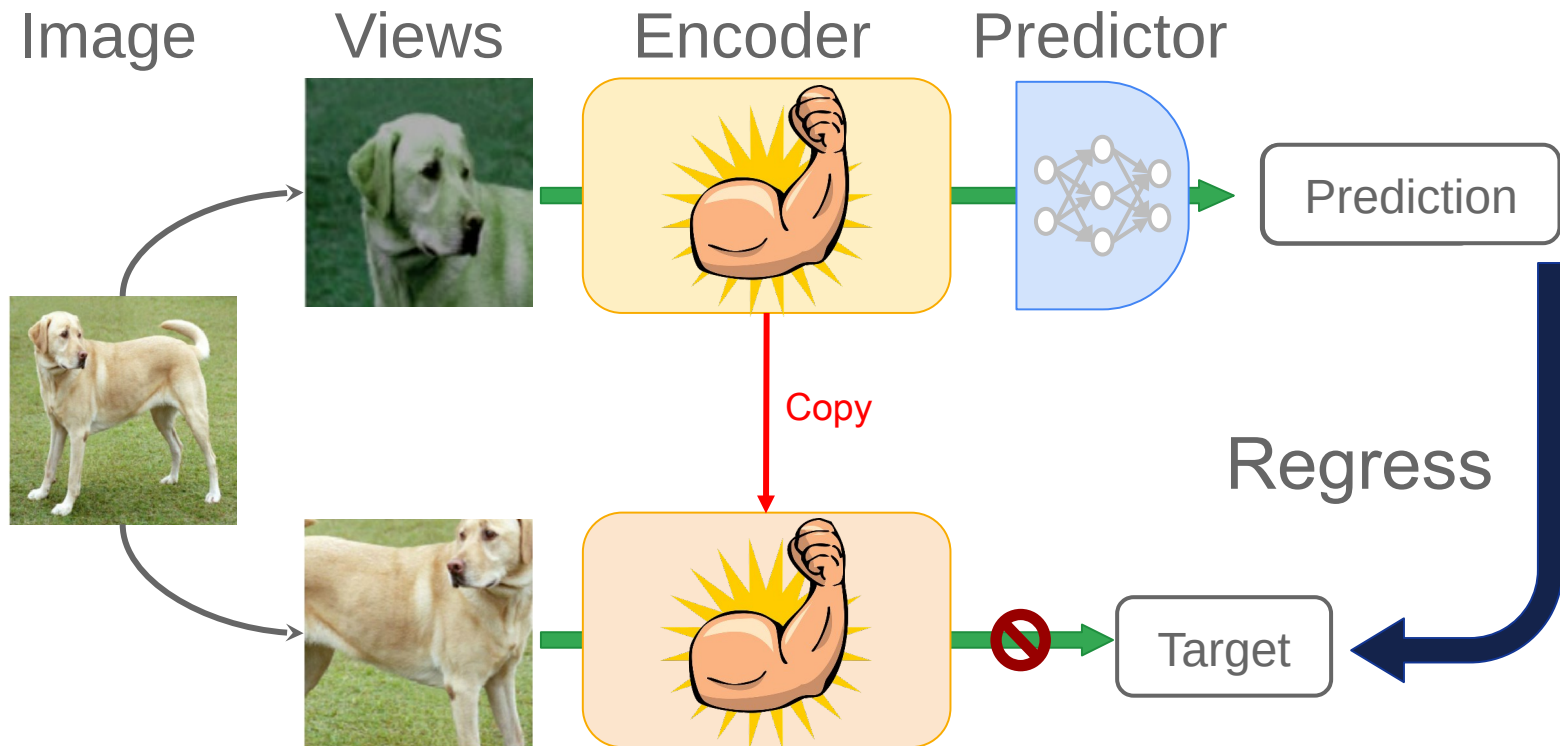
BYOL main intuition



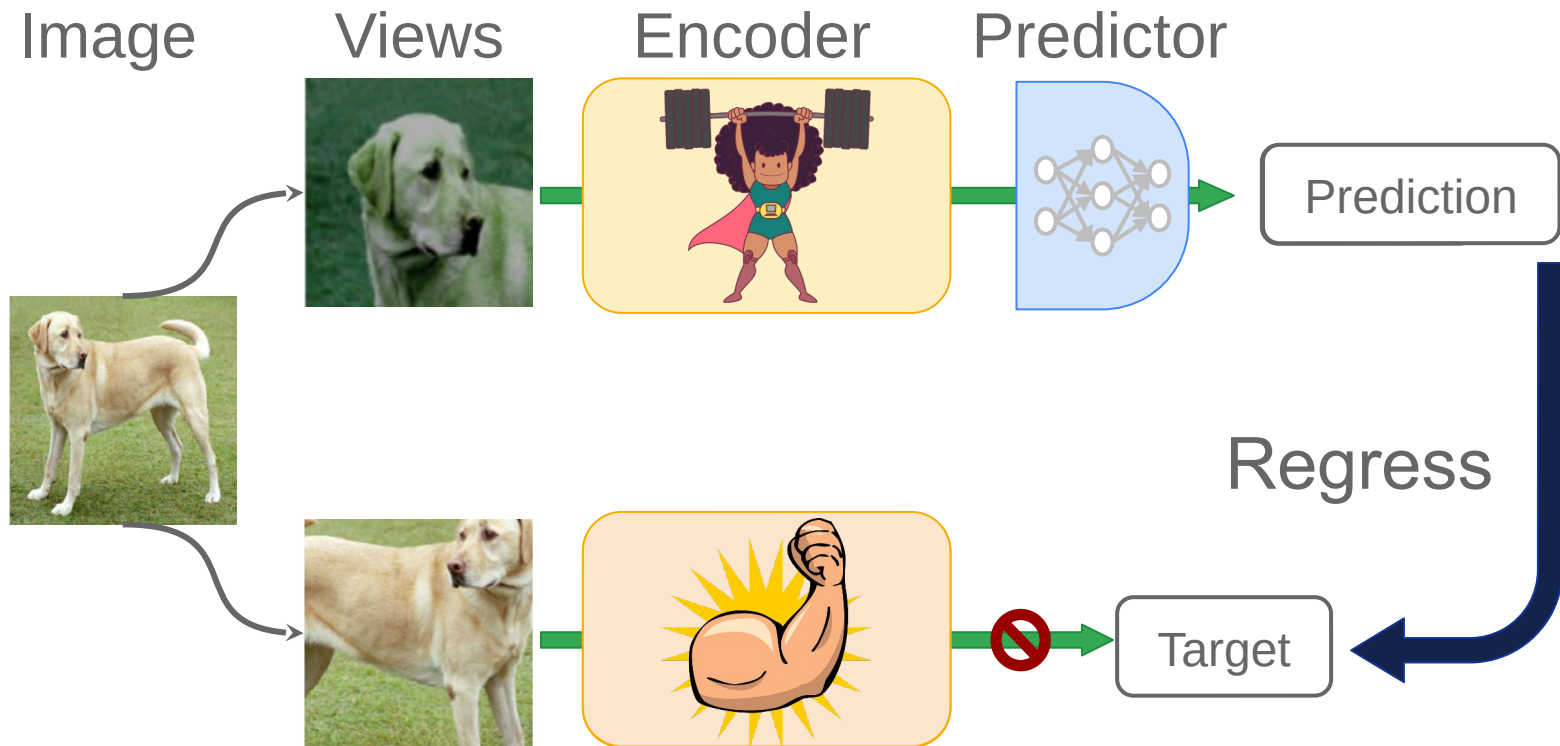
BYOL main intuition



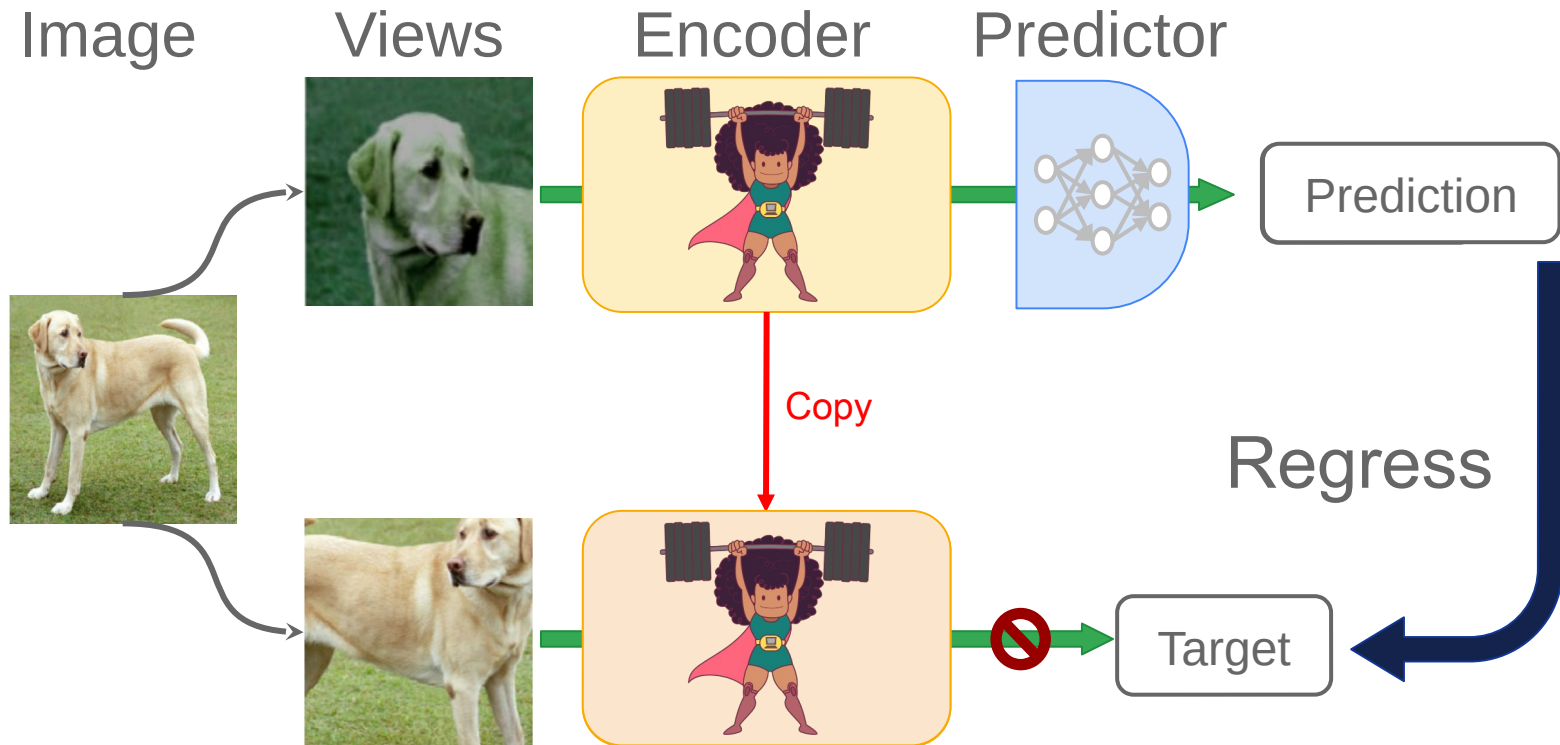
BYOL main intuition



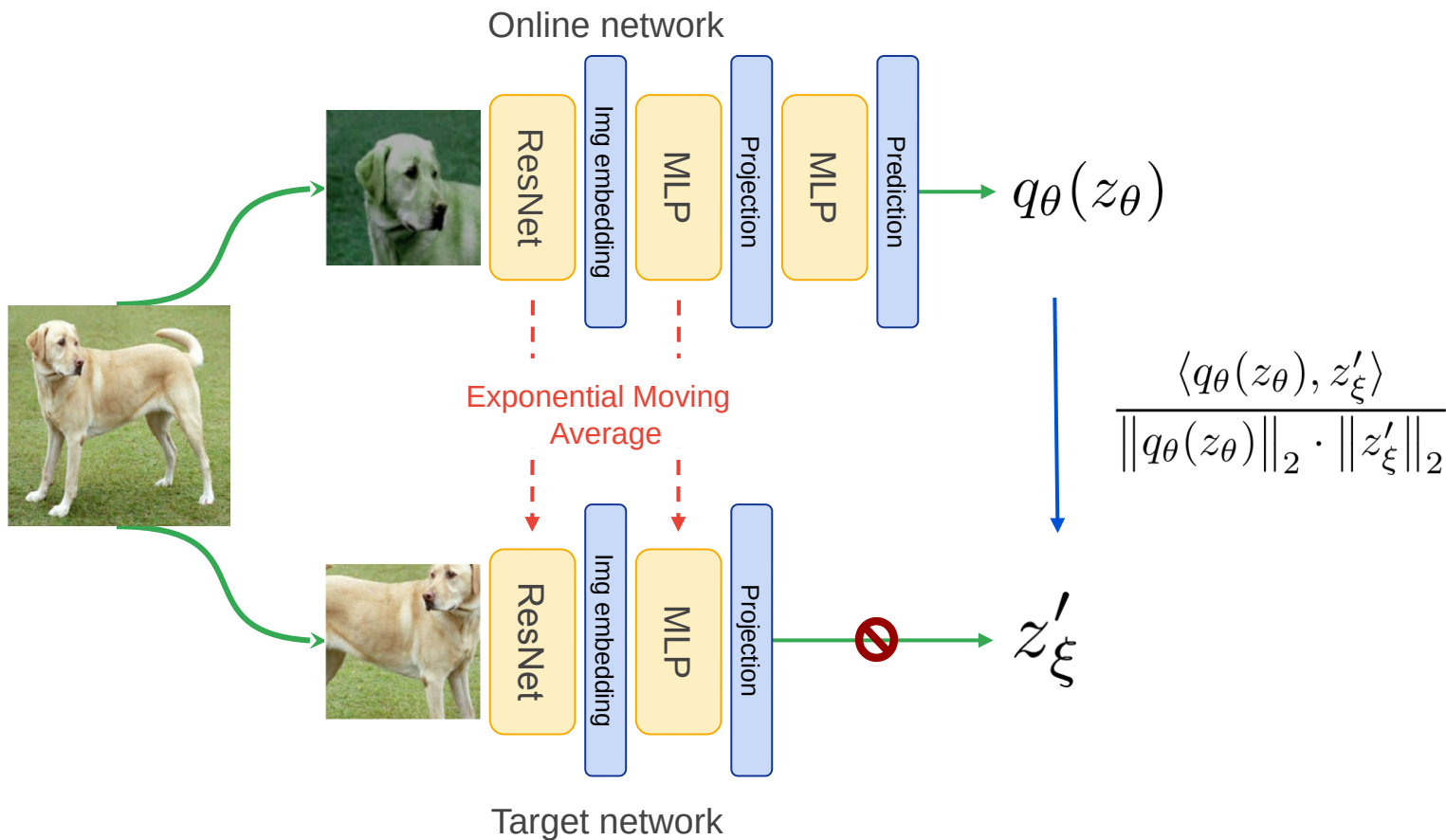
BYOL main intuition



BYOL main intuition



BYOL Architecture



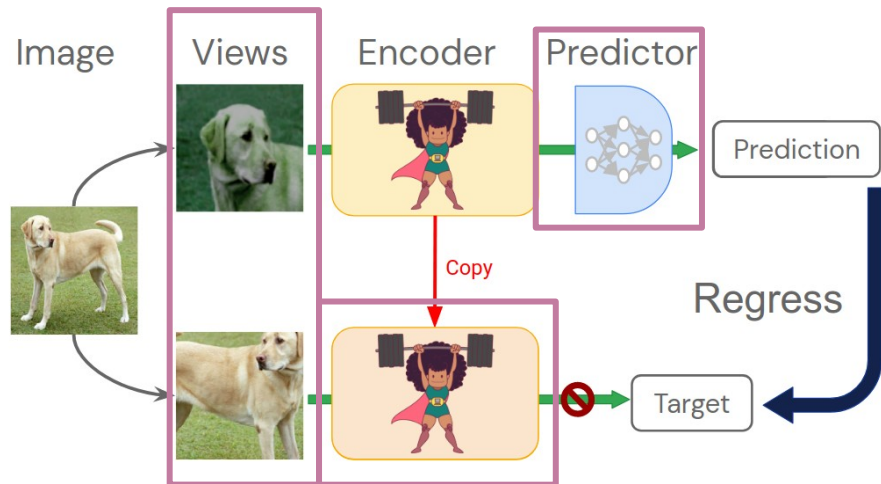
BYOL's highlights

Key ingredients:

- Image transformations.
- Target network.
- Additional predictor on top of online network.

Interest of the method:

- Simple training procedure.
- No negative examples [details 3 slides later].
- Work at the embedding level, e.g. no-pseudo labels.





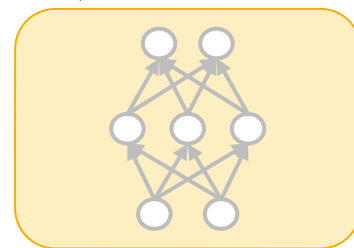
3

Performance
of BYOL



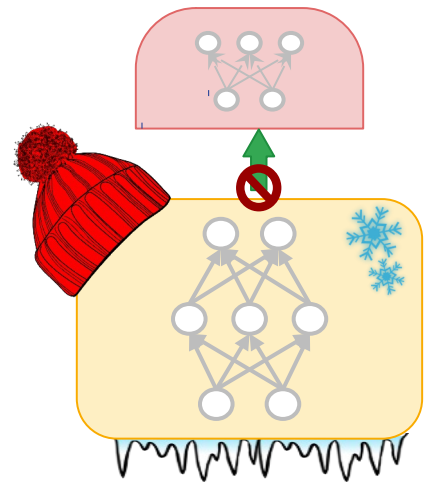
Linear Evaluation Protocol on ImageNet

Step 1: Train a “representation” on ImageNet without any labels.



ResNet

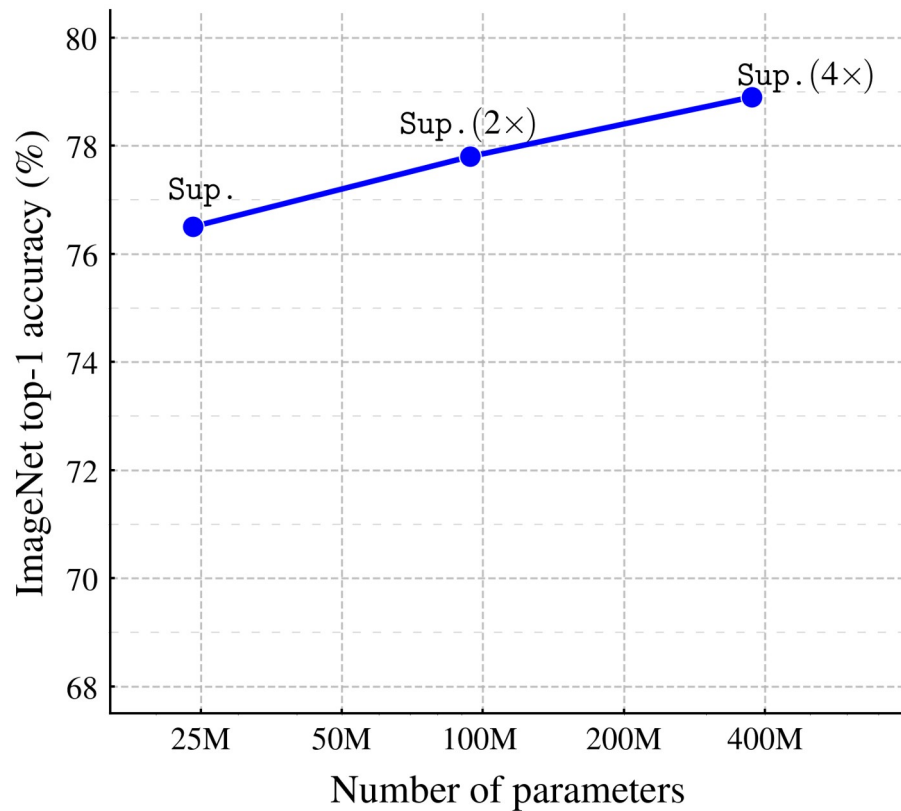
Step 2: On top of the **frozen representation, train a linear classifier on ImageNet with label information.**



Linear
Classifier

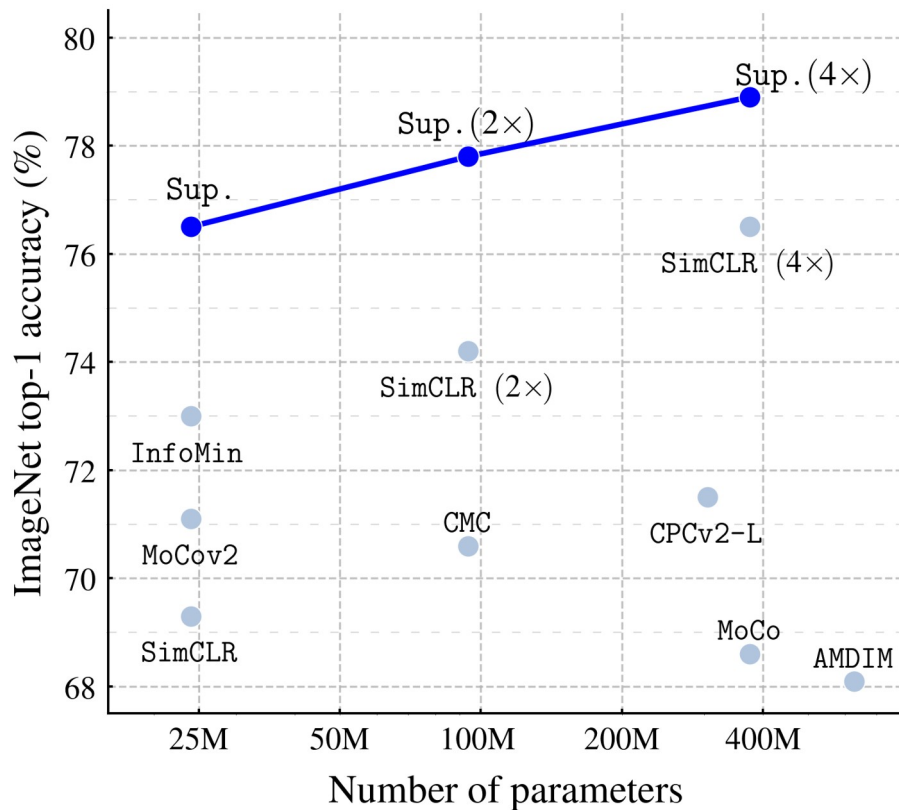
ResNet

Linear Evaluation Performance on ImageNet



Note: these supervised baselines are from SimCLR (Chen et al., ICML 2020)

Linear Evaluation Performance on ImageNet



Note: these supervised baselines are from SimCLR (Chen & Hinton, ICML 2020)

CPCv2: van den Oord et al., *Representation learning with contrastive predictive coding*. 2018

AMDIM: Bachman et al., *Learning representations by maximizing mutual information across views*. 2019

CMC: Tian et al., *Contrastive multiview coding*. 2019.

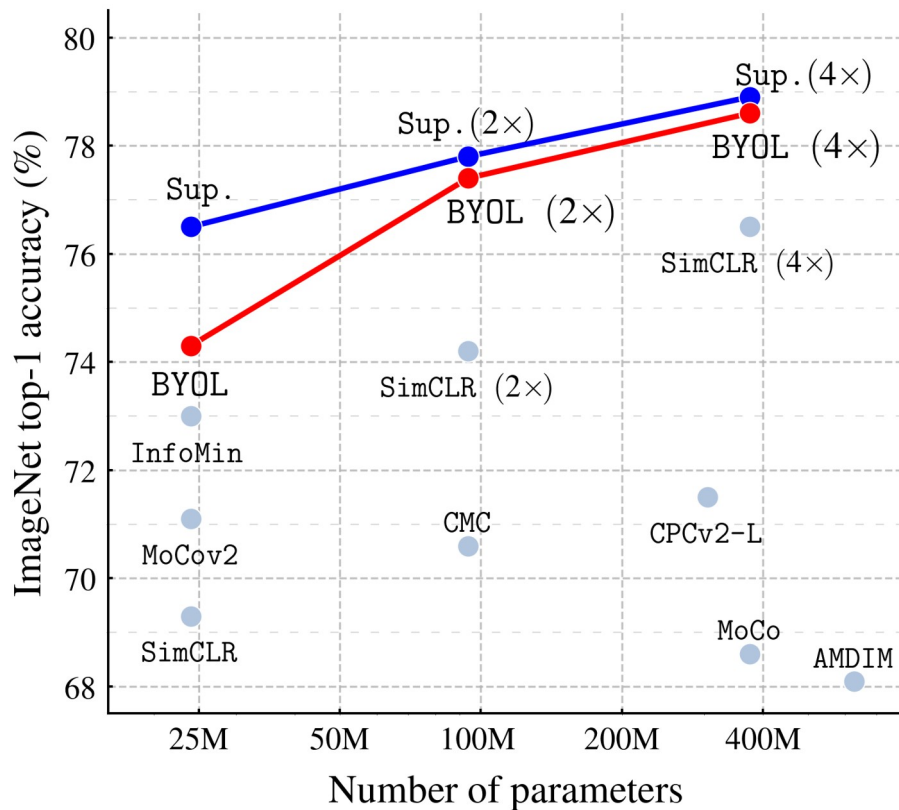
MoCo: He et al., *Momentum contrast for unsupervised visual representation learning*. 2019

InfoMin: Tian et al., *What makes for good views for contrastive learning*. 2020

MoCov2: Jain et al., *Improved baselines with momentum contrastive learning*. 2020

SimCLR: Chen et al., *A simple framework for contrastive learning of visual representations*. 2020

Linear Evaluation Performance on ImageNet



Note: these supervised baselines are from SimCLR (Chen & Hinton, ICML 2020)

CPCv2: van den Oord et al., *Representation learning with contrastive predictive coding*. 2018

AMDIM: Bachman et al., *Learning representations by maximizing mutual information across views*. 2019

CMC: Tian et al., *Contrastive multiview coding*. 2019.

MoCo: He et al., *Momentum contrast for unsupervised visual representation learning*. 2019

InfoMin: Tian et al., *What makes for good views for contrastive learning*. 2020

MoCov2: Jain et al., *Improved baselines with momentum contrastive learning*. 2020

SimCLR: Chen et al., *A simple framework for contrastive learning of visual representations*. 2020

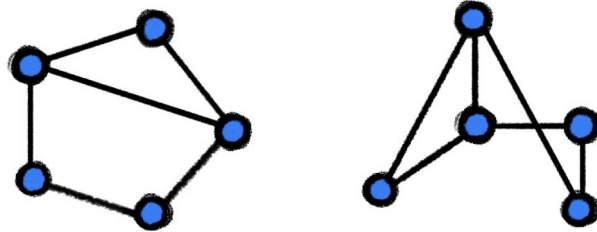
Self-Supervised Learning on Graphs

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou,
Eva L Dyer, Remi Munos, Petar Veličković, Michal Valko

Thanks to Petar Veličković for help with slides!

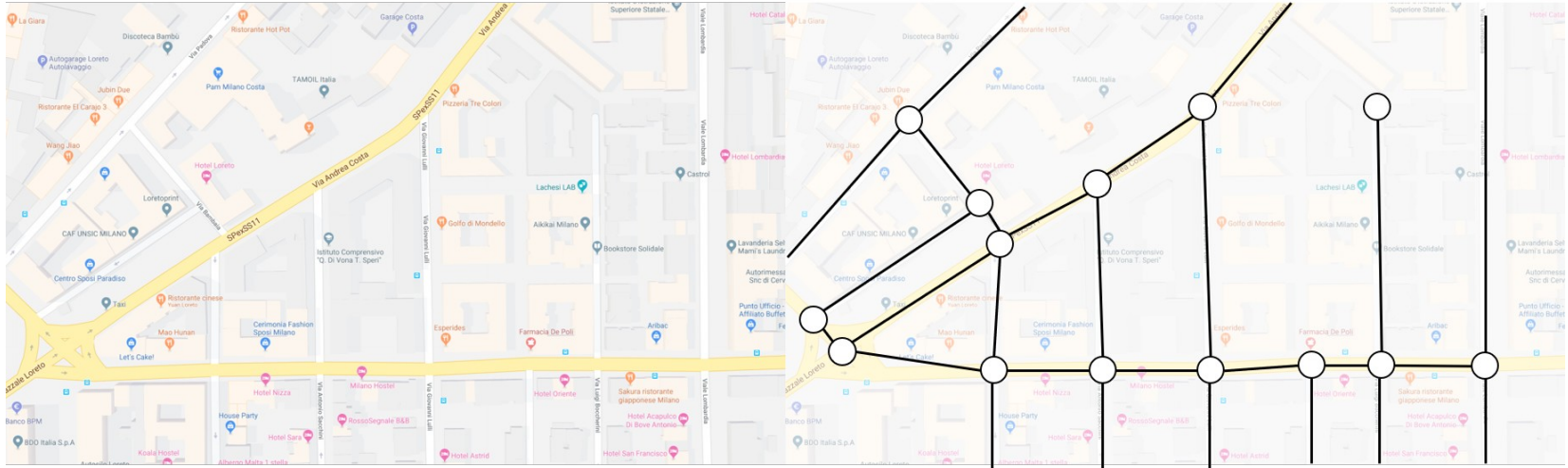
Graphs are Everywhere!

- Data with special structure:
 - Nodes = entities
 - Edges = connections between nodes
 - Graphs = collection of nodes with edges



Traffic maps are graphs!

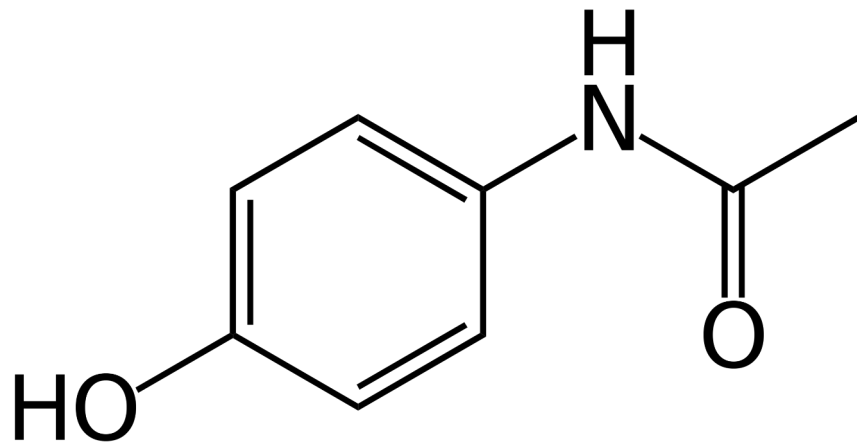
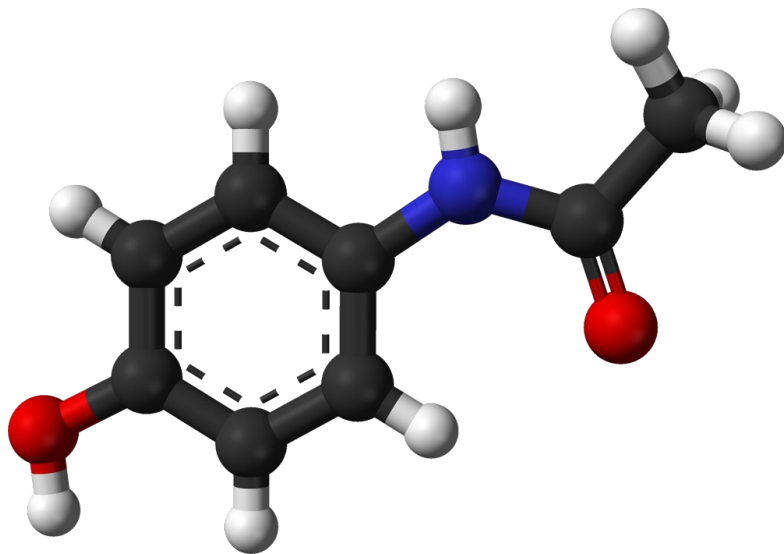
- Transportation networks (e.g. *Google Maps*) naturally modelled as



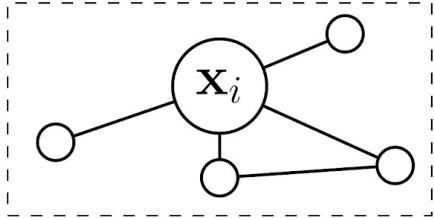
- Nodes as **intersections**, edges as **roads**
- Many natural node/edge-level **features** in this data!
- Possible task of interest: **ETA prediction**

Molecules are graphs!

- A very natural way to represent molecules
 - **Atoms** as nodes, **bonds** as edges
 - Features such as **atom type**, **charge**, **bond type**...
 - Possible task - predict whether molecule inhibits diseases

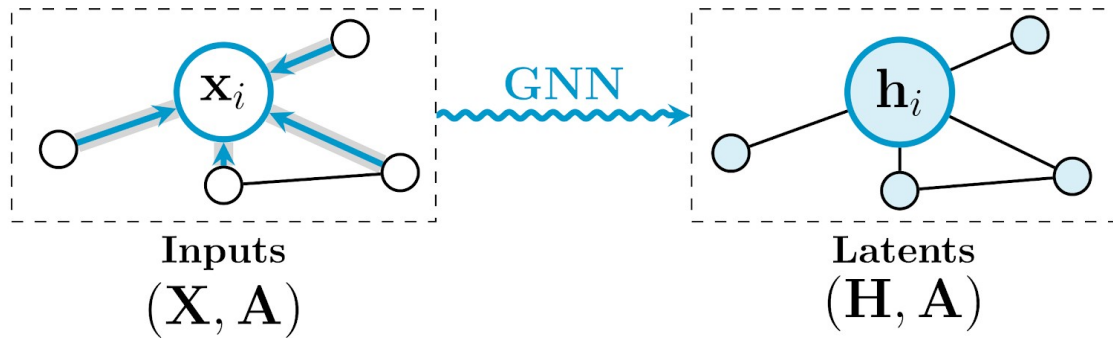


How to learn from graphs?

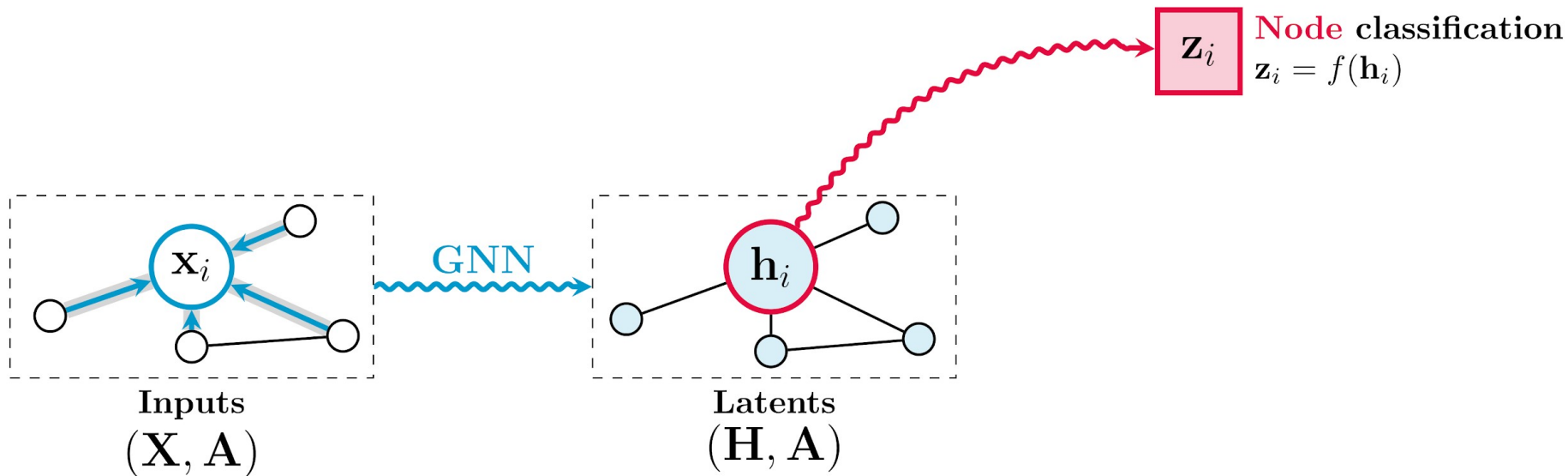


Inputs
(\mathbf{X}, \mathbf{A})

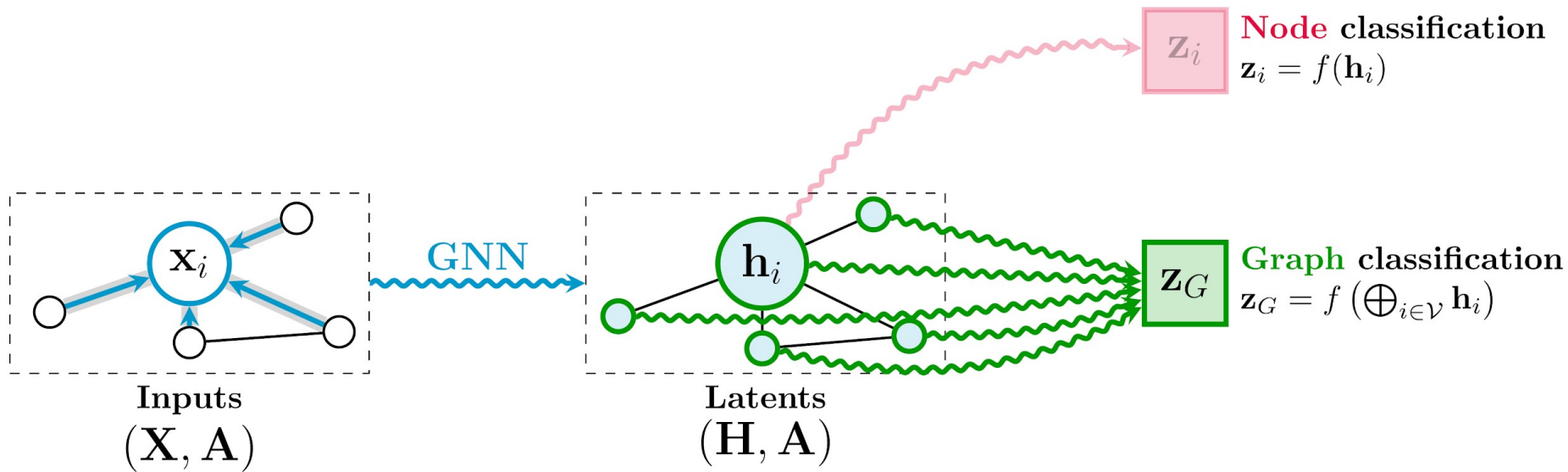
Graph Neural Networks!



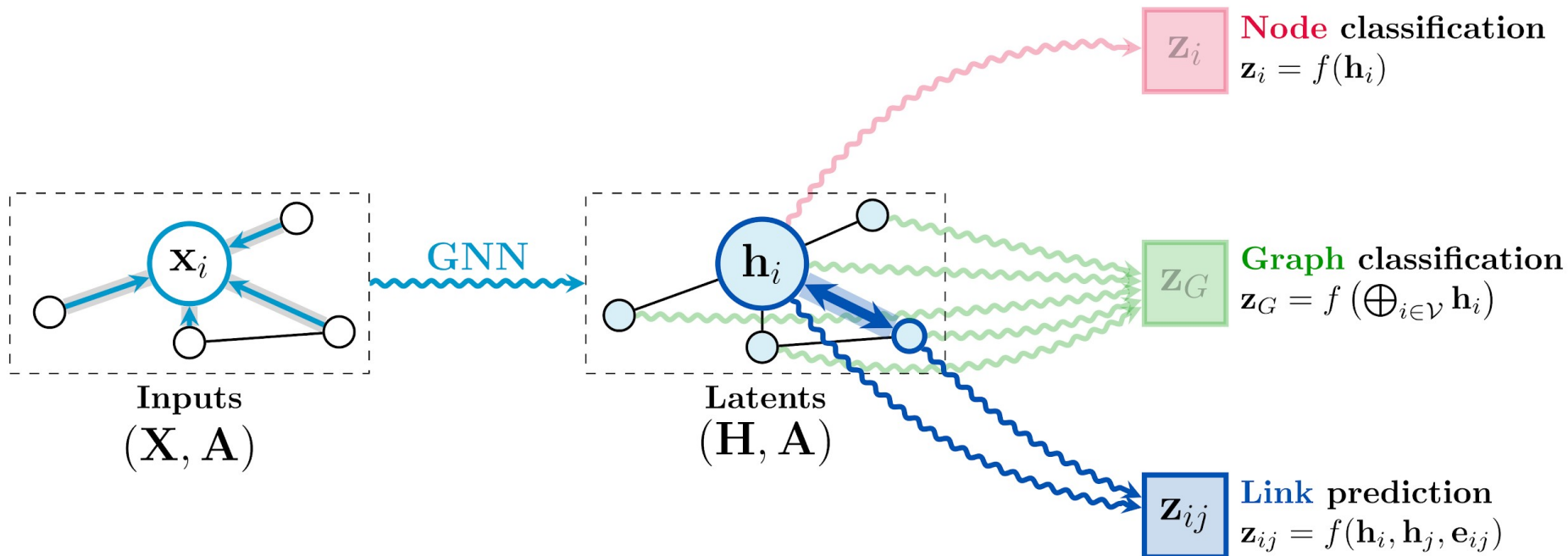
Node-level representations



Graph-level representations



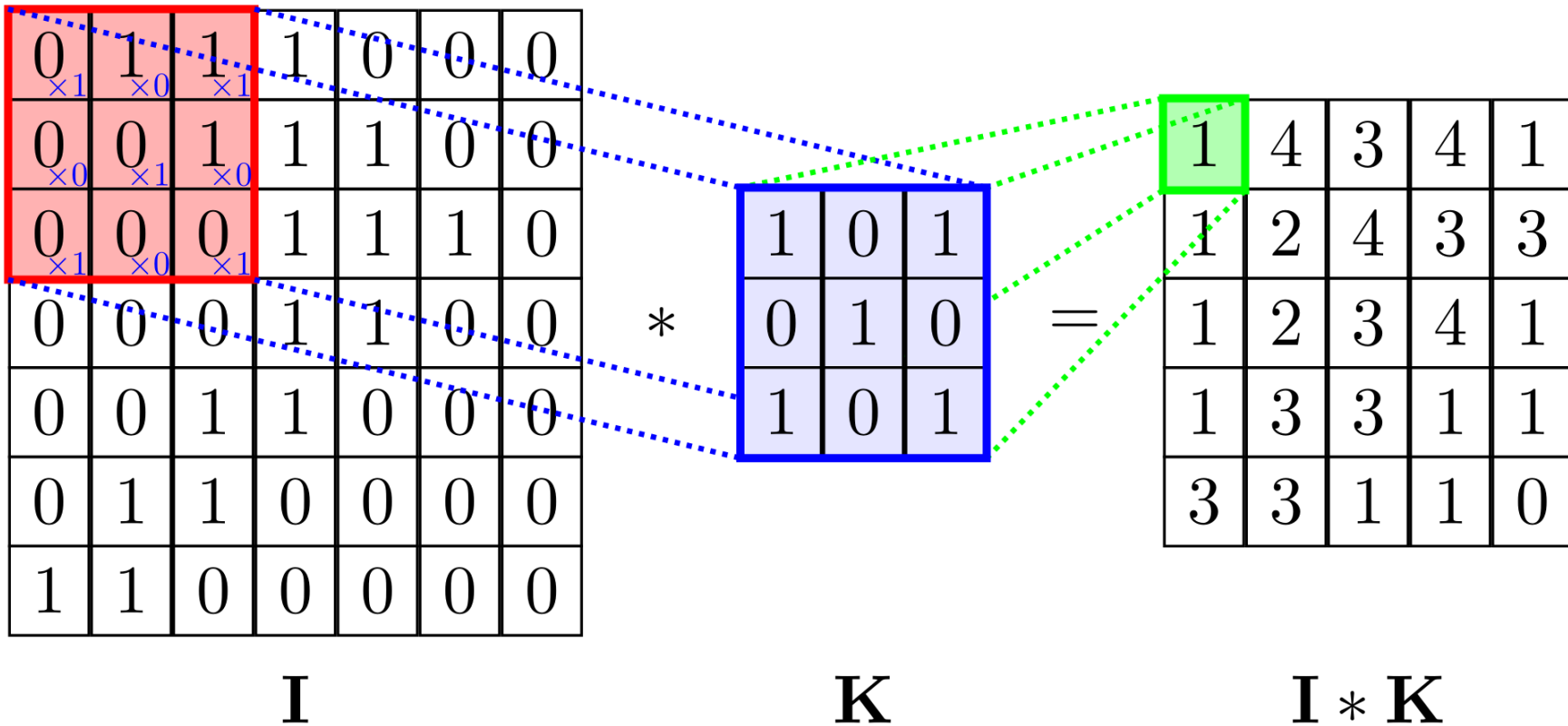
Edge-level representations



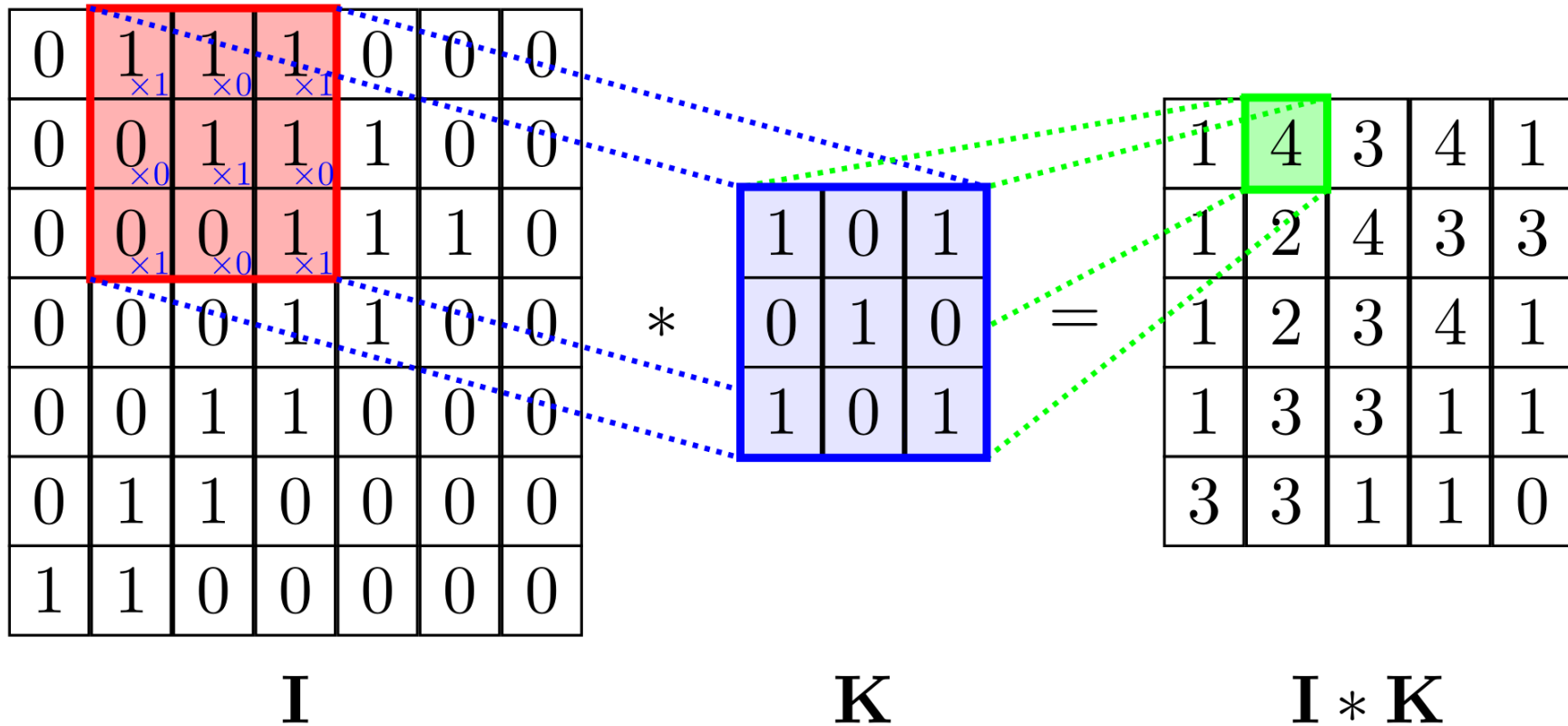
Graph Neural Networks

- What do we want in a neural network acting over graphs?
- Desiderata:
 - Use graph structure - node/edge features, connections between nodes
 - Not sensitive to order in which node / neighbors are processed - *permutation (equi/in)variant*
- Starting point: let's take inspiration from image domain!

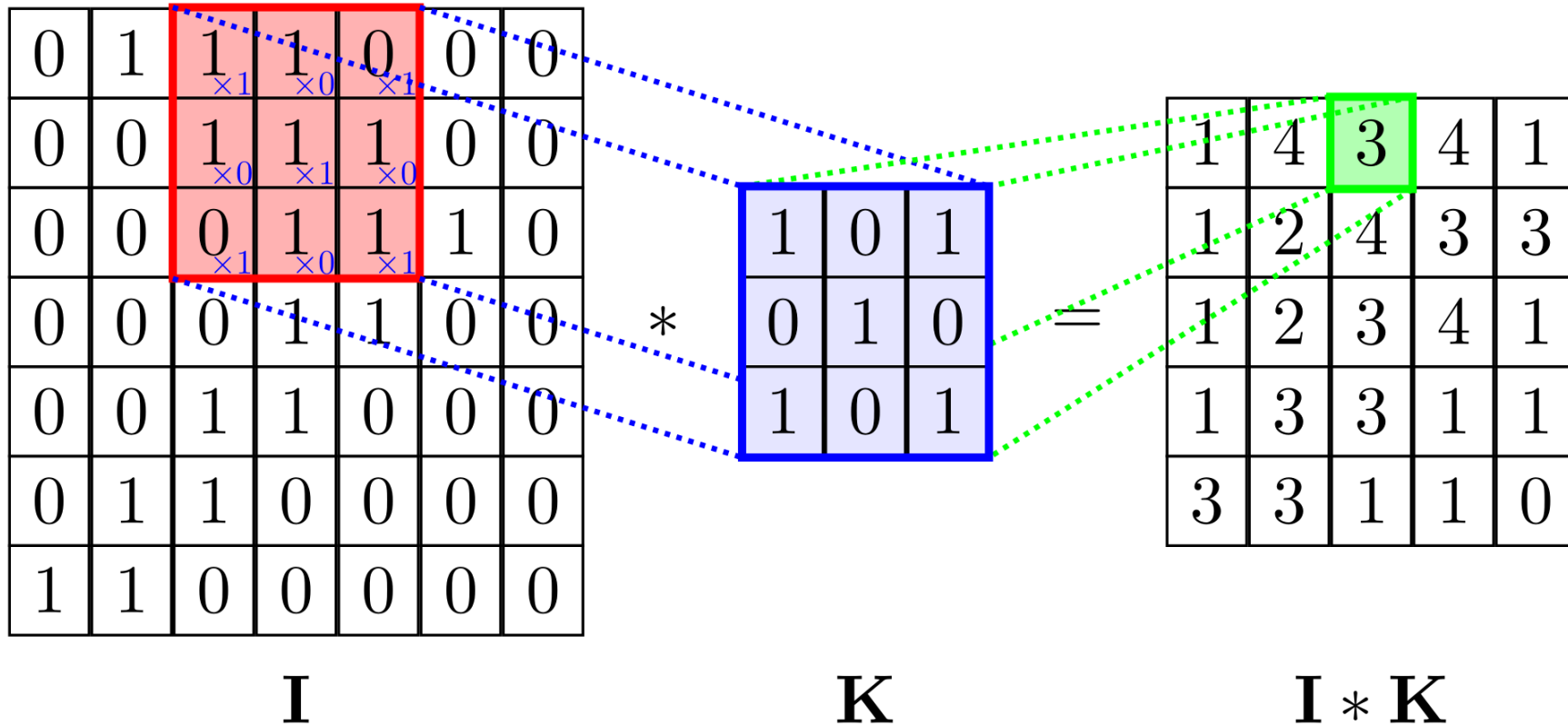
Convolutional Neural Networks



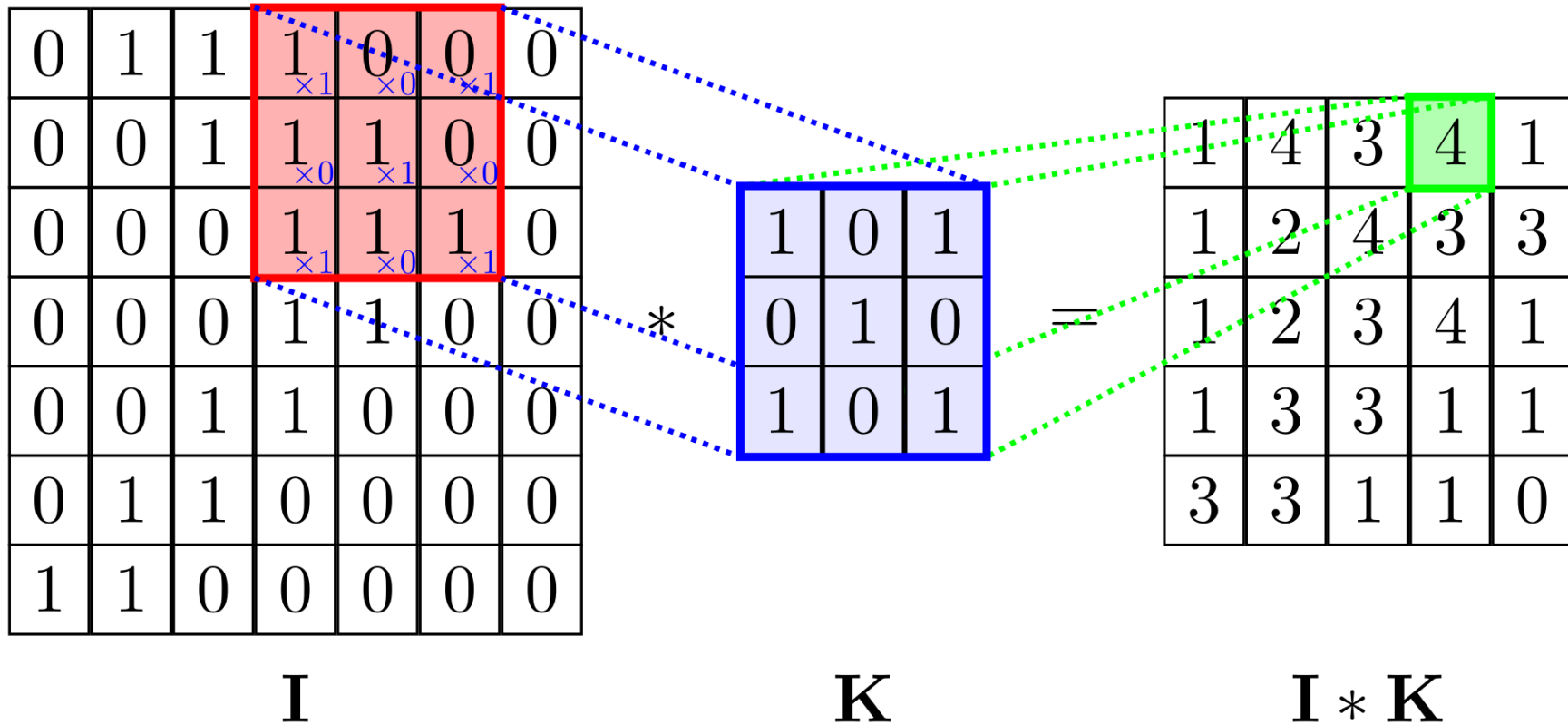
Convolutional Neural Networks



Convolutional Neural Networks



Convolutional Neural Networks



Convolutional Neural Networks

- **Translational invariance**
- Patterns are interesting irrespective of *location* in image
- **Locality**: neighbouring pixels affect more than distant
- Images are essentially graphs
 - Pixels = nodes arranged in grid connectivity pattern
 - What about **arbitrary** graphs?

Graph Convolutional Networks (GCNs)

- Features of neighbours aggregated with fixed weights, c_{ij}

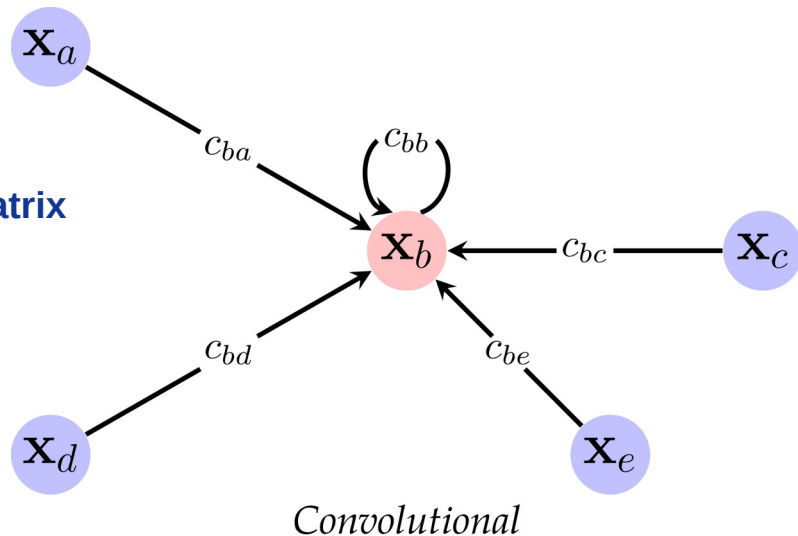
$$\mathbf{h}_i = \phi \left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij} \psi(\mathbf{x}_j) \right)$$

- Usually, the weights depend directly on **adjacency matrix**

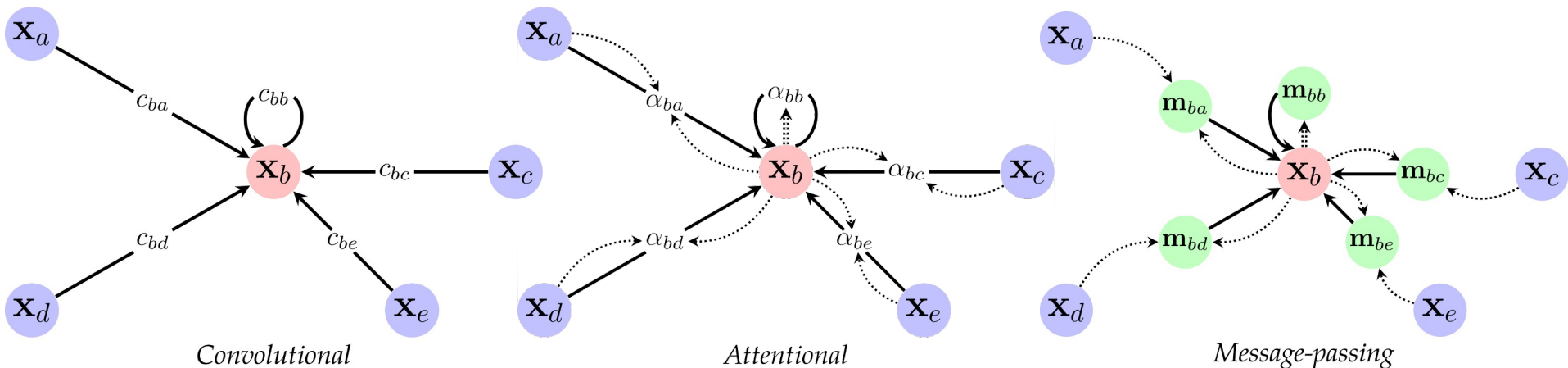
- ChebyNet (Defferrard *et al.*, NeurIPS'16)
- GCN (Kipf & Welling, ICLR'17)
- SGC (Wu *et al.*, ICML'19)

- Useful for **homophilous** graphs and **scaling up**

- When edges encode *label similarity*



The three “flavours” of GNN layers



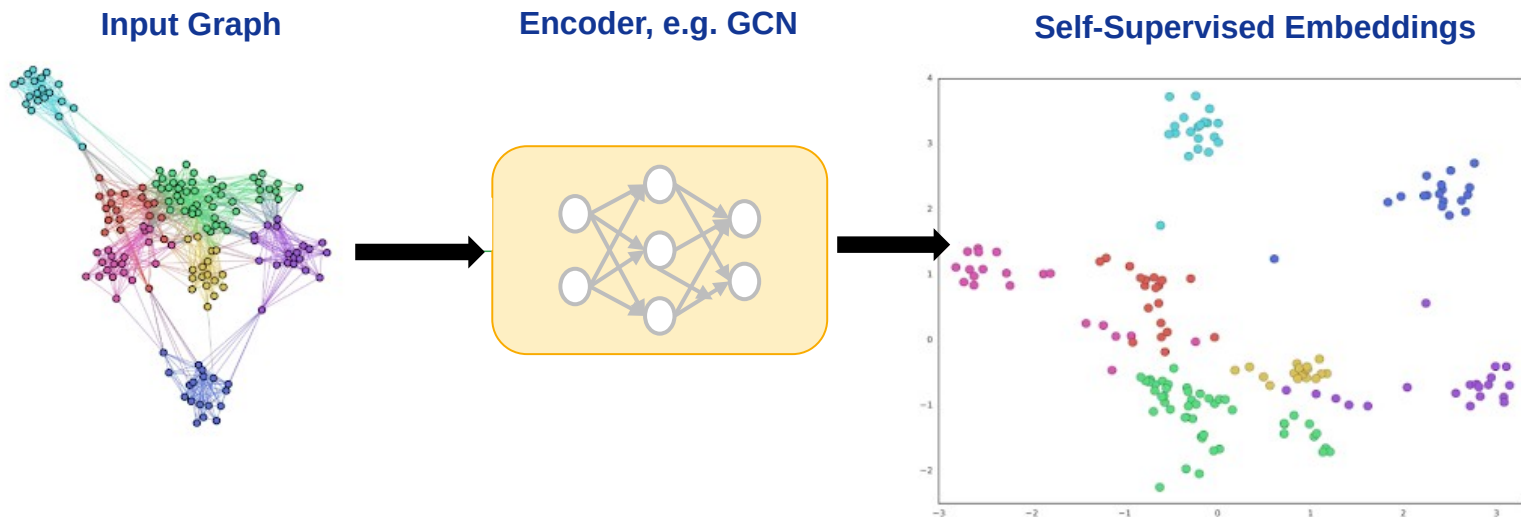
$$\mathbf{h}_i = \phi \left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij} \psi(\mathbf{x}_j) \right)$$

$$\mathbf{h}_i = \phi \left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} a(\mathbf{x}_i, \mathbf{x}_j) \psi(\mathbf{x}_j) \right)$$

$$\mathbf{h}_i = \phi \left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} \psi(\mathbf{x}_i, \mathbf{x}_j) \right)$$

Graph Representation Learning

- Goal: Learn meaningful node representations *without supervision*
- Why?
 - Unlabeled data cheaper
 - Pre-training for downstream tasks
 - Auxiliary signal for semi-supervised training

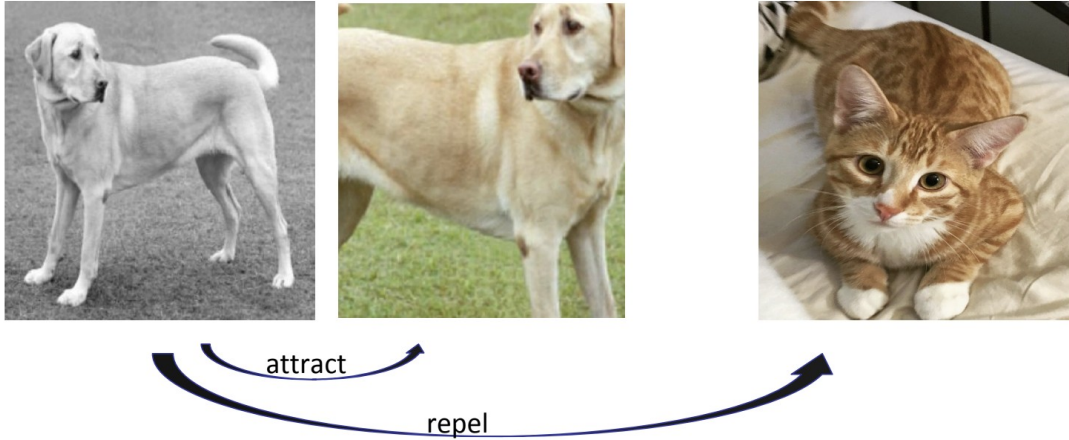


Early methods: Random-walk objectives

- What makes an embedding “good”?
 - Graphs carry interesting **structure**!
 - Good node representations should **preserve** it.
- Simplest notion of graph structure is an *edge*.
 - Features of nodes i and j should be predictive of existence of edge (i, j) !
 - Generalize slightly: nodes i and j co-occur in a short random walk
 - Very similar to NLP methods such as *word2vec*
- Dominated unsupervised graph representation learning prior to GNNs!
 - DeepWalk, node2vec
 - Do not scale to large graphs easily, do not work with GNN encoders

Current hot methods: Contrastive

- Contrastive methods
 - Push together similar objects (*positive examples*)
 - Pull apart dissimilar objects (*negative examples*)

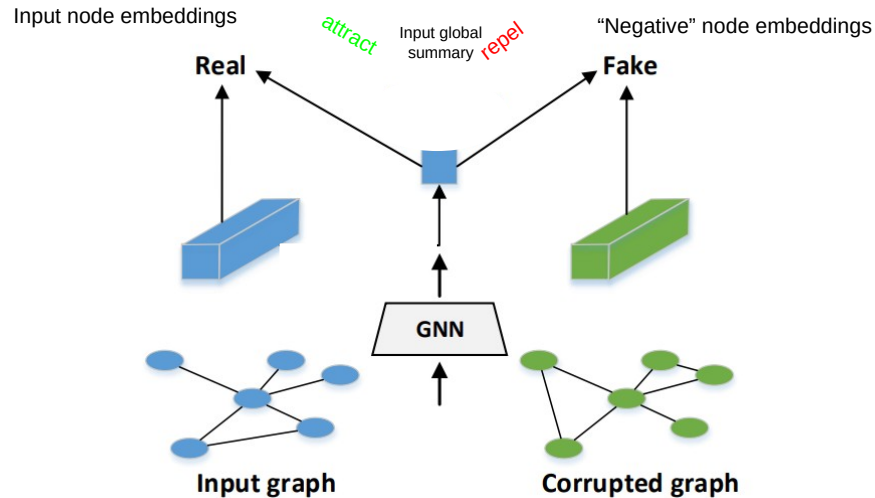


- Aim: stop contrasting dissimilar objects!

... but why?

Drawbacks of Contrastive Methods

- Case Study #1: Deep Graph Infomax (DGI)
 - Contrast against “negative” graph

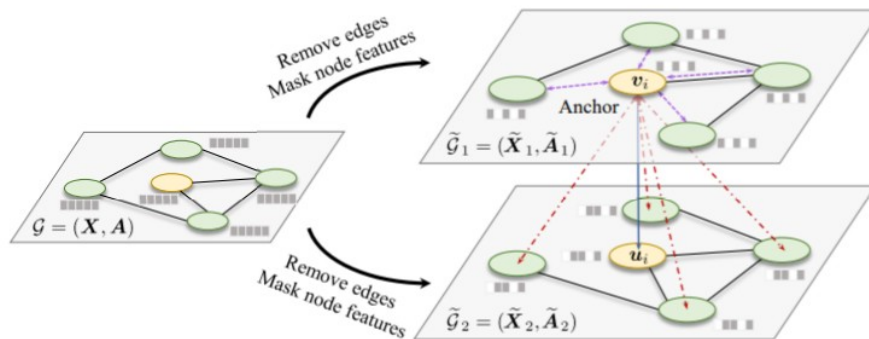


Problem #1: Hard to define negative examples

- Many datasets = *single* graph, no “other” graph

Drawbacks of Contrastive Methods

- Case Study #2: GRACE
 - Positive example = same node across views
 - Negative example = every other pair



Problem #2: All-vs-all contrastive scales quadratically

- Subsampling uniformly is bad
- Choosing smartly is hard

DeepMind

Bootstrapped
GRaph
Latents
(BGRL)

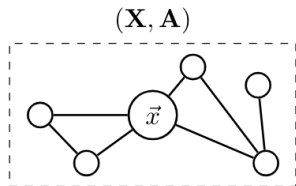


Bootstrapped Graph Latents (BGRL)

- Bootstrap embeddings from each node = no negative examples

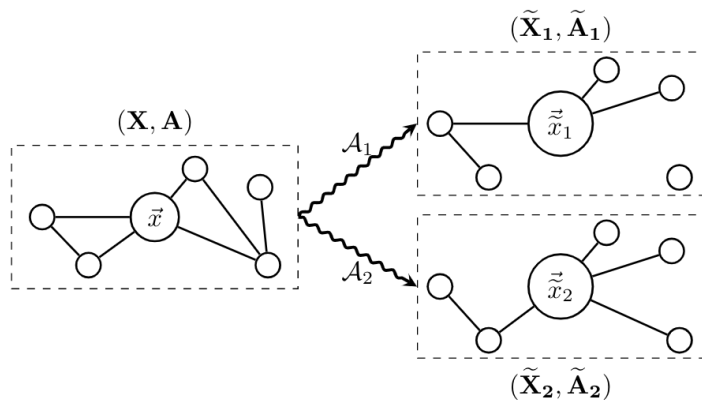
Bootstrapped Graph Latents (BGRL)

- Bootstrap embeddings from each node = no negative examples
 - Given a graph



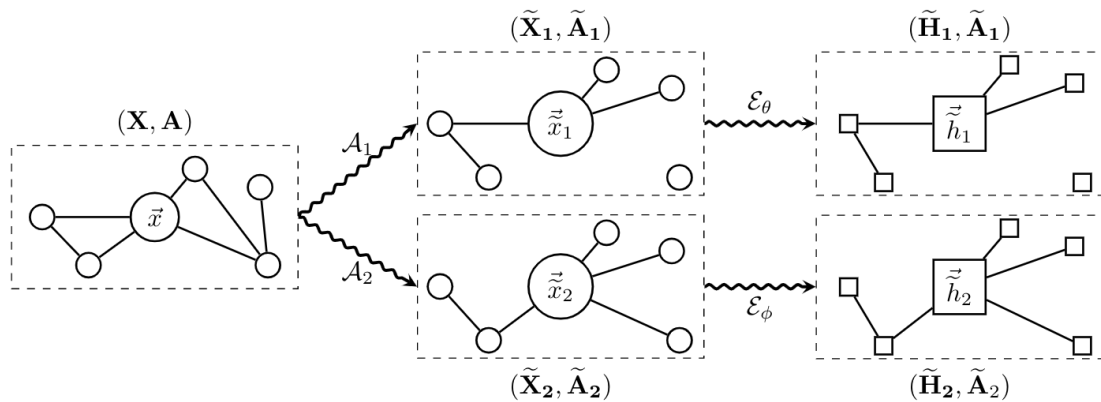
Bootstrapped Graph Latents (BGRL)

- Bootstrap embeddings from each node = no negative examples
 - Generate 2 augmented views
 - Augmentations = transformations embeddings invariant to



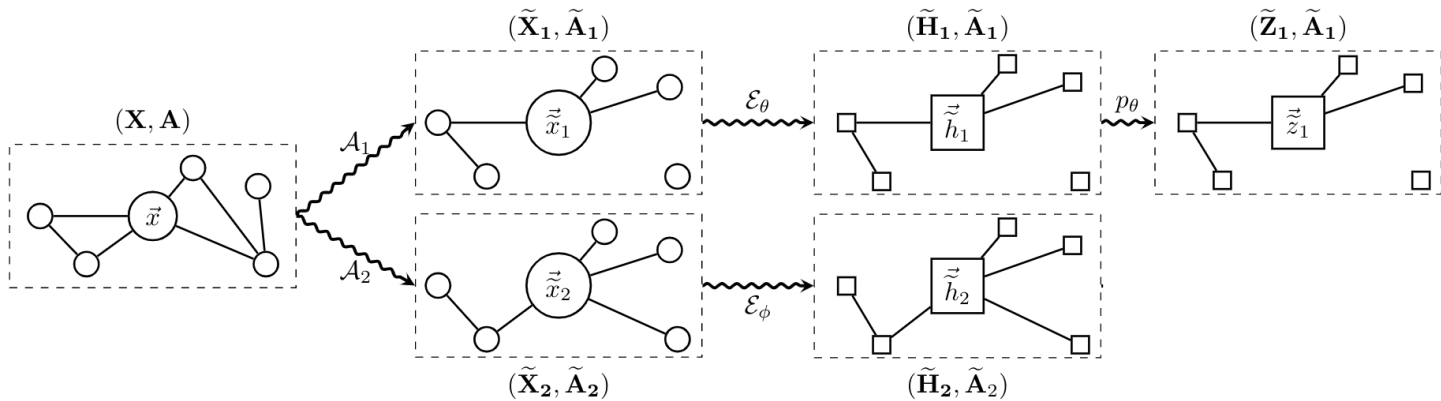
Bootstrapped Graph Latents (BGRL)

- Bootstrap embeddings from each node = no negative examples
 - Two encoders: θ *online*, Φ *target*
 - Compute h_1, h_2 respectively



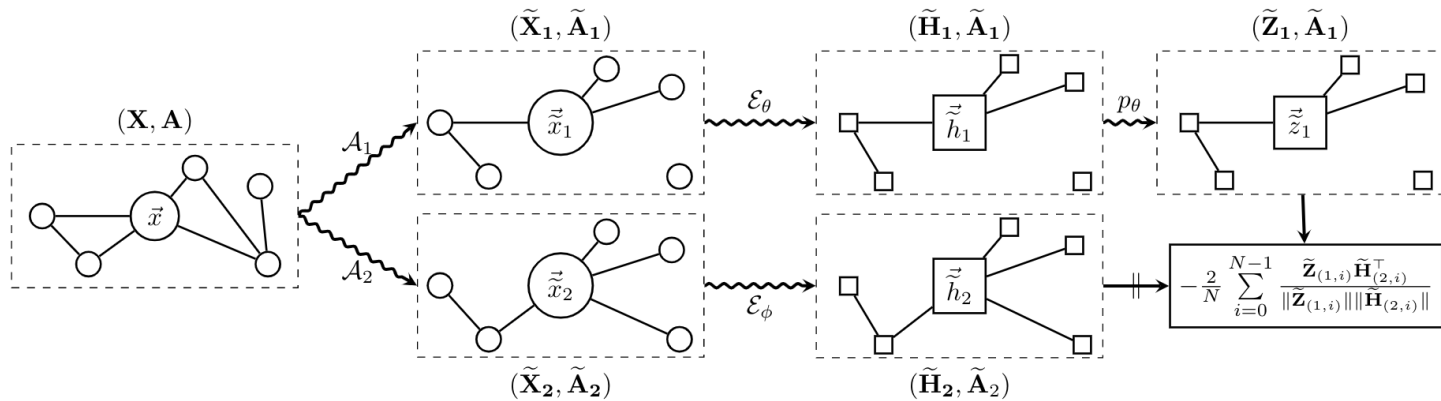
Bootstrapped Graph Latents (BGRL)

- Bootstrap embeddings from each node = no negative examples
 - h_1 trained to be *predictive* of h_2
 - $p_\theta(h_1) = z_1$



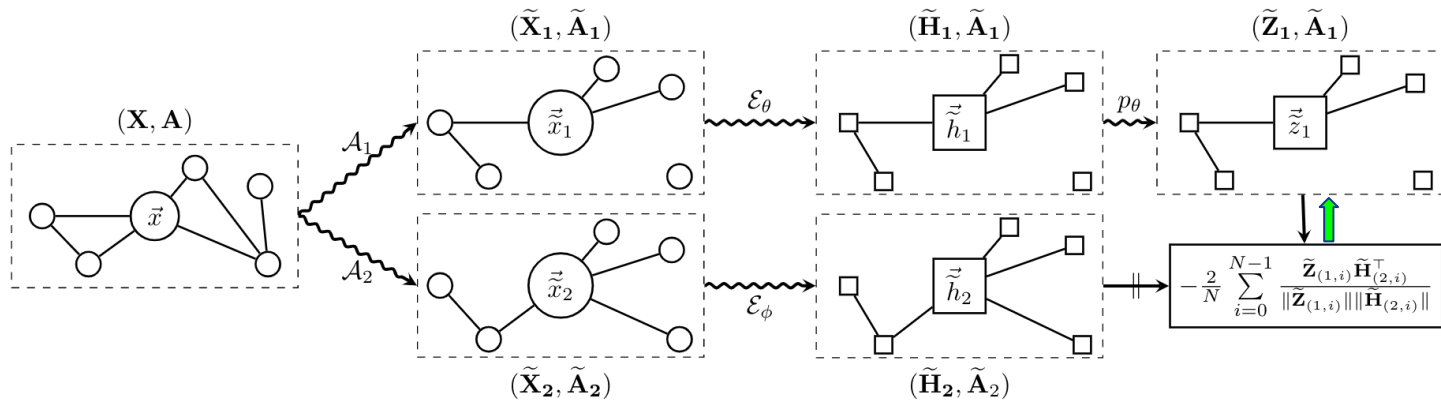
Bootstrapped Graph Latents (BGRL)

- Bootstrap embeddings from each node = no negative examples
 - z_1 pushed towards h_2



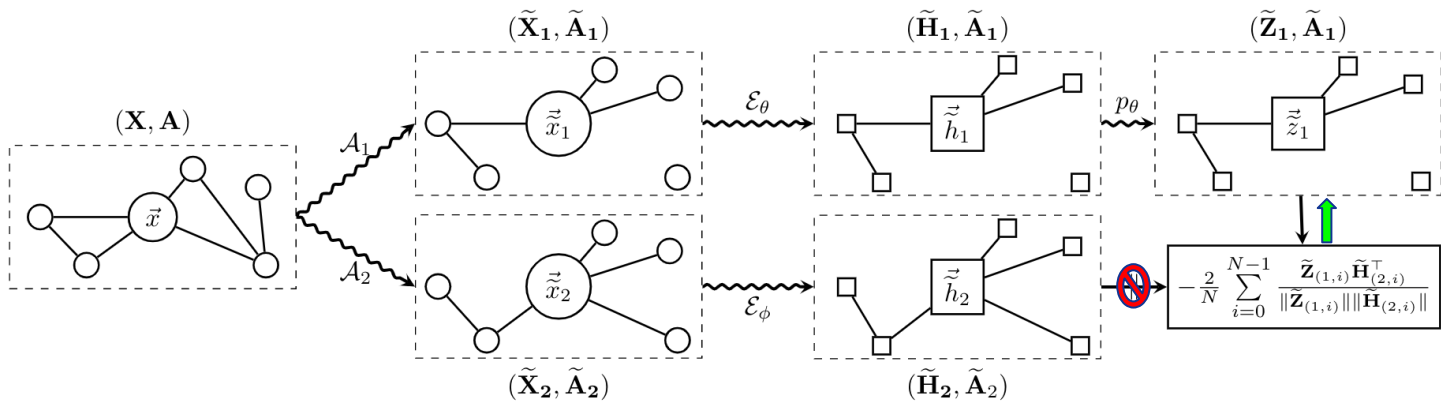
Bootstrapped Graph Latents (BGRL)

- Bootstrap embeddings from each node = no negative examples
 - Flow gradients through θ



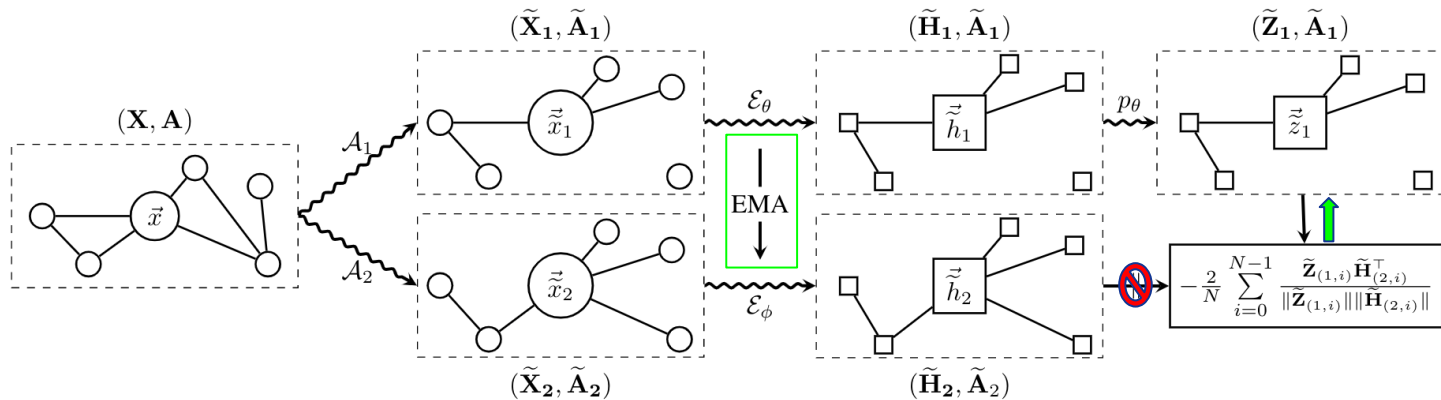
Bootstrapped Graph Latents (BGRL)

- Bootstrap embeddings from each node = no negative examples
 - Block gradients through Φ



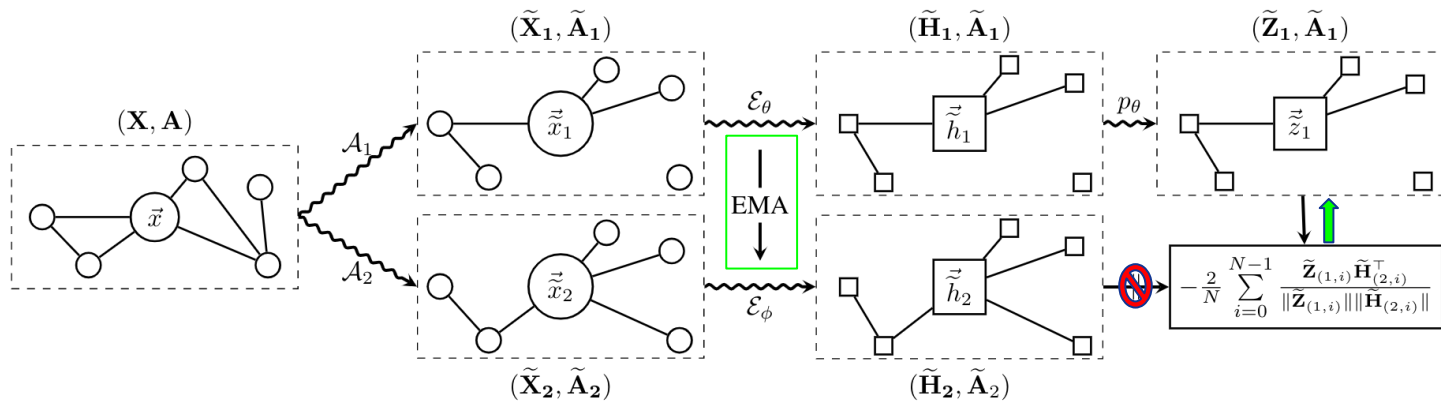
Bootstrapped Graph Latents (BGRL)

- Bootstrap embeddings from each node = no negative examples
 - Φ updated as EMA of θ



Bootstrapped Graph Latents (BGRL)

- Adaptation from BYOL - no projector network



- Undesirable/trivial solutions exist (e.g. $\theta = \Phi$)
 - Not obtained as (θ, Φ) update does not minimize any loss

Graph Augmentations

- Design decision, perturbations that do not change semantics
- For images, intuitive to design
 - Flipping/cropping/color distortions typically not change class
- For graphs, very unintuitive!
 - Perturb *whole graph*
 - But learn embeddings for *nodes*
 - It would be like augmenting an image but learning pixel-level!
- So simple, cheap augmentations done:
 - Randomly drop certain edges
 - Random node feature masking
 - Not perfect, still open area of research!

Mehdi Azabou, Venkataramana Ganesh, Shantanu Thakoor, Chi-Heng Lin, Lakshmi Sathidevi, Ran Liu, Michal Valko, Petar Veličković, Eva L Dyer: *Half-Hop: A graph upsampling approach for slowing down message passing*. in [International Conference on Machine Learning \(ICML 2023\)](#)

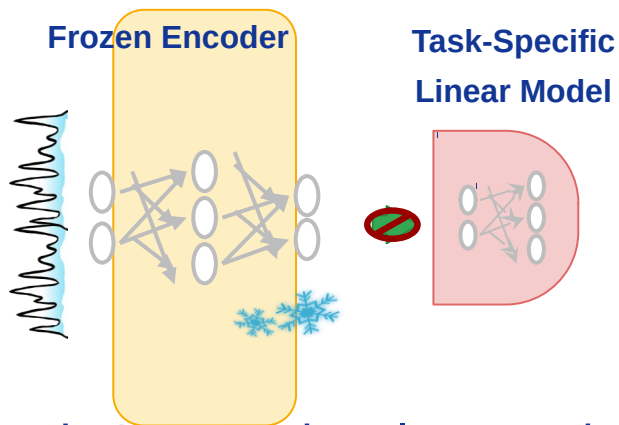
DeepMind

Experiments



Experimental Setup

- Node classification, GRACE current best
- Linear evaluation protocol



- Encoders: Graph Convolutional Networks (GCNs)
BGRL predictor: MLP
- Simple augmentations, masking with *fixed* probability

Experimental Setup

- Report results relative to randomly initialized GCN
- Very strong baseline!
 - Random GCNs = good inductive bias
 - Linear classifier on top works as normal
 - Surpasses pure supervised in some cases!

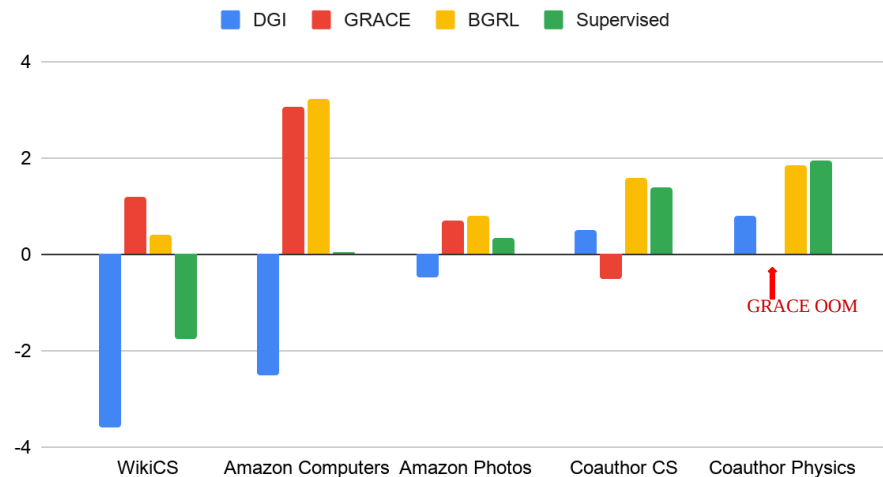
Datasets

- Transductive tasks:
 - Single graph, all nodes known during training, labels only available for training nodes
 - WikiCS, Coauthor CS/Physics, ogbn-arXiv: citations networks, classify paper topic
 - Amazon Computers/Photos: co-purchase graphs, classify product type
- Inductive tasks:
 - Dataset of many graphs, train on some/test on others
 - PPI: dataset of protein-protein interactions, predict biological properties

Experimental Results

- Citations/Co-purchase graphs, $O(10k)$ nodes → quadratic possible

Accuracy Relative to Random Embeddings

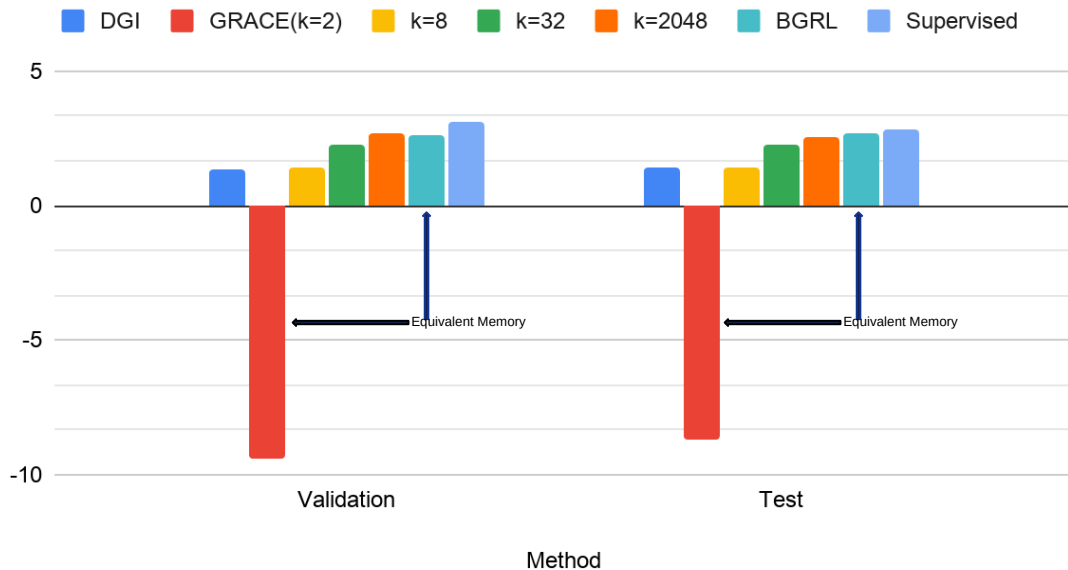


- Not only is BGRL \geq other methods, memory usage is **5-10x smaller**

Scaling Up to Larger Graphs

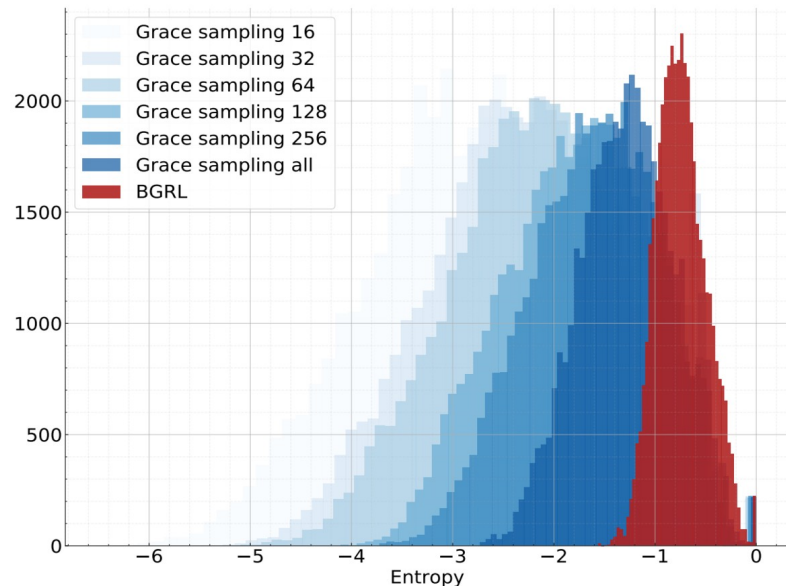
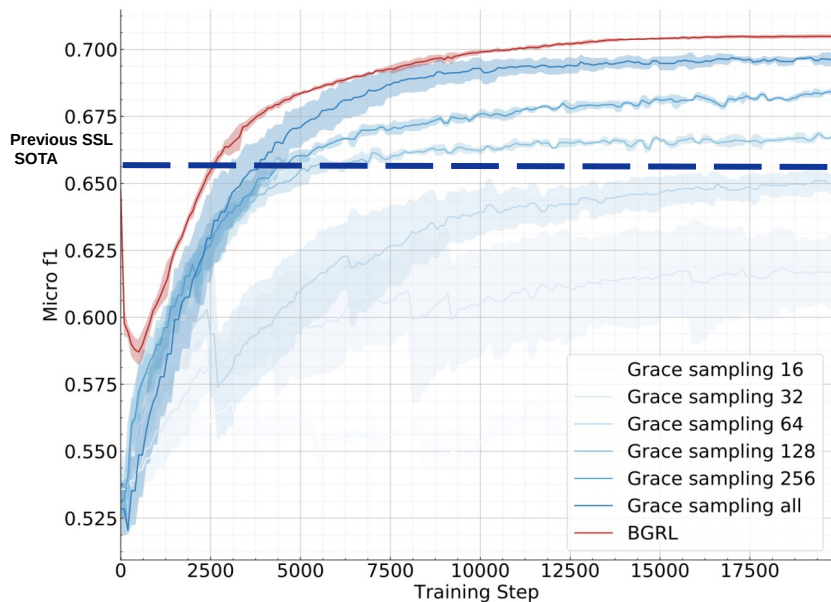
- OGB arXiv dataset, 170k nodes
- Subsample k negatives per node for GRACE
 - $k=2 \approx$ BGRL in asymptotic memory

Accuracy Relative to Random Embeddings



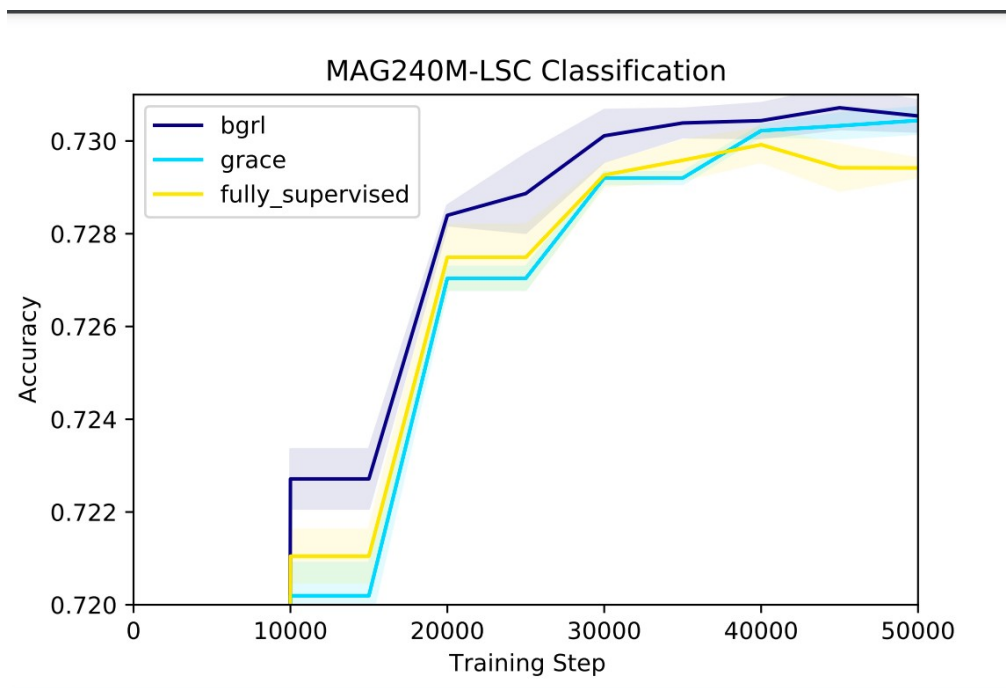
Pushing Performance on PPI

- PPI: biological networks of protein interactions, $O(50k)$ nodes
 - Huge gap between self-supervised and fully supervised
 - Graph Attentional encoders



Unlocking performance on 1000x larger dataset

- KDD Cup 2021:
OGB-LSC challenge,
dataset with 240M
nodes / 1B edges
- BGRL was key to
DeepMind team
awarded as Top-3
- BGRL works even with:
 - 1000x larger data
 - Expressive MPNNs
 - Mixing with
supervised signals



Conclusions

- Main takeaways:
 - BGRL competitive with contrastive methods without negative examples
 - Huge wins in memory and performance in some cases
 - Likely to be more easily applied to larger graphs without design choices

- Future directions:
 - Naturally extends to learning graph-level embeddings
 - Experimenting with stronger encoder architectures
 - Research into stronger graph-based augmentations

DeepMind

Thank You!

... Questions?

