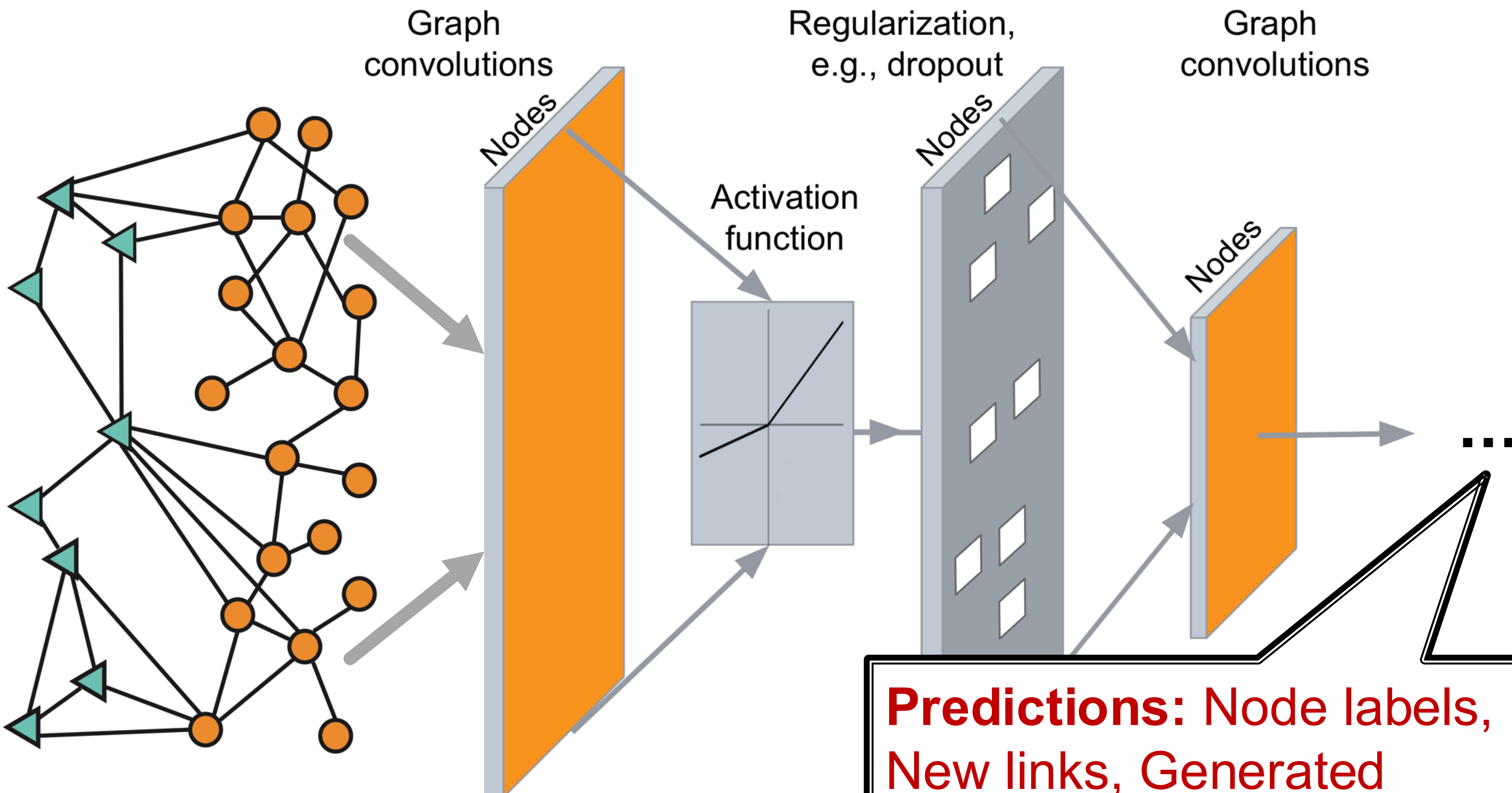
Design Space of Graph Neural Networks

CS224W: Machine Learning with Graphs
Charilaos Kanatsoulis, Stanford University
http://cs224w.stanford.edu

Announcements

- Congratulations on completing this class! We are happy that you joined us.
  - Today is the last lecture!
- **Exam grades have been released!**
  - Regrade requests open until 12/10 EOD
- **Colab 5** due today (12/4)
- **Project Report** due next Thursday (12/11)
  - We will open a form on Gradescope to upload a link to your Medium article and code

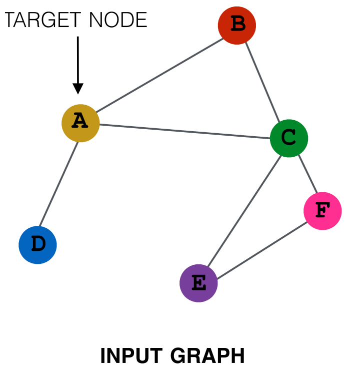
CS224W: Deep Learning in Graphs

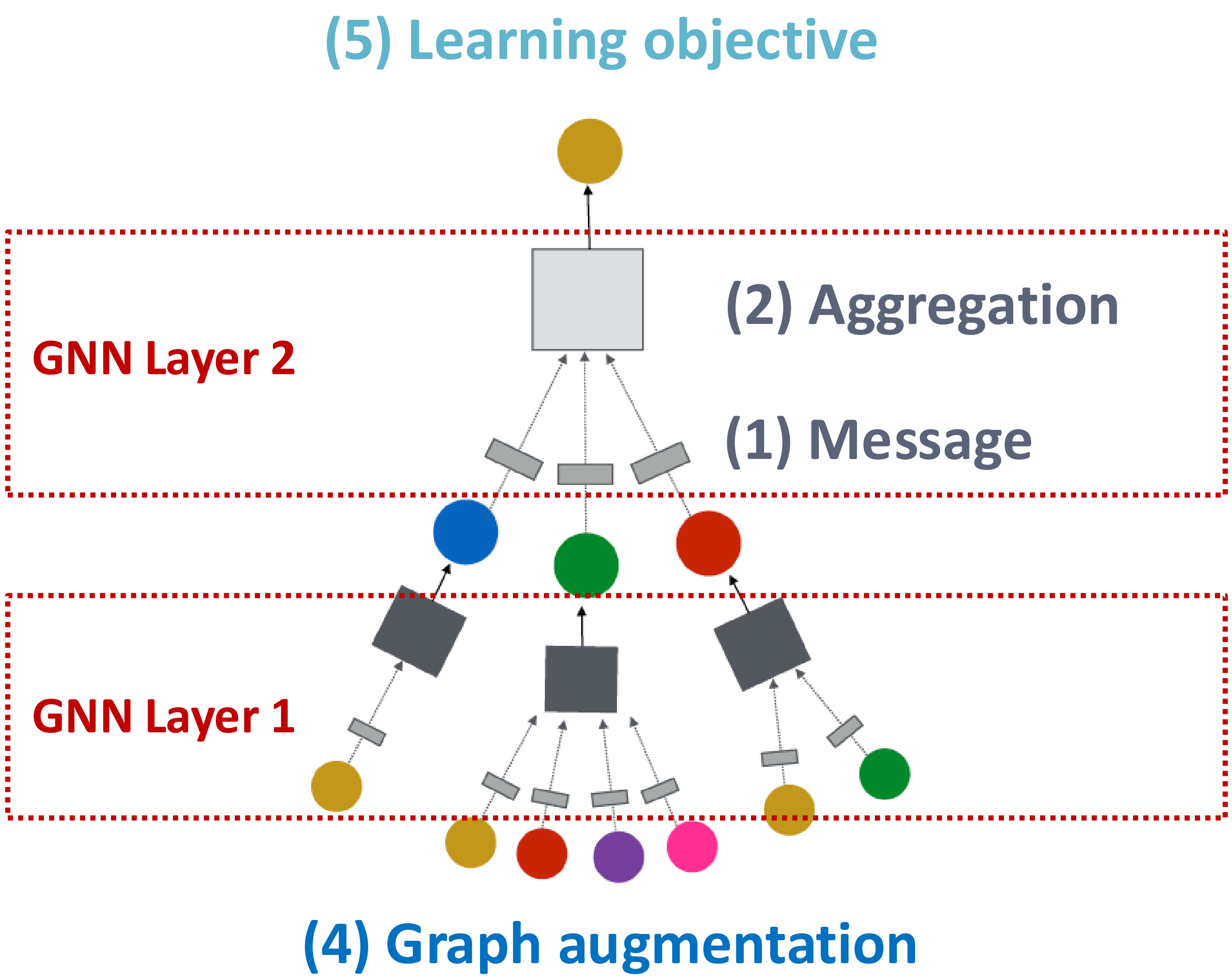


Input: Network

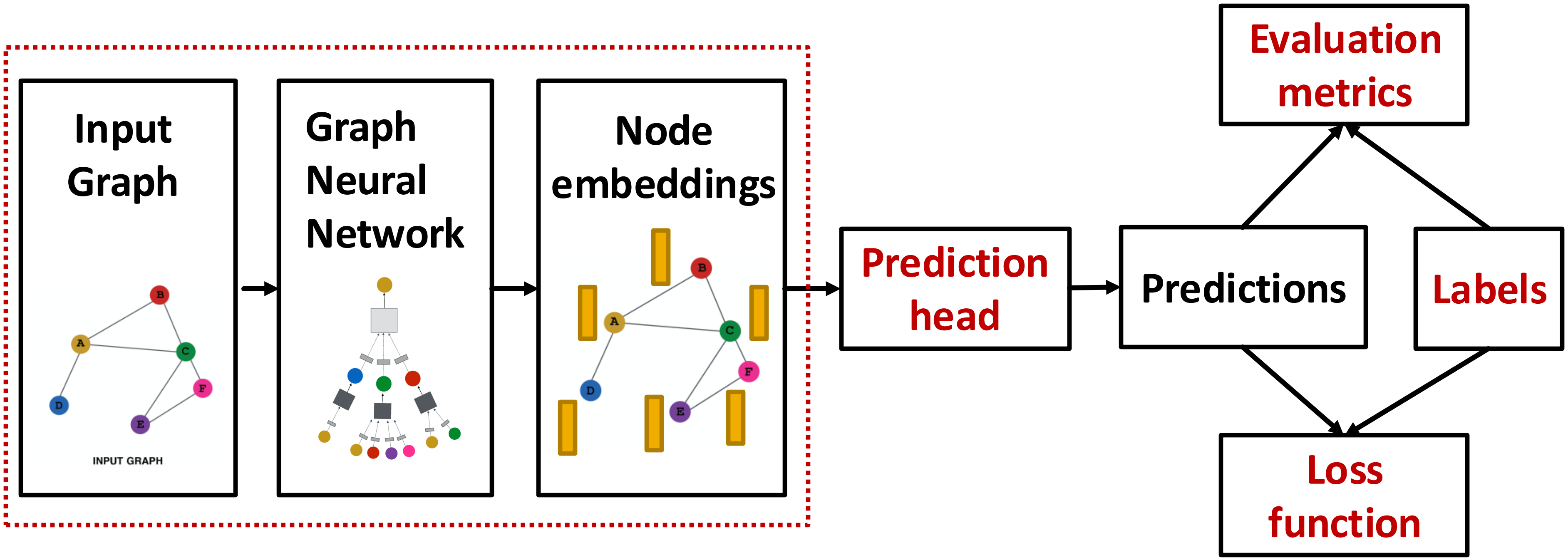
Predictions: Node labels, New links, Generated graphs and subgraphs

A General GNN Framework





GNN Training Pipeline



Output of a GNN: set of node embeddings

$$\{\mathbf{h}_v^{(L)}, \forall v \in G\}$$

Key Questions for GNN Design

- **GNN architectural design:**
  - How to find a good GNN design for a specific GNN task?
- **Important but challenging:**
  - Domain experts want to use SOTA GNN on their specific tasks, however...
    - There are tons of possible GNN architectures
      - GCN, GraphSAGE, GAT, GIN, …
    - **Issue:** Best design in one task can perform badly for another task
    - Redo hyperparameter grid search for each new task is NOT feasible
- **Topic for today:**
  - Study for the ***GNN design space and task space***
  - **GraphGym**, a powerful platform for exploring different GNN designs and tasks

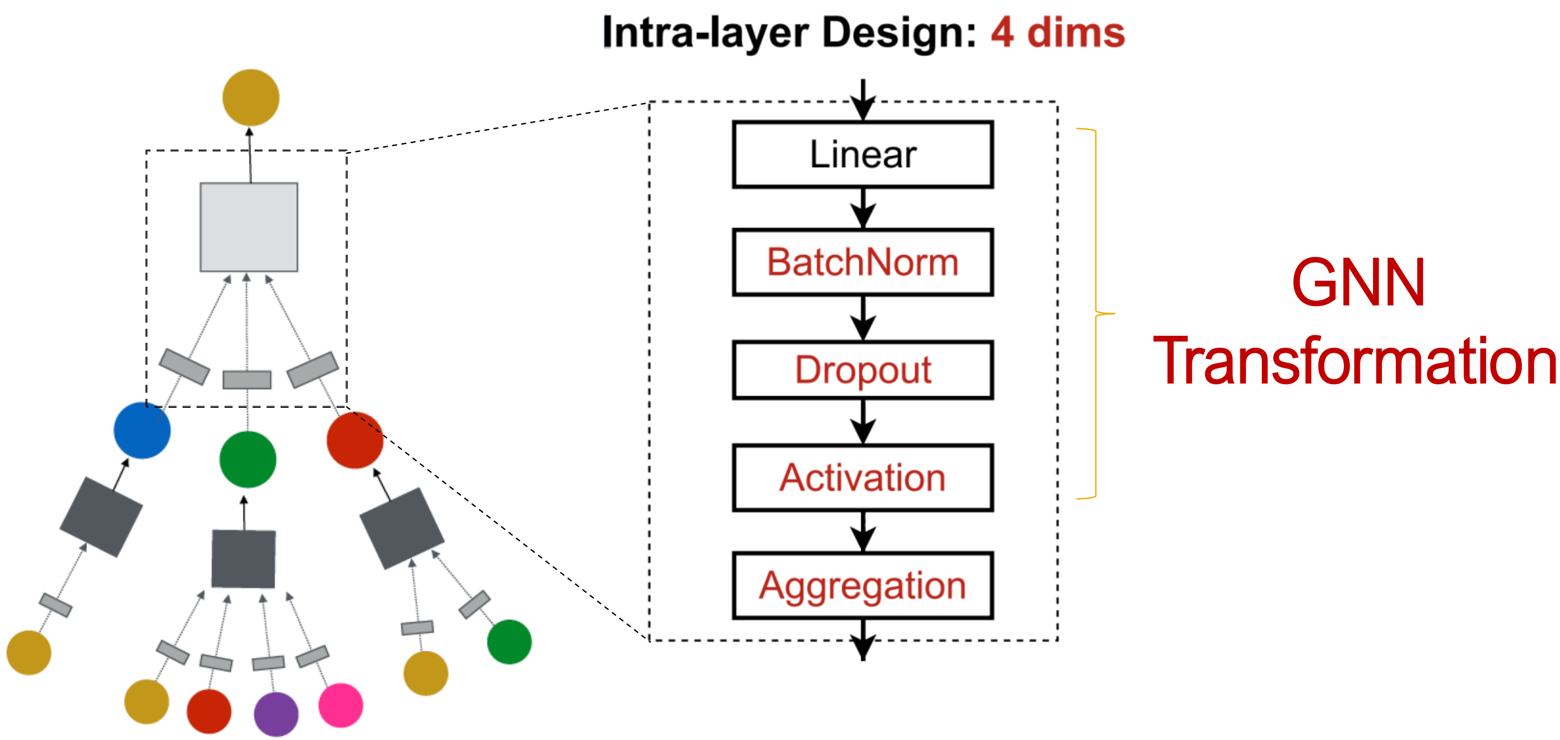
Background: Terminology

- **Design:** a concrete model instantiation
  - E.g., a 4-layer GraphSAGE
- **Design dimensions** characterize a design
  - E.g., the number of layers $L \in \{2, 4, 6, 8\}$
- **Design choice** is the actual selected value in the design dimension
  - E.g., the number of layers $L = 2$
- **Design space** consists of a Cartesian product of design dimensions
- **Task:** A specific task of interest
  - E.g., node classification on Cora, graph classification on ENZYMES
- **Task space** consists of all the tasks we care about

Recap: GNN Design Space

Intra-layer Design:
GNN Layer = Transformation + Aggregation

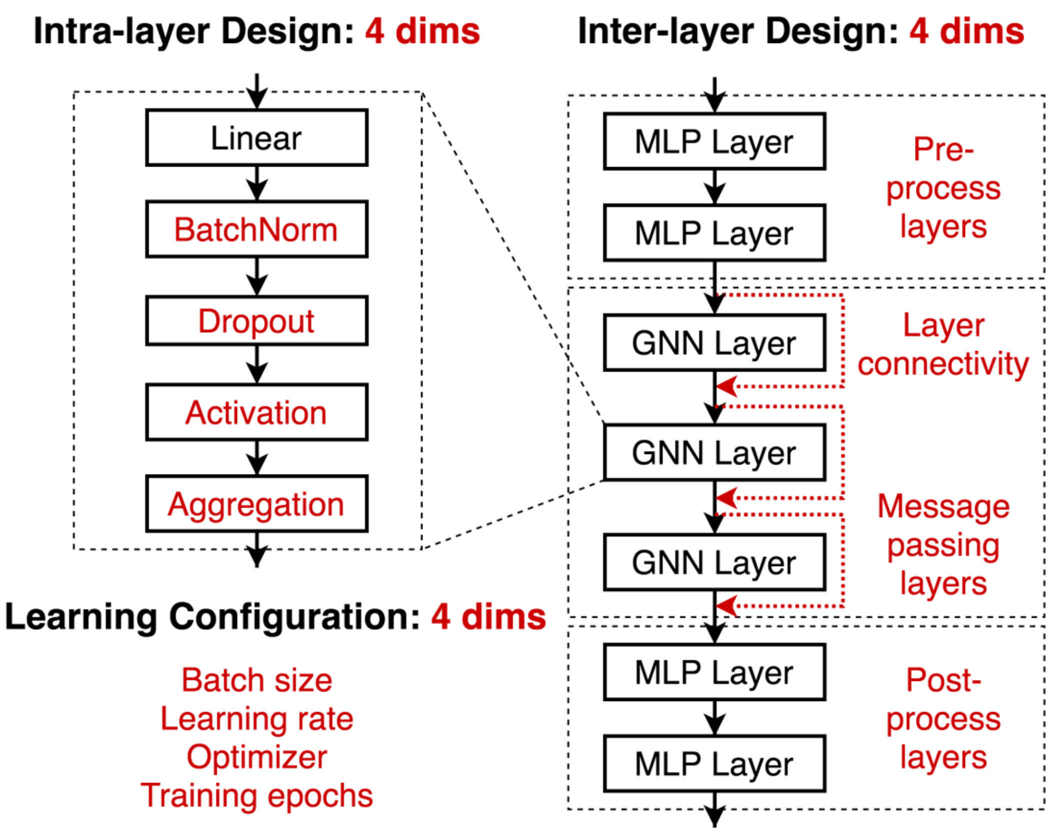
- We propose a general instantiation under this perspective



Recap: GNN Design Space

Inter-layer Design

- **We explore different ways of organizing GNN layers**

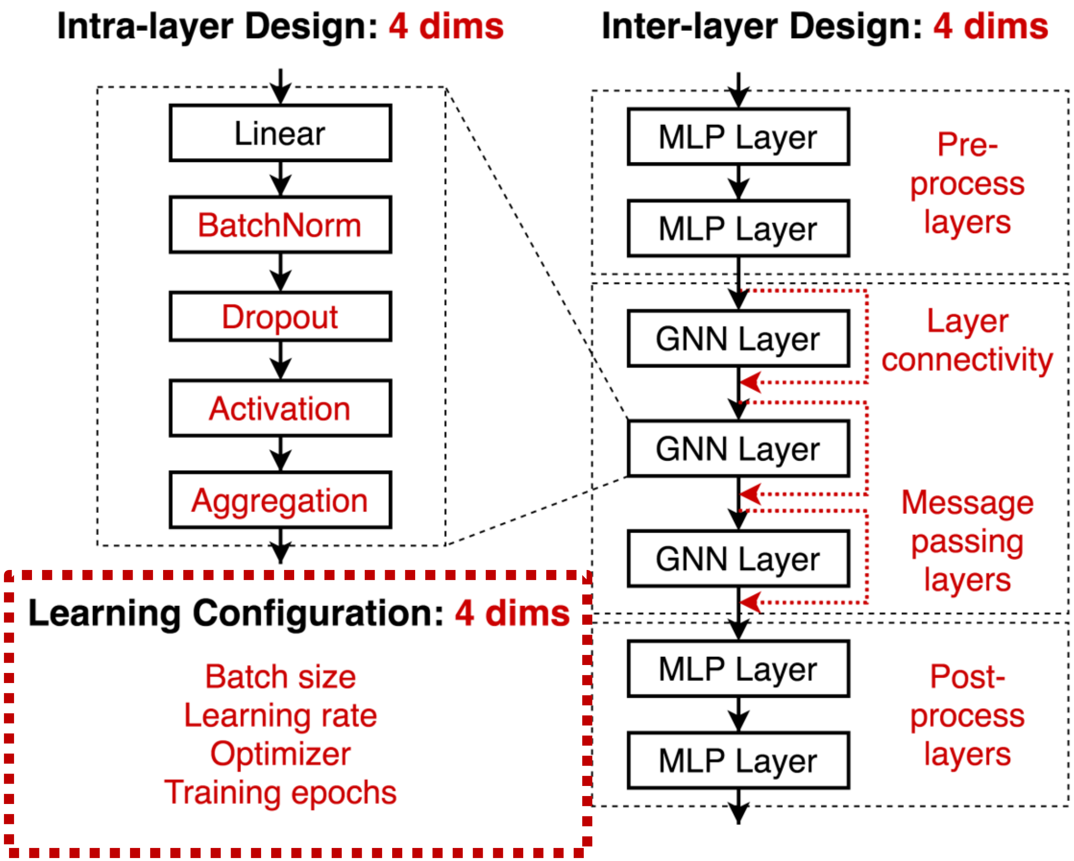


Pre-process layers:
Important when expressive node feature encoder is needed
E.g., when nodes are images/text

Skip connections:
Improve deep GNN's performance

Post-process layers:
Important when reasoning or transformation over node embeddings are needed
E.g., graph classification, knowledge graphs

Recap: GNN Design Space



Learning configurations

- Often neglected in current literature
- But we found they have high impact on performance

Summary: GNN Design Space

- **Overall: A GNN design space**
  - Intra-layer design

Batch Normalization	Dropout	Activation	Aggregation
True, False	False, 0.3, 0.6	ReLU, PReLU, Swish	Mean, Max, Sum

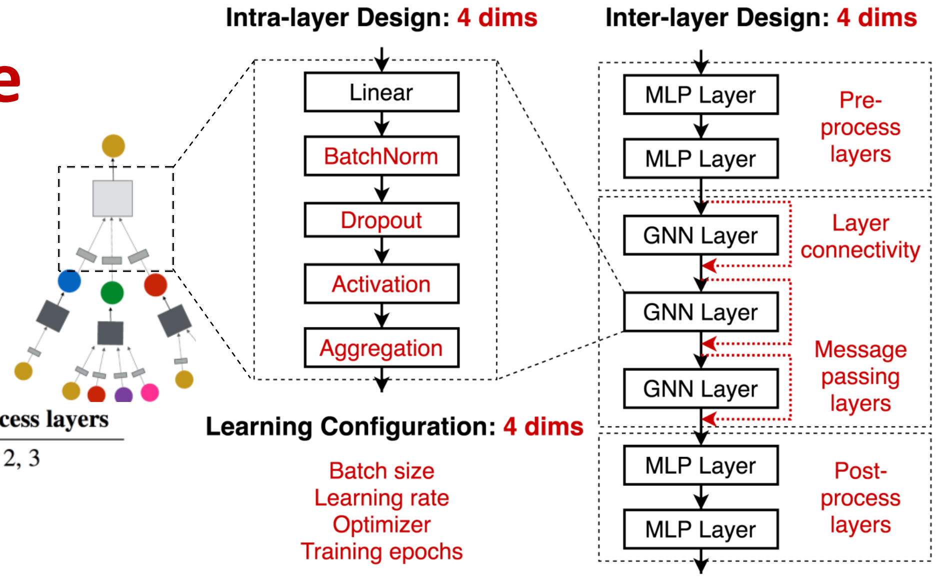
  - Inter-layer design

Layer connectivity	Pre-process layers	Message passing layers	Post-precess layers
Stack, Skip-Sum, Skip-Cat	1, 2, 3	2, 4, 6, 8	1, 2, 3

  - Learning configuration

Batch size	Learning rate	Optimizer	Training epochs
16, 32, 64	0.1, 0.01, 0.001	SGD, Adam	100, 200, 400

  - **In total:** 315K possible designs



- **Our Purpose:**
  - We don't want to (and we cannot) cover all the possible designs
  - **A mindset transition:** We want to demonstrate that **studying a design space is more effective than studying individual GNN designs**

A General GNN Task Space

- **Categorizing GNN tasks**
  - **Common practice**: node / edge / graph level task
  - Reasonable but not precise
    - **Node prediction:** predict clustering coefficient vs. predict a node's subject area in a citation networks – **completely different task**
  - But creating a precise taxonomy of GNN tasks is very hard!
    - **Subjective**; **Novel GNN tasks** can always emerge
- **Our innovation: a quantitative task similarity metric**
  - **Purpose: understand GNN tasks, transfer the best GNN models across tasks**

A General GNN Task Space

- **Quantitative task similarity metric**
  - **1)** Select **"anchor" models** $(M_1, \ldots, M_5)$
  - **2)** Characterize a task by ranking the performance of anchor models
  - **3)** Tasks with similar rankings are considered as similar

Task Similarity Metric

	Anchor Model Performance ranking					Similarity to Task A
Task A	M_1	M_2	M_3	M_4	M_5	1.0
Task B	M_1	M_3	M_2	M_4	M_5	0.8
Task C	M_5	M_1	M_4	M_3	M_2	-0.4

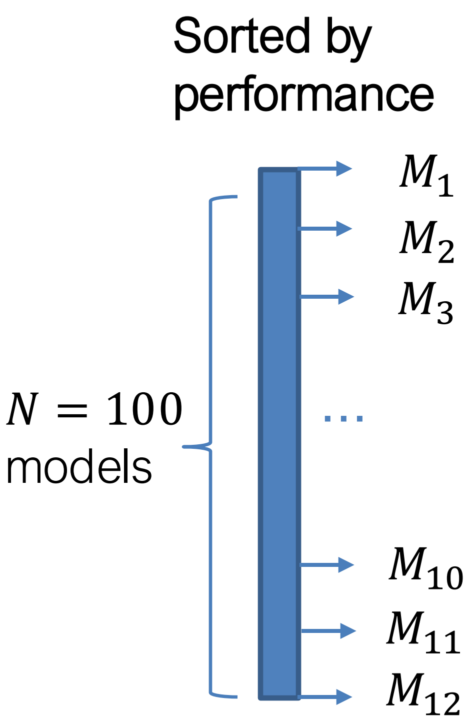
Task A is similar to Task B

Task A is not similar to Task C

- **How do we select the anchor models?**

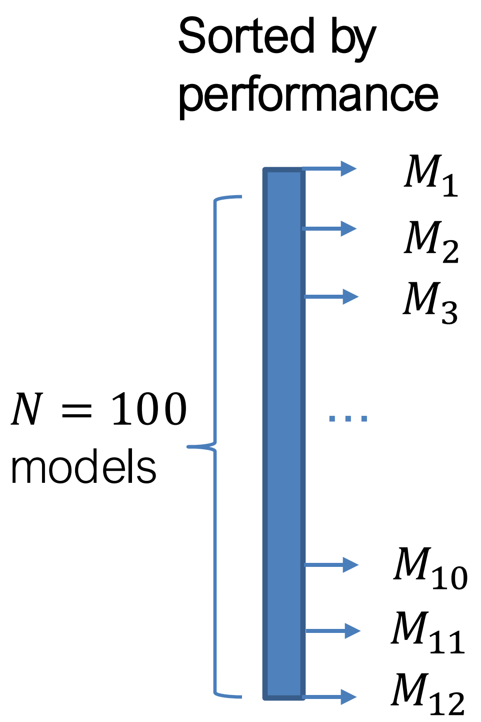
A General GNN Task Space

- **Selecting the anchor models**
  - 1) Select a small dataset
    - E.g., node classification on Cora
  - 2) Randomly **sample N models from our design space**, run on the dataset
    - E.g., we sample 100 models
  - 3) Sort these models based on their performance:



A General GNN Task Space

- **Selecting the anchor models**
  - 1) Select a small dataset
    - E.g., node classification on Cora
  - 2) Randomly **sample N models from our design space**, run on the dataset
    - E.g., we sample 100 models
  - 3) Sort these models based on their performance: **evenly select M models as the anchor models**, whose **performance range from the worst to the best**
    - E.g., we sample 12 models in our experiments
  - **Goal: Cover a wide spectrum of models:** A bad model in one task could be great for another task



A General GNN Task Space

- **We collect 32 tasks**: node / graph classification

(We include link prediction results in the Appendix)

Task name
node-AmazonComputers-N/A-N/A
node-AmazonPhoto-N/A-N/A
node-CiteSeer-N/A-N/A
node-CoauthorCS-N/A-N/A
node-CoauthorPhysics-N/A-N/A
node-Cora-N/A-N/A
node-scalefree-clustering-pagerank
node-scalefree-const-clustering
node-scalefree-const-pagerank
node-scalefree-onehot-clustering
node-scalefree-onehot-pagerank
node-scalefree-pagerank-clustering
node-smallworld-clustering-pagerank
node-smallworld-const-clustering
node-smallworld-const-pagerank
node-smallworld-onehot-clustering
node-smallworld-onehot-pagerank
node-smallworld-pagerank-clustering
graph-PROTEINS-N/A-N/A
graph-BZR-N/A-N/A
graph-COX2-N/A-N/A
graph-DD-N/A-N/A
graph-ENZYMES-N/A-N/A
graph-IMDB-N/A-N/A
graph-scalefree-clustering-path
graph-scalefree-const-path
graph-scalefree-onehot-path
graph-scalefree-pagerank-path
graph-smallworld-clustering-path
graph-smallworld-const-path
graph-smallworld-onehot-path
graph-smallworld-pagerank-path
graph-ogbg-molhiv-N/A-N/A

6 Real-world node classification tasks

12 Synthetic node classification tasks
Predict node properties:
- Clustering coefficient
- PageRank

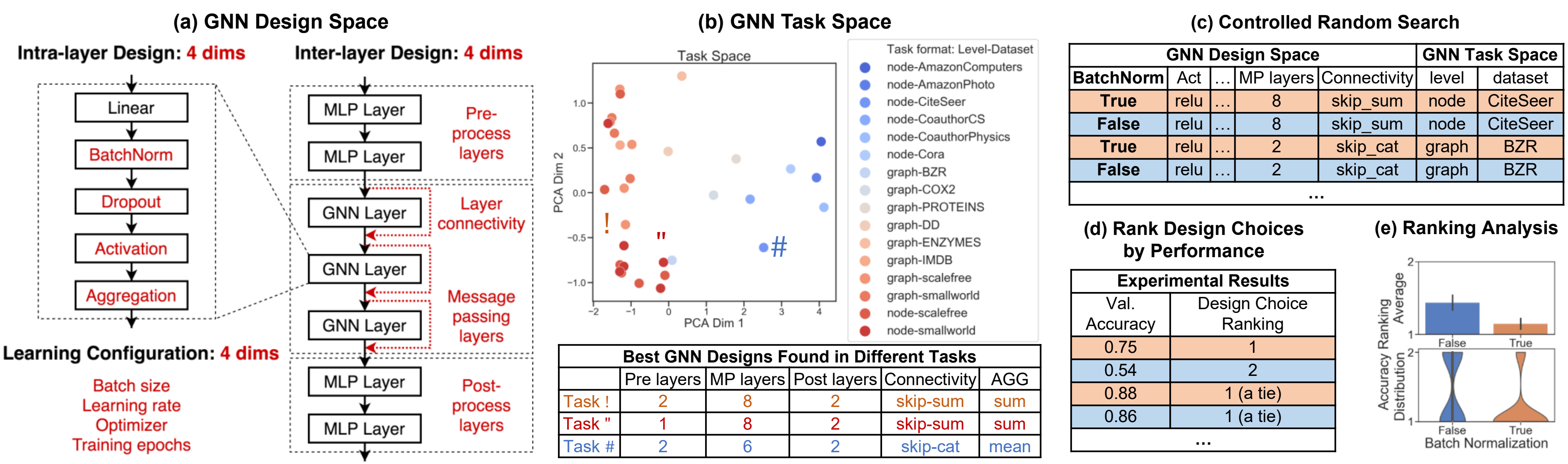
6 Real-world graph classification tasks

8 Synthetic graph classification tasks
Predict graph properties:
- Average path length

J. You, R. Ying, J. Leskovec. Design Space of Graph Neural Networks, NeurIPS 2020

GNN Design Space

- Systematic investigation of:
  - **General guidelines for GNN design**
  - **Understandings of GNN tasks**
  - **Transferring best GNN designs across tasks**



Evaluating GNN Designs

- **Evaluating a design dimension**:
  - "Is BatchNorm generally useful for GNNs?"
- **The common practice:**
  - **(1) Pick one model** (e.g., a 5-layer 64-dim GCN)
  - **(2) Compare two models**, with BN = True / False
- **Our approach:**
  - Note that **we have defined** $315\text{K (models)} * 32\text{ (tasks)} \approx$ **10M model-task combinations**
  - **(1) Sample from 10M possible model-task combinations**
  - **(2) Rank the models** with BN = True / False
- How do we make it **scalable & convincing?**

Evaluating GNN Designs

- **Evaluating a design dimension**: Controlled random search
  - **a) Sample random model-task configurations**, perturb `BatchNorm = [True, False]`
  - **Here we control the computational budget for all the models**

(a) Controlled Random Search

GNN Design Space					GNN Task Space	
BatchNorm	Activation	…	Message layers	Layer Connectivity	Task level	dataset
True	relu	…	8	skip_sum	node	CiteSeer
False	relu	…	8	skip_sum	node	CiteSeer
True	relu	…	2	skip_cat	graph	BZR
False	relu	…	2	skip_cat	graph	BZR
...						
True	prelu	…	4	stack	graph	scale free
False	prelu	…	4	stack	graph	scale free

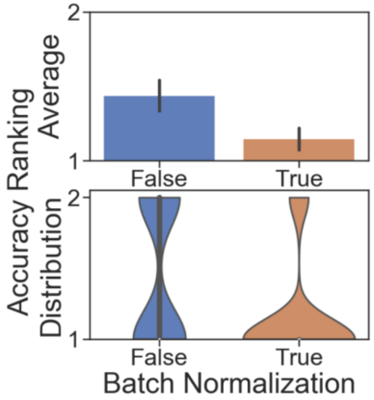
Evaluating GNN Designs

- **b) Rank** `BatchNorm = [True, False]` by their performance (lower ranking is better)
- **c) Plot Average / Distribution of the ranking** of `BatchNorm = [True, False]`

(b) Rank Design Choices by Performance

GNN Design Space	Experimental Results	
BatchNorm	Val. Accuracy	Design Choice Ranking
True	0.75	1
False	0.54	2
True	0.88	1 (a tie)
False	0.88	1 (a tie)
True	0.89	1
False	0.36	2

(c) Ranking Analysis



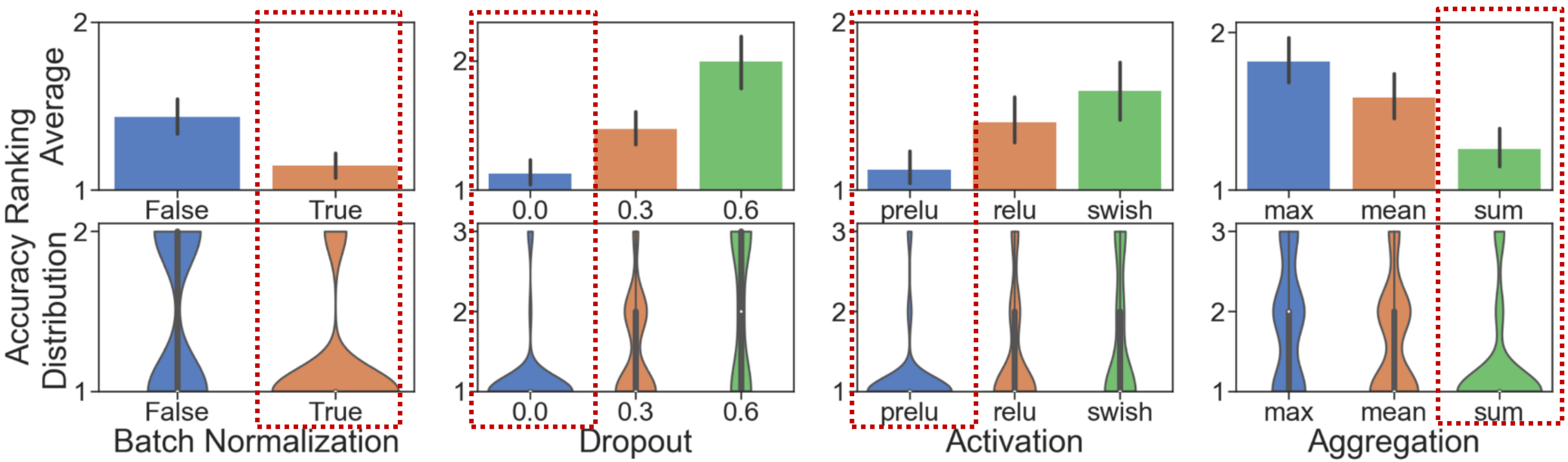
- **Summary: Convincingly evaluate any new design dimension**, e.g., evaluate a new GNN layer we propose

Results 1: A Guideline for GNN Design

- Certain design choices exhibit clear advantages
  - Intra-layer designs:

Explanation:
GNNs are hard to optimize

Explanation:
This is our new finding!



Explanation:
GNNs experience
underfitting more often

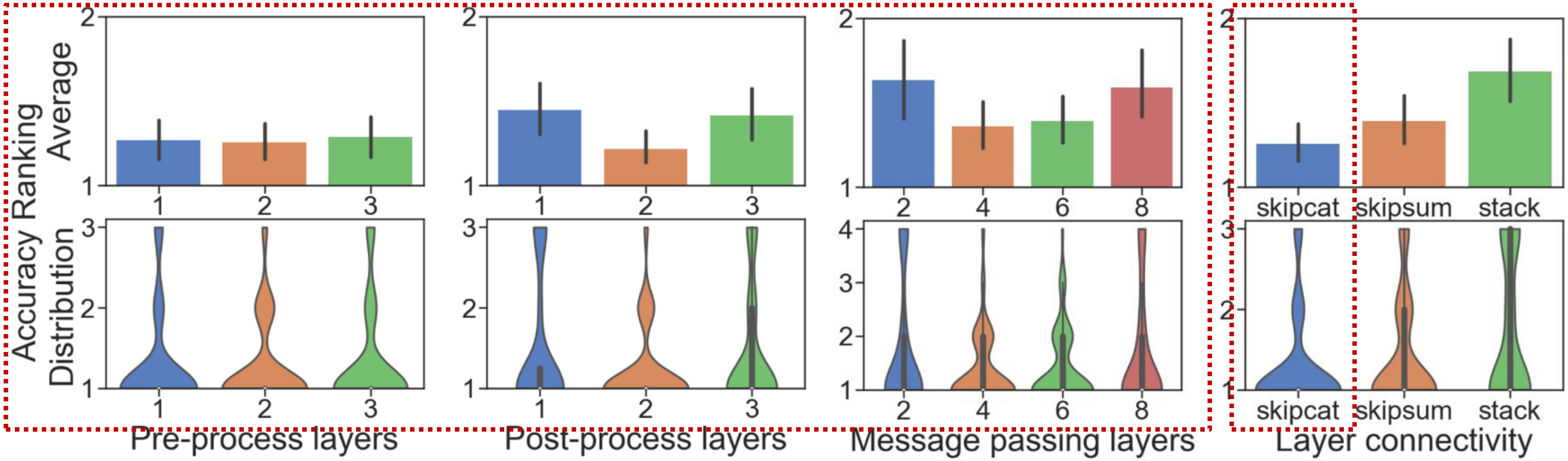
Explanation:
Sum is the most expres
aggregator

Results 1: A Guideline for GNN Design

- Certain design choices exhibit clear advantages
  - Inter-layer designs

    Optimal number of layers is hard to decide

    Highly dependent on the task

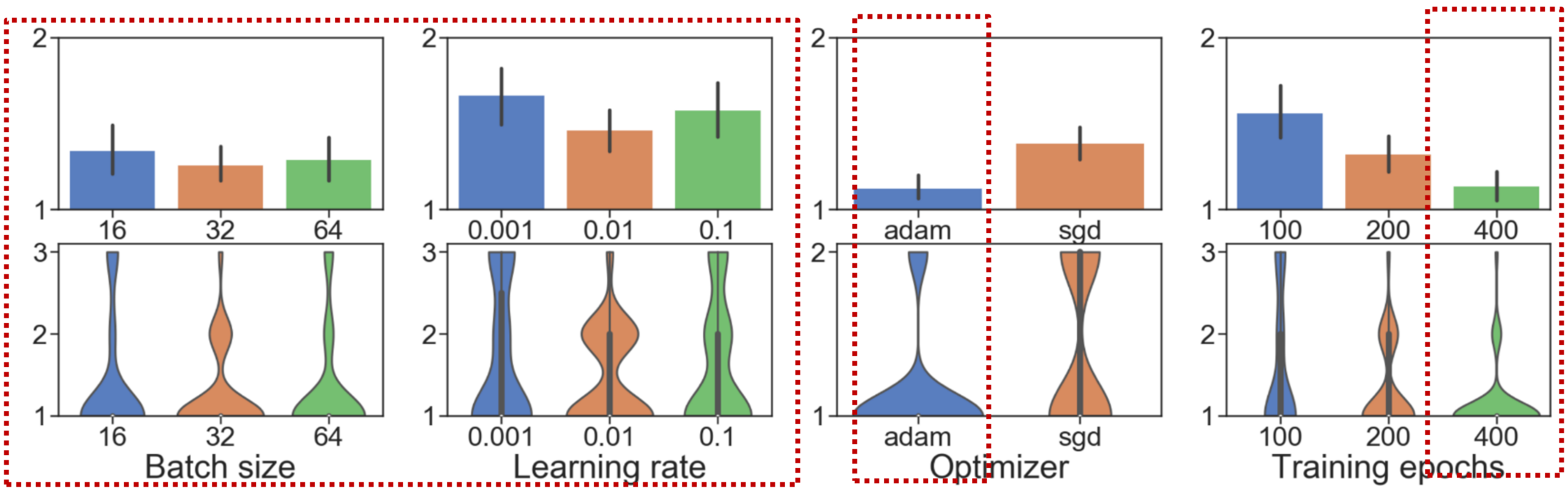


Explanation:
Skip connection enable hierarchical node representation

Results 1: A Guideline for GNN Design

- Certain design choices exhibit clear advantages
  - Learning configurations

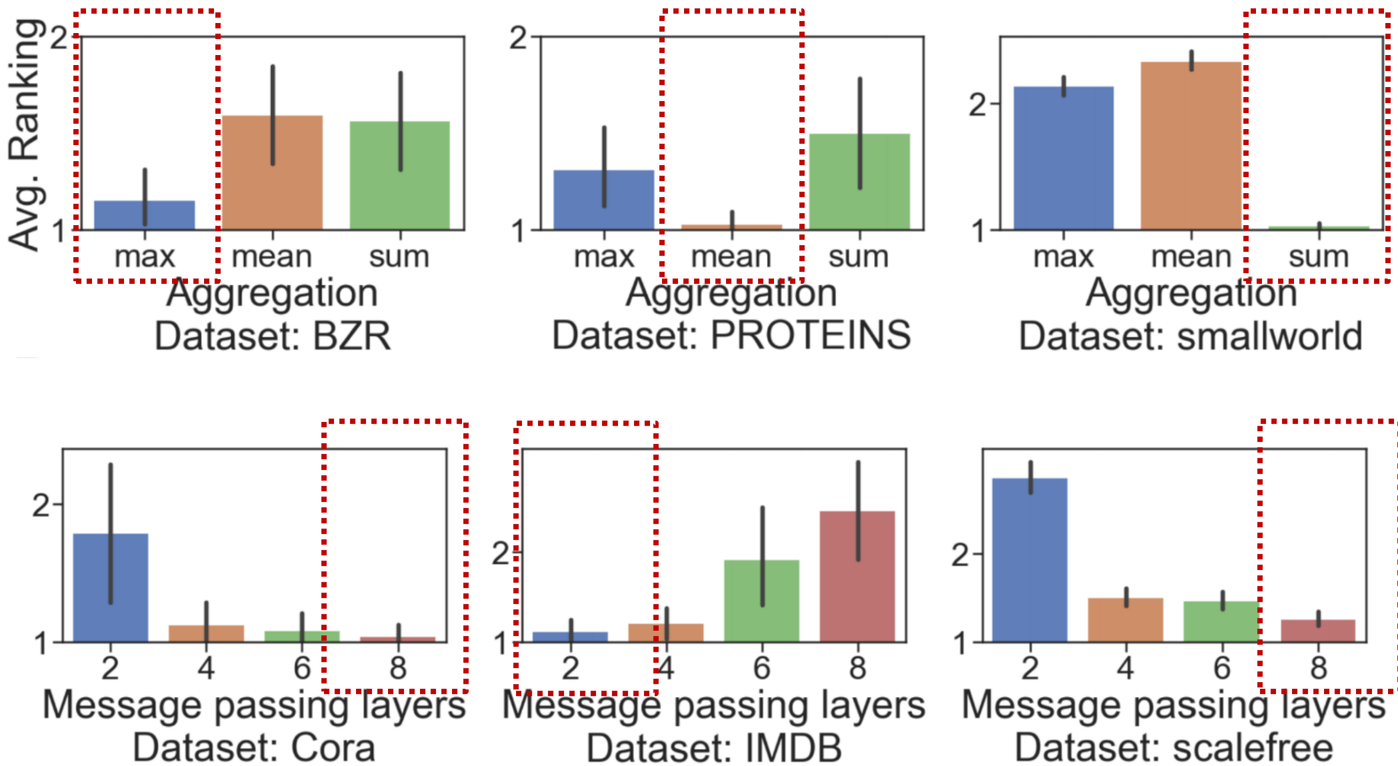
Optimal batch size and learning rate is hard to decide
Highly dependent on the task



Explanation:
Adam is more robust
More training epochs is better

Results 2: Understanding GNN Tasks

- **Best GNN designs in different tasks vary significantly**
  - Motivate that studying a task space is crucial

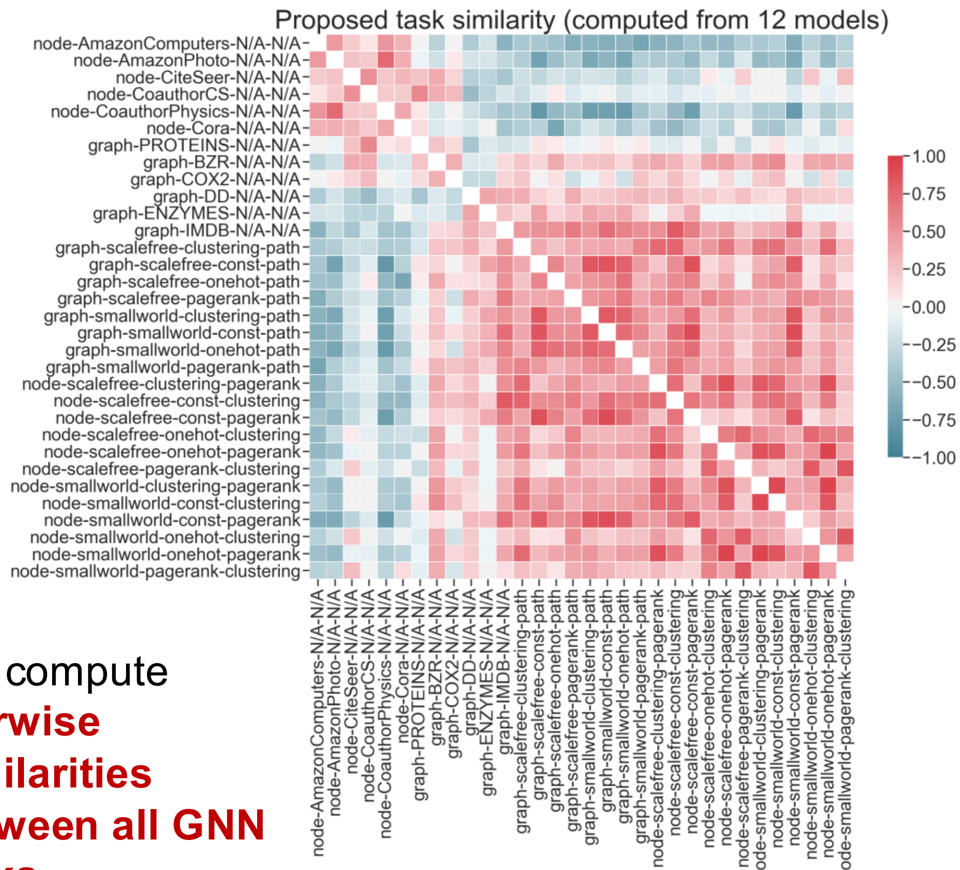


Results 2: Understanding GNN Tasks

- **Build a GNN task space**

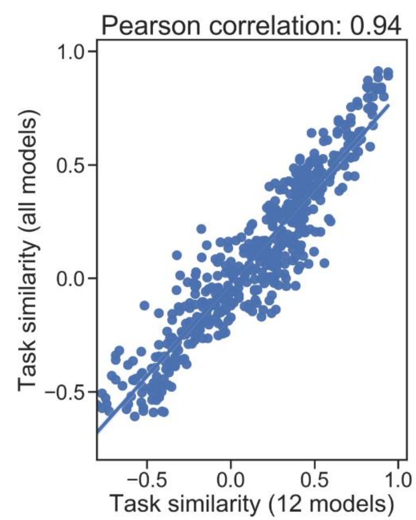
Recall how we **compute task similarity**

	Anchor Model Performance ranking					Similarity to Task A
Task A	M_1	M_2	M_3	M_4	M_5	1.0
Task B	M_1	M_3	M_2	M_4	M_5	0.8
Task C	M_5	M_1	M_4	M_3	M_2	-0.4



Format of tasks: Level-Dataset-Feature-Label

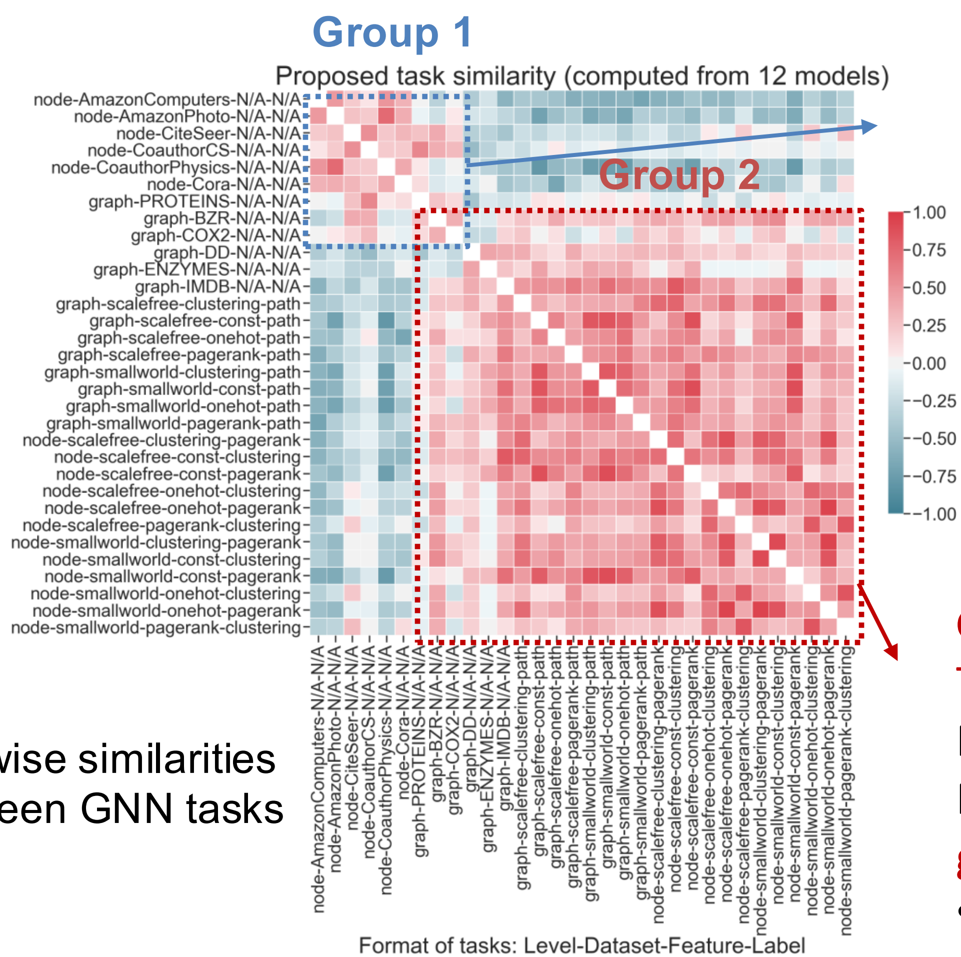
We compute **pairwise similarities between all GNN tasks**



Task similarity computation is cheap:
Using **12 anchor models** is a good approximation!

Results 2: Understanding GNN Tasks

- **GNN task space is informative**



Pairwise similarities between GNN tasks

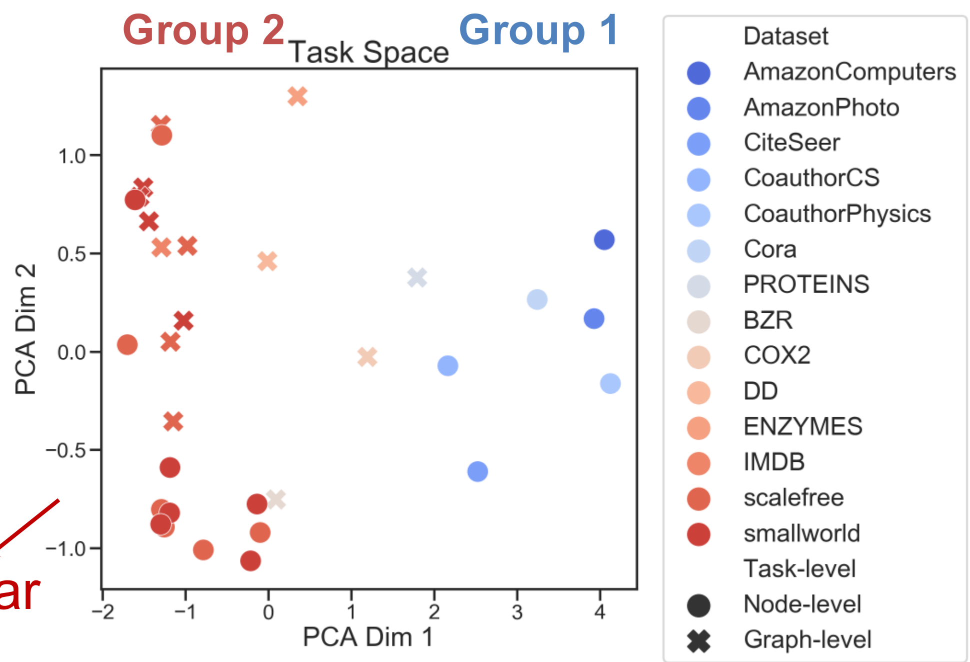
Group 1:
Tasks rely on **feature information**
Node/graph classification tasks, where input graphs have high-dimensional features
- Cora graph has 1000+ dim node feature

Group 2:
Tasks rely on **structural information**
Nodes have few features
Predictions are highly dependent on graph structure
- Predicting clustering coefficients

Results 2: Understanding GNN Tasks

- **GNN task space is informative**

PCA



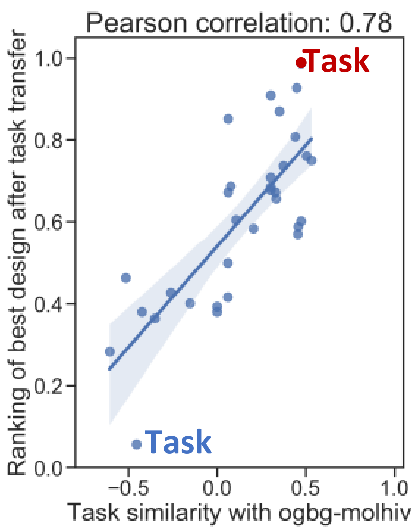
Similar tasks have similar best architectures

Best GNN Designs Found in Different Tasks

	Pre layers	MP layers	Post layers	Connectivity	AGG
Task A	2	8	2	skip-sum	sum
Task B	1	8	2	skip-sum	sum
Task C	2	6	2	skip-cat	mean

Results 3: Transfer to Novel Tasks

- **Case study**: generalize best models to **unseen** OGB ogbg-molhiv task:
  - **ogbg-molhiv is unique from other tasks**: 20x larger, imbalanced (1.4% positive) and requires out-of-distribution generalization
- **Concrete steps for applying to a novel task:**
  - **Step 1:** Measure 12 anchor model performance on the new task
  - **Step 2:** Compute similarity between the new task and existing tasks
  - **Step 3:** Recommend the best designs from existing tasks with high similarity



Results 3: Transfer to Novel Tasks

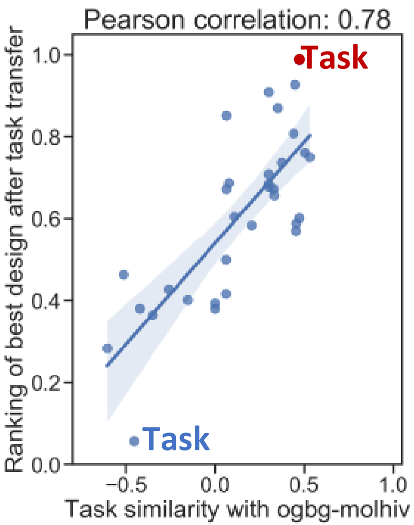
- **Our task space can guide best model transfer to novel tasks!**

We pick 2 tasks:

Task A: Similar to OGB
Task B: Not similar to OGB

Findings:

Transfer the best model from Task A achieves SOTA on OGB
Transfer the best model from Task B performs badly on OGB

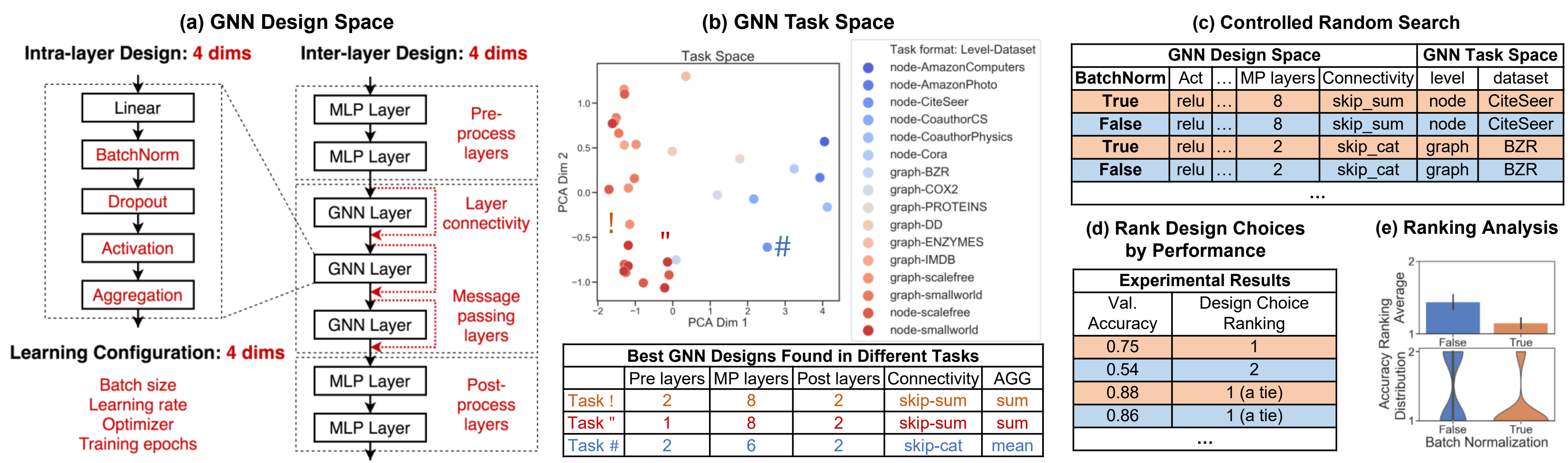


	Task A: graph-scalefree-const-path	**Task B: node-CoauthorPhysics**
Best design in our design space	(1, 8, 3, skipcat, sum)	(1, 4, 2, skipcat, max)
Task Similarity with ogbg-molhiv	**0.47**	**-0.61**
Performance after transfer to ogbg-molhiv	***0.785***	***0.736***

Previous SOTA: 0.771

GNN Design Space: Summary

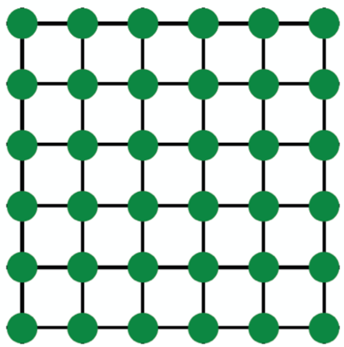
- Systematic investigation of:
  - **General guidelines for GNN design**
  - **Understandings of GNN tasks**
  - **Transferring best GNN designs across tasks**
  - **GraphGym:** Easy-to-use **code platform for GNN**

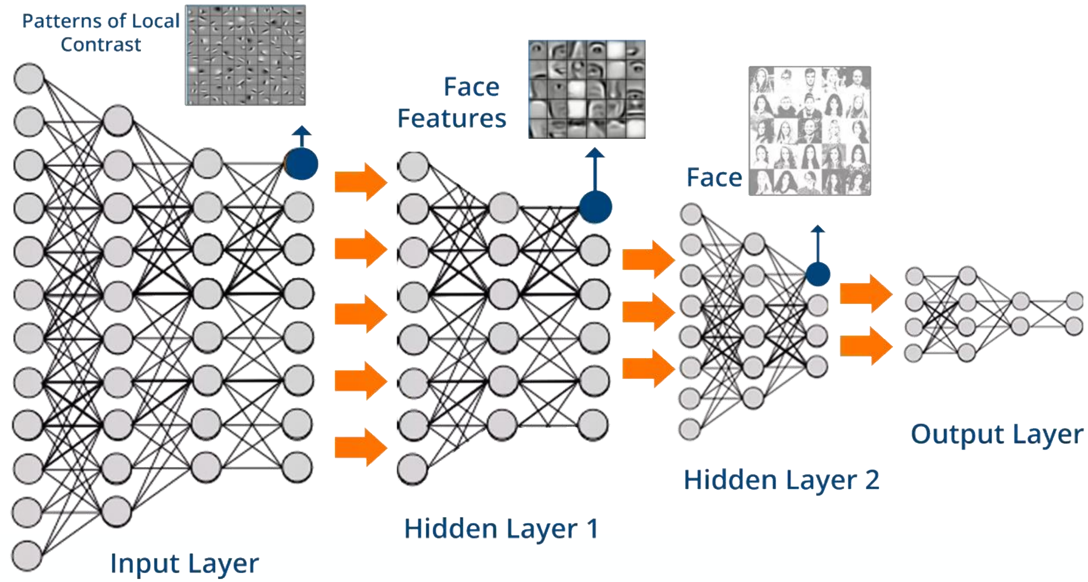


CS224W: Wrap-Up

CS224W: Machine Learning with Graphs
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

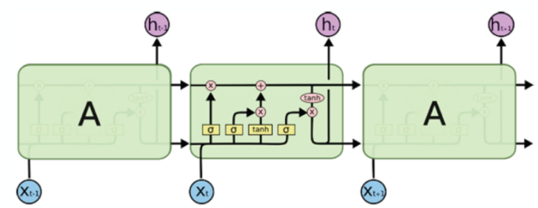
Modern ML Toolbox





Images

Text/Speech



Modern deep learning toolbox is designed for simple sequences & grids

This Course

How can we develop neural networks that are much more broadly applicable?

<u>Graphs</u> are the new frontier of deep learning

Graphs and Relational Data

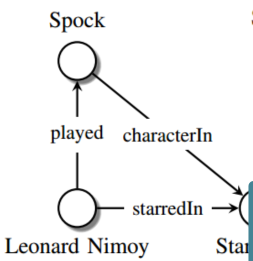
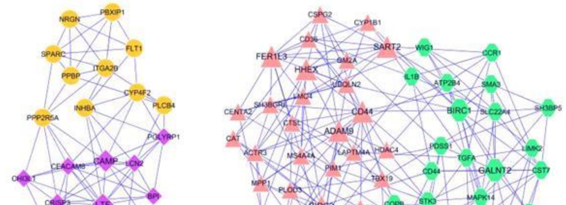
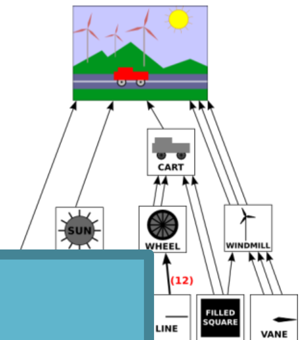


Image cred

Know





ath.hws.edu

Graphs

Main question:

How do we take advantage of relational structure for better prediction?

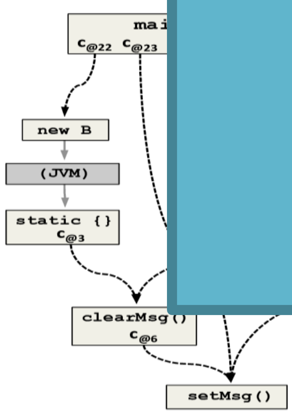


Image credit: ResearchGate

Code Graphs

Image credit: MDPI

Molecules

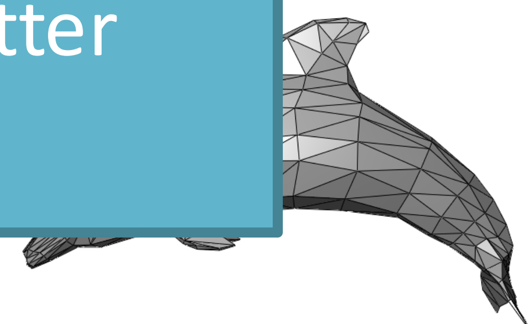
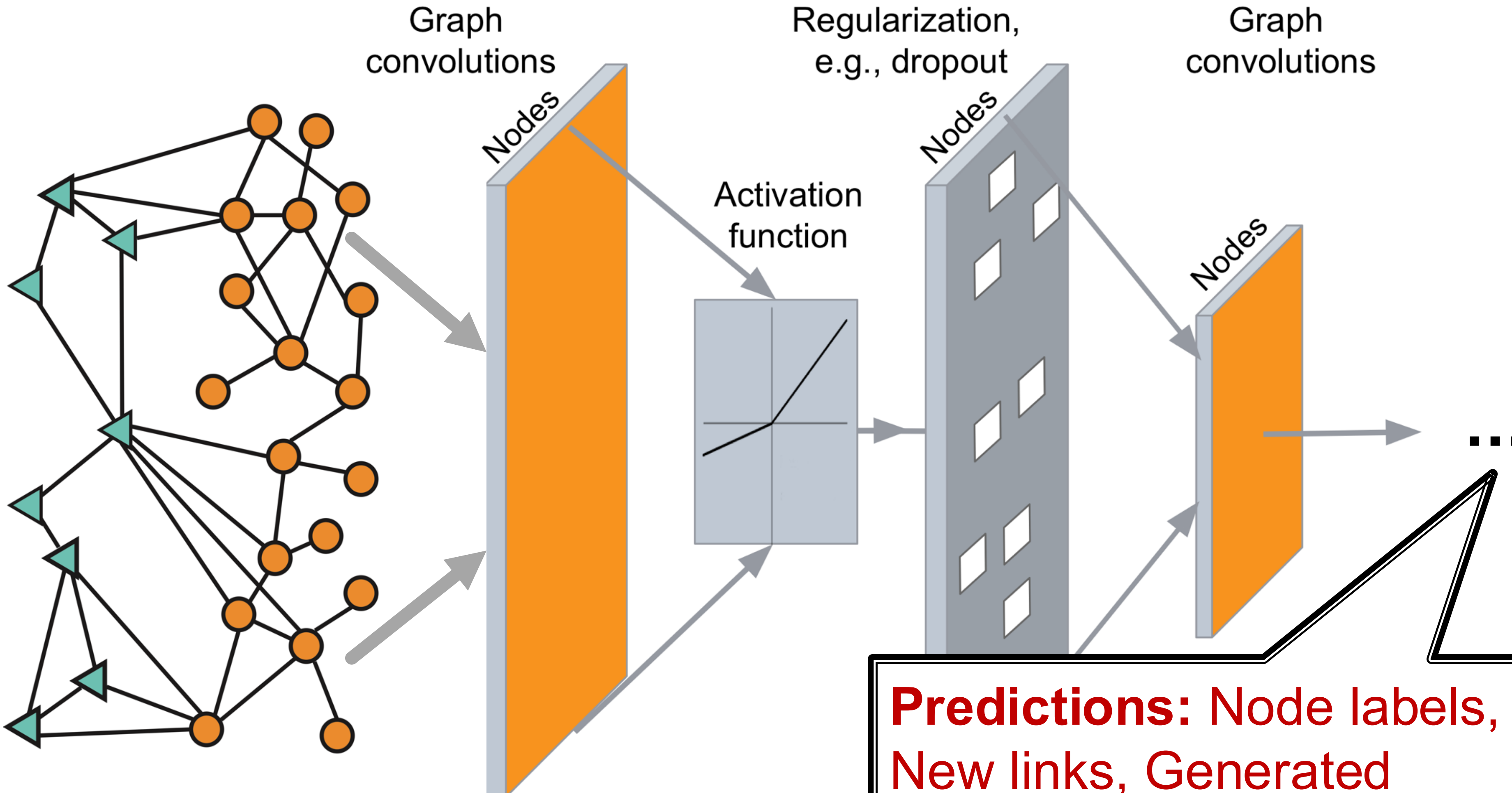


Image credit: Wikipedia

3D Shapes

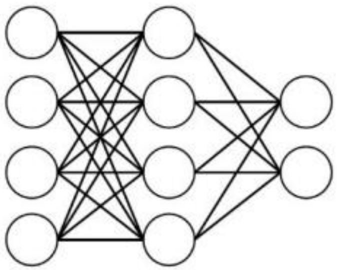
CS224W: Deep Learning in Graphs



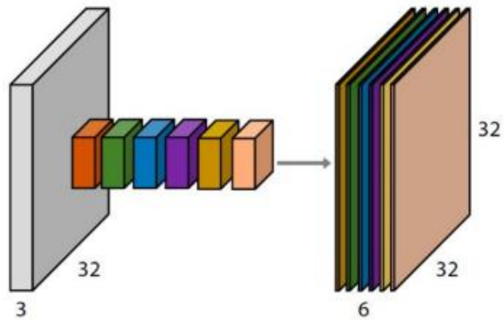
Predictions: Node labels, New links, Generated graphs and subgraphs

Input: Network

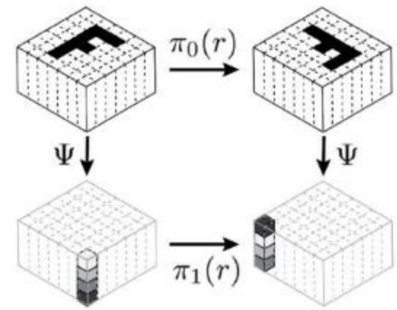
Models of Interest: Invariances



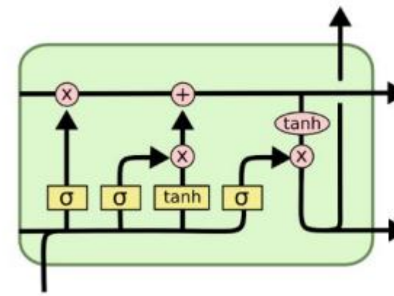
Perceptrons
Function regularity



CNNs
Translation

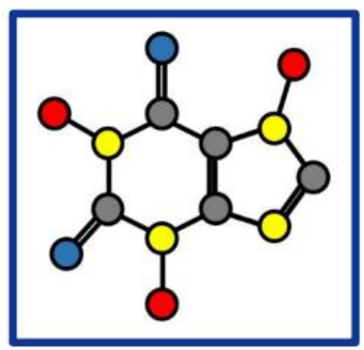


Group-CNNs
Translation+Rotation,
Global groups



LSTMs
Time warping

DeepSets / Transformers
Permutation



GNNs
Permutation

Intrinsic CNNs
Isometry / Gauge choice

The Bottom Line

- **There is exciting relational structure in many many real-world problems**
  - Molecules/Proteins as strings vs. graphs
  - Relational databases (primary-foreign key structure)
- **Identifying and harnessing this relational structure leads to better predictions**
  - AlphaFold
  - Biomedicine
  - Recommender systems

You learned a lot!

- **Theory:**
  - Models, architectures, approaches
- **Practice:**
  - Collab notebooks
  - Homeworks
- **Creative research:**
  - Course project
- **The real-world use cases and applications**

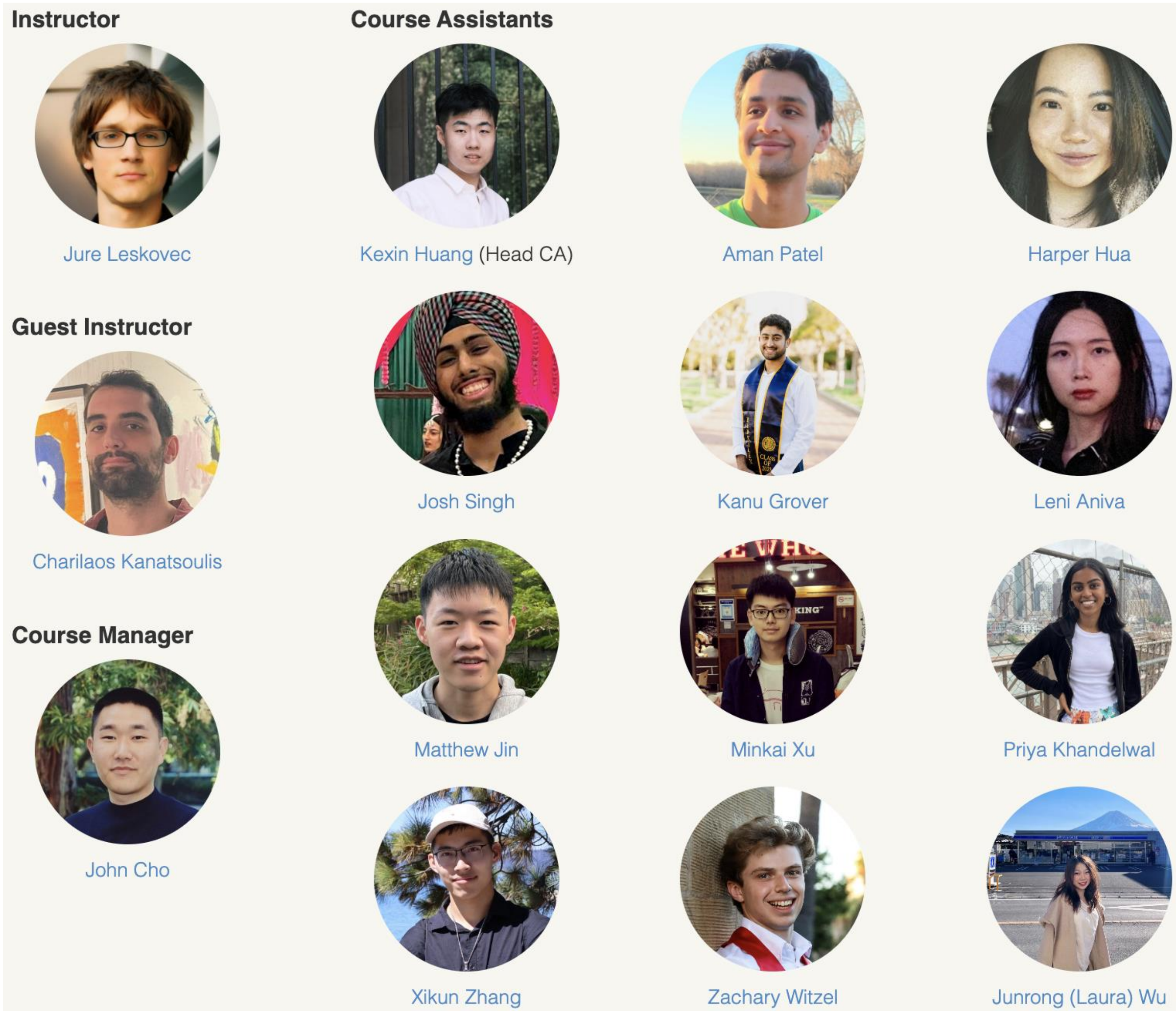
What Next?

- **Project write-ups:**
  - Thurs Dec. 12, Midnight **(11:59PM)** Pacific Time

No late days!

- **Courses:**
- **CS246: Mining Massive Datasets (Winter)**
  - Data Mining & Machine Learning for big data
    - (big==doesn't fit in memory/single machine)
    - Fast clever algorithms for real-world problems
    - Distributed data processing frameworks: MapReduce, Spark

Thank you, team!!!



I am very proud of everyone!

- **You Have Done a Lot!!!**
- **And (hopefully) learned a lot!!!**
  - Answered questions and proved many interesting results
  - Implemented a number of methods
  - **And are doing excellently on the project!**

Thank You for the Hard Work!!!