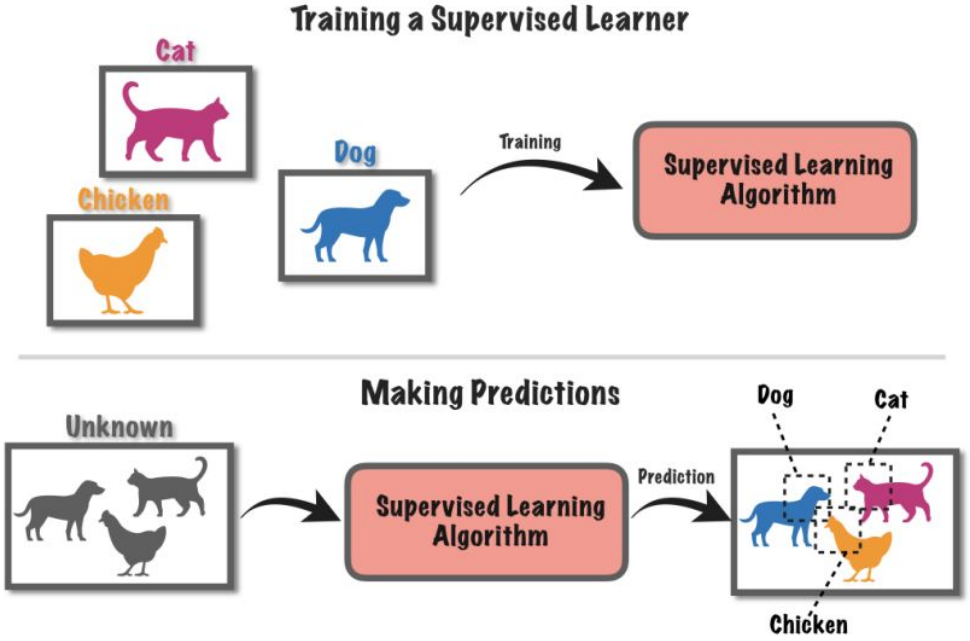
TEXAS A&M UNIVERSITY
Engineering

TOWARDS SELF-SUPERVISED LEARNING AND EXPLAINING OF DEEP MODELS

Yaochen Xie

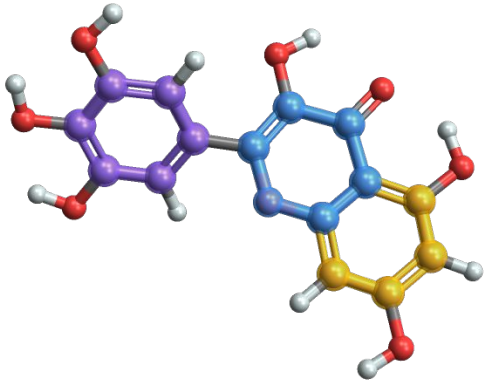
Supervised Learning

- The most common learning setting for deep models
- Requires data with paired labels



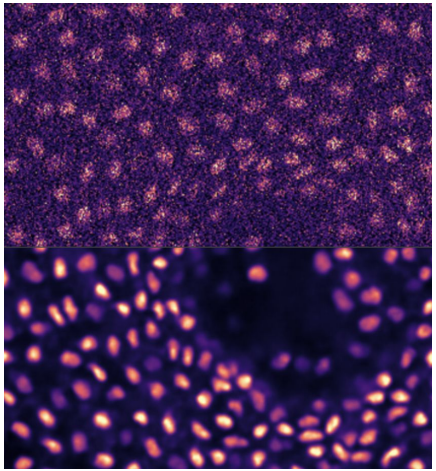
https://towardsdatascience.com/supervised-vs-unsupervised-learning-in-2-minutes-72dad148f242

Labels Are Expensive/Unavailable When ...



Drug Discovery

https://www.wolfram.com/language/12/molecular-structure-and-computation/molecule-graphs.html.en?product=mathematica



Augmented Microscopy

Weigert, Martin, et al. "Content-aware image restoration: pushing the limits of fluorescence microscopy." Nature methods 15.12 (2018): 1090.

Industrial Large-Scale Models

https://www.fincash.com/l/basics/gafam-stocks

How about if we…

Drug Discovery

Learn from a huge amount of unlabeled molecules?

Augmented Microscopy

Perform denoising without ground-truth images?

Industrial Large-Scale Models

Learn an encoding model for general purpose?

How about if we...

TEXAS A&M UNIVERSITY
Engineering

Drug Discovery

Learn from a huge amount of unlabeled molecules?

Augmented Microscopy

Perform denoising without ground-truth images?

Industrial Large-Scale Models

Learn an encoding model for general purpose?

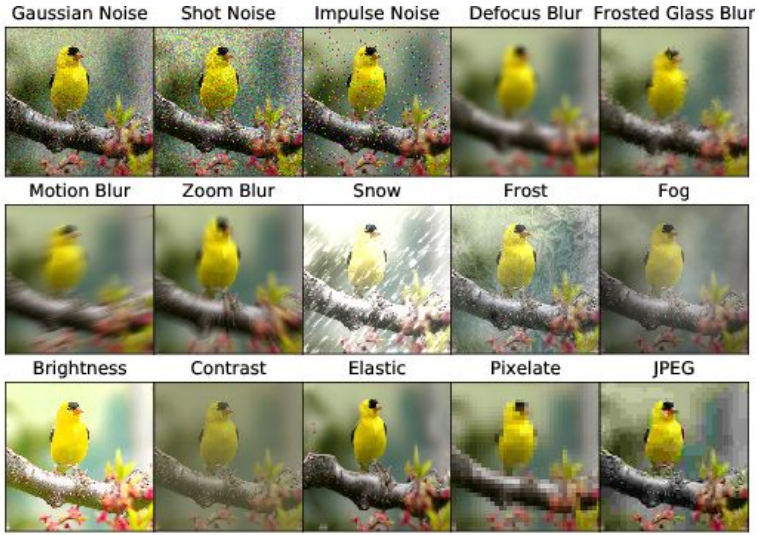
We can involve the **supervision** from data it**self**!

Self-supervised Learning

- Theory-grounded self-supervised learning
  - Self-supervised image denoising (NeurIPS'20)
  - Self-supervised graph representation learning (ICML'22)
- Self-supervised explainer models (NeurIPS'22)
- Future directions of SSL

Theory-grounded self-supervised learning

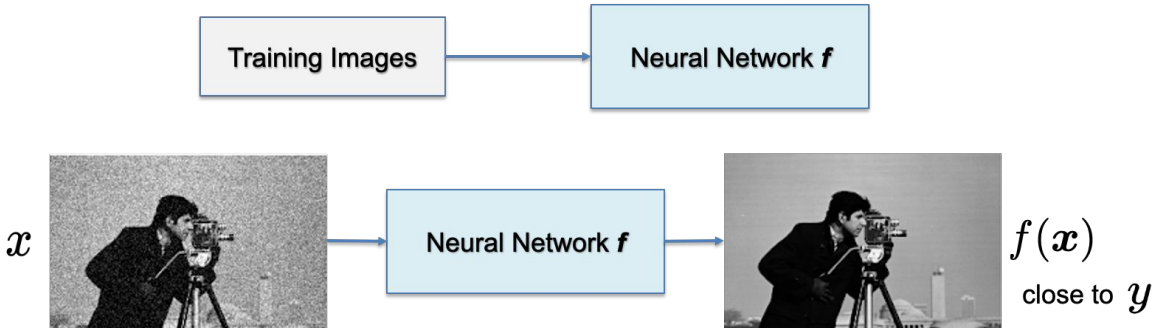
- A general goal of learning: to learn good representation to perform given downstream task
- What are good representations?
  - Informative
  - Robust to noise
- Supervised v.s. Self-supervised



Hendrycks and Dietterich. "Benchmarking neural network robustness to common corruptions and perturbations." ICLR 2019.

Supervised image denoising

- The most common setting to training a learning-based denoising model
  - Samples of paired noisy-clean images (x_i, y_i) are given as the training data.
  - Training with the supervised loss, *e.g.*, MSE.



***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Supervised image denoising

- The most common setting to training a learning-based denoising model
  - Samples of paired noisy-clean images (x_i, y_i) are given as the training data.
  - Training with the supervised loss, *e.g.*, MSE.

Clean images are usually unavailable in most real world scenarios!

***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Self-Supervised image denoising

Self-supervised denoising: an even more strict setting of available data

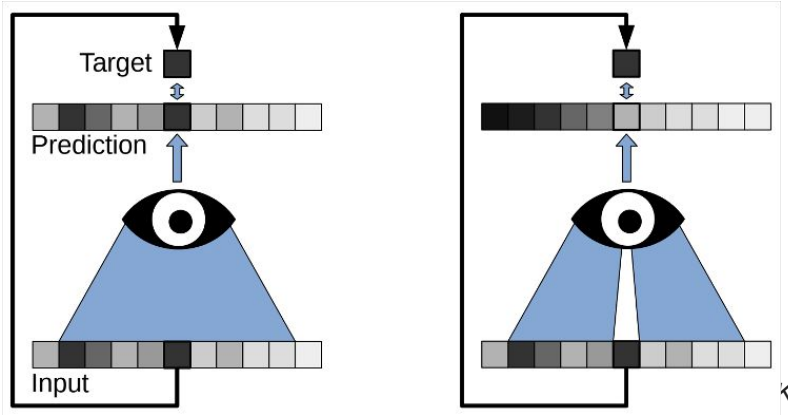
- Only (unpaired) noisy images x are given in the training dataset.
- How to design a task to utilize x as both the source and the target, with certain restrictions to avoid learning identical mapping.

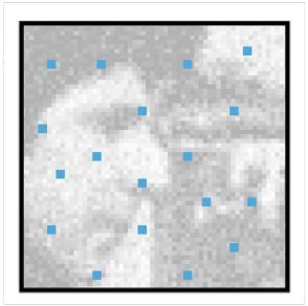
***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Self-Supervised image denoising

Self-supervised denoising: an even more strict setting of available data

- Only (unpaired) noisy images x are given in the training dataset.
- Blind-spot denoising





Krull A et al. Noise2void-learning denoising from single noisy images. CVPR 2019.

Self-Supervised image denoising

Blind-spot denoising

- Theoretical framework: a relationship between losses

$$\mathbb{E}_x \|f(\boldsymbol{x}) - \boldsymbol{x}\|^2 = \mathbb{E}_{x,y} \|f(\boldsymbol{x}) - \boldsymbol{y}\|^2 + \mathbb{E}_{x,y} \|\boldsymbol{x} - \boldsymbol{y}\|^2 - 2\langle f(\boldsymbol{x}) - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y}\rangle$$

Self-supervised loss for blind-spot-based methods

Supervised loss

Is zero when assuming f is "$\mathcal{J}$-invariant"

- Basic idea: using a self-supervised loss to approximate the supervised denoising.

***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Self-Supervised image denoising

Blind-spot denoising

- Theoretical framework: a relationship between losses

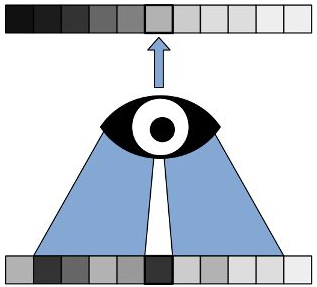
$$\mathbb{E}_x \|f(x) - x\|^2 = \mathbb{E}_{x,y} \|f(x) - y\|^2 + \mathbb{E}_{x,y} \|x - y\|^2 - 2\langle f(x) - y, x - y \rangle$$

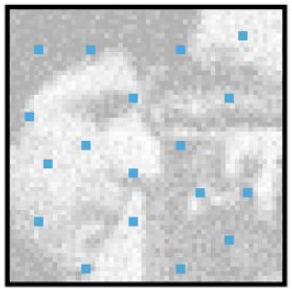
Self-supervised loss for blind-spot-based methods

Supervised loss

Is zero when assuming f is "$\mathcal{J}$-invariant"

- Issues?





***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Noise2Same

- Trains the denoising model without assuming "J -invariant" f.
- Utilizes information of center pixel without knowing the noise model or post-processing. And hence can be applied in broader scenarios.

$$\mathbb{E}_x \|f(x) - x\|^2 = \mathbb{E}_{x,y} \|f(x) - y\|^2 + \mathbb{E}_{x,y} \|x - y\|^2 - 2\langle f(x) - y, x - y \rangle$$

Part of the *Self-supervised loss in Noise2Same*

Supervised loss

*We **bound this term** in a self-supervised fashion, instead of assuming "$\mathcal{J}$-invariant" and making it zero*

***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Noise2Same

- We derive a self-supervised upper bound of the MSE loss

Theorem 1. *Consider a normalized noisy image $\boldsymbol{x} \in \mathbb{R}^m$ (obtained by subtracting the mean and dividing by the standard deviation) and its ground truth signal $\boldsymbol{y} \in \mathbb{R}^m$. Assume the noise is zero-mean and i.i.d among all the dimensions, and let J be a subset of m dimensions uniformly sampled from the image $\boldsymbol{x}$. For any $f : \mathbb{R}^m \to \mathbb{R}^m$, we have*

$$\mathbb{E}_{x,y} \left\| f(\boldsymbol{x}) - \boldsymbol{y} \right\|^2 + \left\| \boldsymbol{x} - \boldsymbol{y} \right\|^2 \leq \mathbb{E}_x \left\| f(\boldsymbol{x}) - \boldsymbol{x} \right\|^2 + 2m \, \mathbb{E}_J \left[\frac{\mathbb{E}_x \left\| f(\boldsymbol{x})_J - f(\boldsymbol{x}_{J^c})_J \right\|^2}{|J|} \right]^{1/2} \quad (6)$$

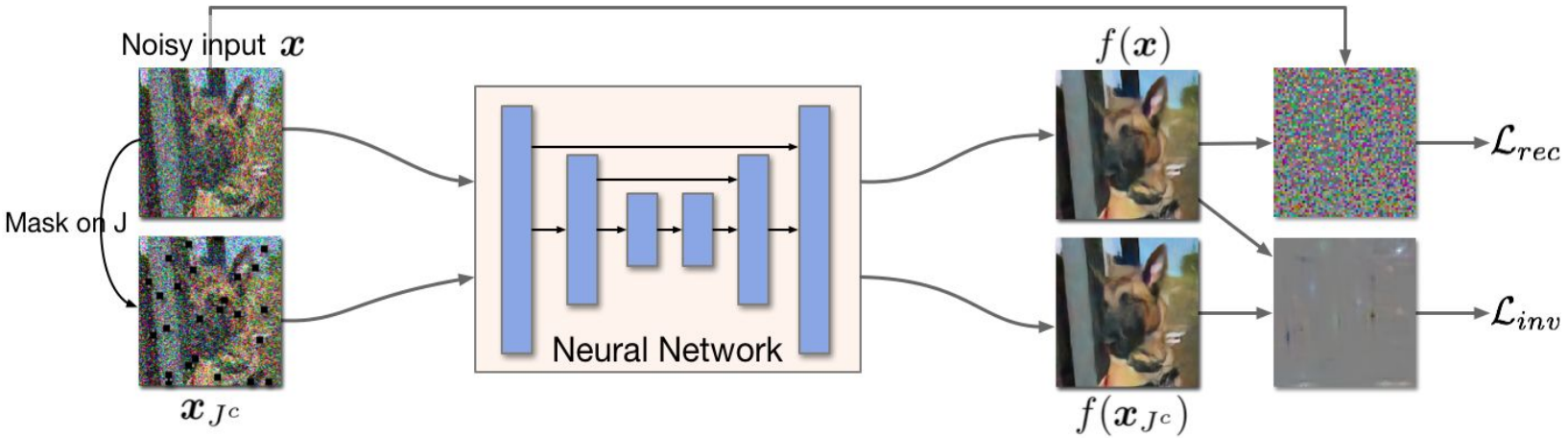
Supervised loss

The self-supervised bound as the new loss

***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Noise2Same

- It is simple to implement the self-supervised bound $L = L_{rec} + \lambda_{inv}\sqrt{L_{inv}}$



***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Noise2Same

How does the invariance term work?

- We derive a tighter upper bound assuming additive noise with known variance.
- The invariance term controls how much information is utilized from center pixels. When the noise is stronger, less center pixel information is used.

Theorem 2. *Given a noisy image $\boldsymbol{x} \in \mathbb{R}^m$ and its ground truth signal $\boldsymbol{y}$, assuming the noise is independent among all dimensions, and further assuming the noise is additive with standard deviation σ and zero-mean, letting J be a subset of m dimensions uniformly sampled from the image $\boldsymbol{x}$, we have*

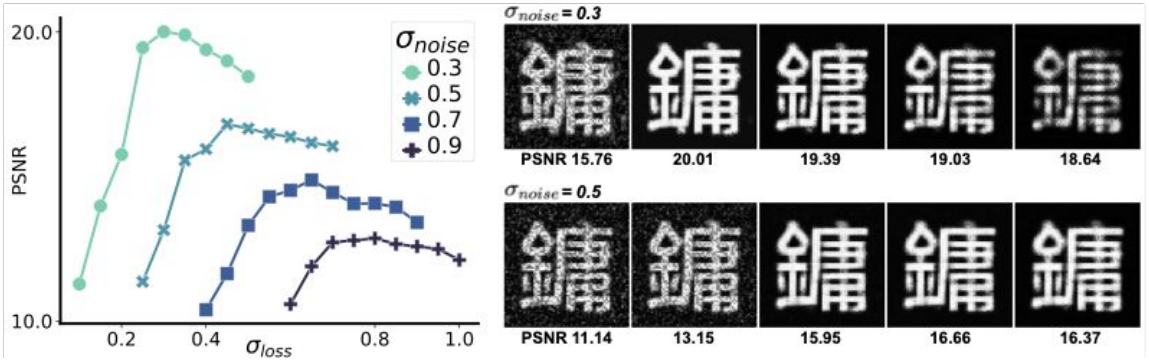
$$\mathbb{E}_{x,y} \|f(\boldsymbol{x}) - \boldsymbol{y}\|^2 + \|\boldsymbol{x} - \boldsymbol{y}\|^2 \leq \mathbb{E}_x \|f(\boldsymbol{x}) - \boldsymbol{x}\|^2 + 2m\sigma \, \mathbb{E}_J \left[\mathbb{E} \|f(\boldsymbol{x})_J - f(\boldsymbol{x}_{J^c})_J\|^2 / |J| \right]^{1/2}$$

***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Noise2Same

How does the invariance term work?

- We derive a tighter upper bound assuming additive noise with known variance.
- The invariance term controls how much information is utilized from center pixels. When the noise is stronger, less center pixel information is used.



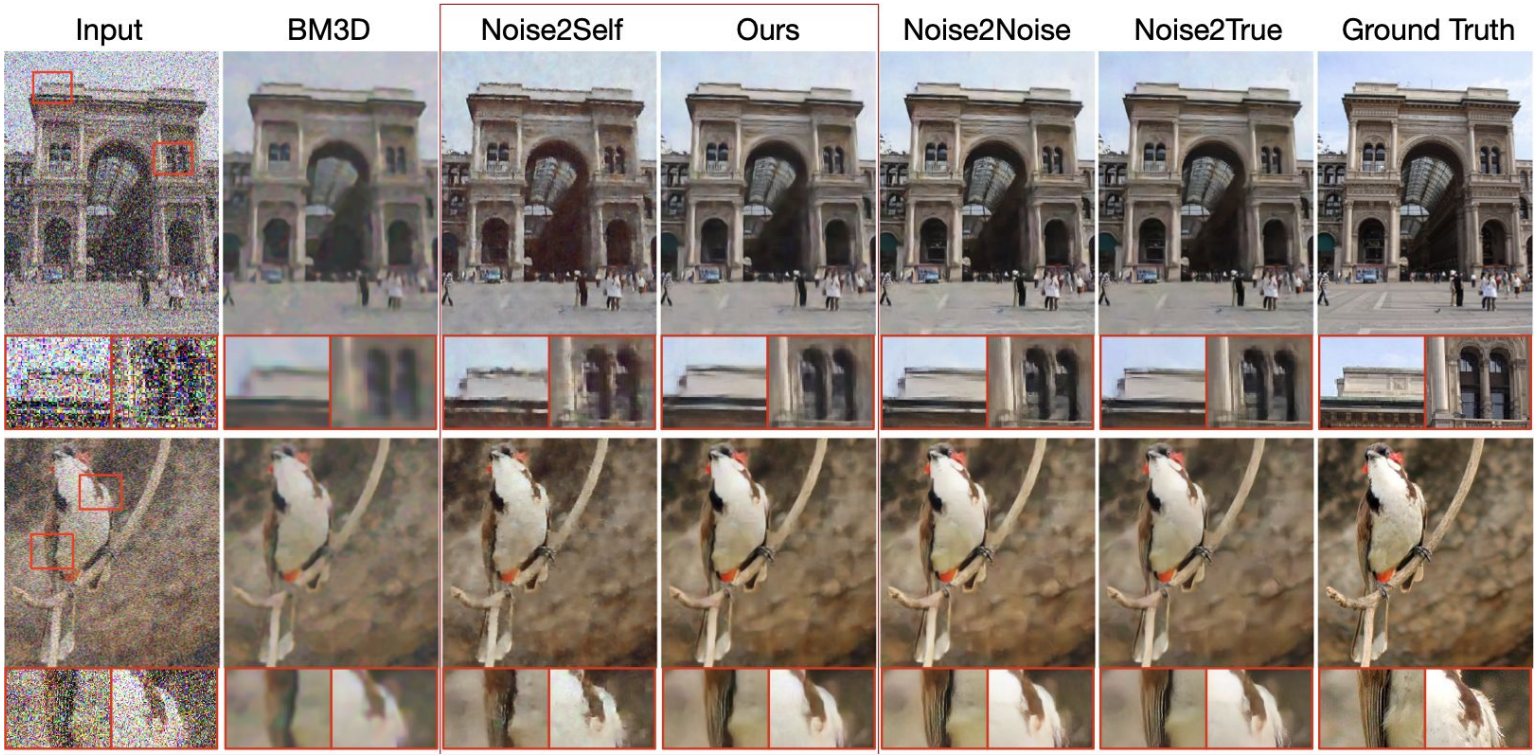
***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Experimental Results

	Methods	Datasets			
		ImageNet	HànZì	Planaria	BSD68
Traditional	Input	9.69	6.45	21.52 / 21.09 / 20.82	20.19
	NLM [3]	18.04	8.41	25.80 / 24.03 / 21.62	22.73
	BM3D [5]	18.74	10.90	-	28.59
Supervised	Noise2True	23.39	15.66	31.57 / 30.15 / 28.13	29.06
	Noise2Noise [13]	23.27	14.30	-	28.86
Self-Supervised + noise model	Laine et al. [12]	-	-	-	28.84
Self-Supervised	Laine et al. [12]	20.89	10.70	-	27.15
	Noise2Void [10]	21.36	13.72	25.84 / 23.57 / 21.60	27.71
	Noise2Self-Noise [1]	20.38	13.94	27.58 / 24.83 / 21.83	26.98
	Noise2Self-Donut [1]	8.62	13.29	27.63 / 24.72 / 21.73	**28.20**
	Noise2Same	**22.26**	**14.38**	**29.48 / 26.93 / 22.41**	27.95

***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

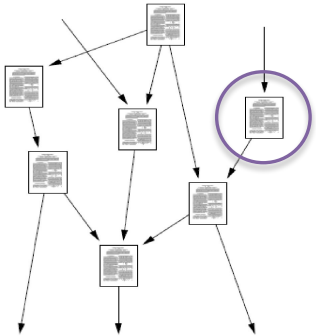
Experimental Results



***Xie** et al., Noise2Same: Optimizing a self-supervised bound for image denoising. NeurIPS'20.*

Self-supervised graph representation learning

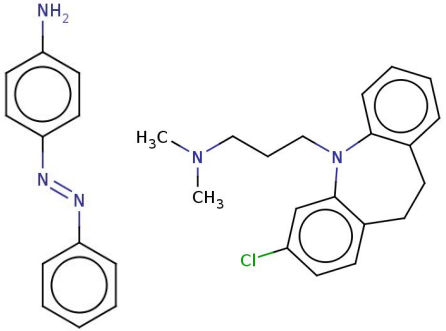
- Graph data



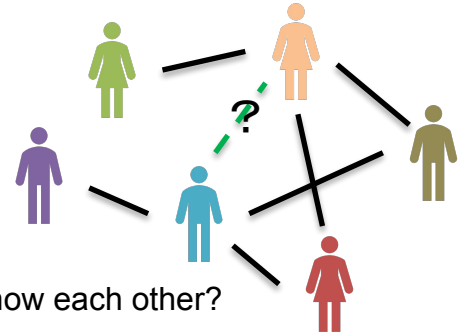
Computer science?

Math?

Chemistry?



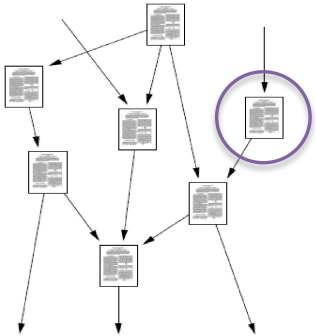
Toxic/Non-toxic?



Know each other?

Self-supervised graph representation learning

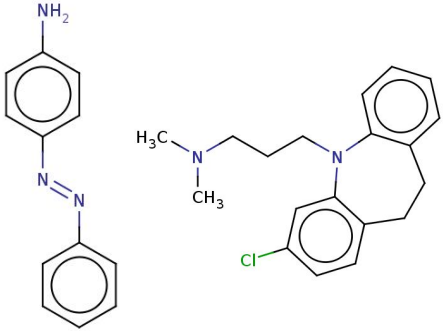
- Graph data



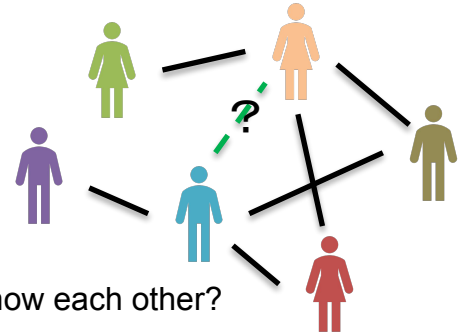
Computer science?

Math?

Chemistry?



Toxic/Non-toxic?

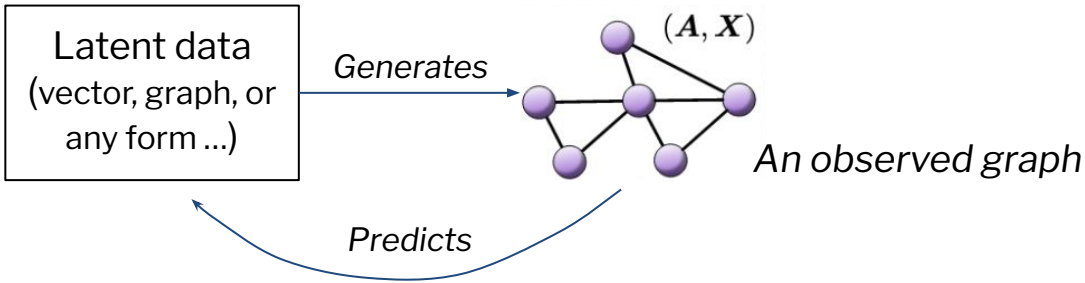


Know each other?

Data acquisition is even more challenging for graph data.

Key Idea: Latent Data

- We consider the concept latent data, where any observed graph $G = (A, X)$ is generated from a corresponding latent data that determine its semantic.



***Xie** et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

Latent Graphs

- We consider the concept latent data, where any observed graph $G = (A, X)$ is generated from a corresponding latent data that determine its semantic.
- WLOG, we specifically consider latent data $G_\ell = (A, F)$ in graph-structure with the same connectivity and satisfying two assumptions (non-structural and unbiased noise).
- Theorems can be generalized with other distances when considering latent data in different forms.

***Xie** et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

Latent Graph Prediction

- We adopt the prediction/reconstruction of the latent graph to derive our predictive SSL task.

$$f^* = \arg\min_f \mathbb{E} \left\| f(\boldsymbol{A}, \boldsymbol{X}) - \boldsymbol{F} \right\|^2$$

- We derive a self-supervised upper bound for the above objective to eliminate the need of unknown ***F***

$$\mathbb{E}_{\boldsymbol{A},\boldsymbol{X},\boldsymbol{F}} \left[\left\| f(\boldsymbol{A}, \boldsymbol{X}) - \boldsymbol{F} \right\|^2 + \left\| \boldsymbol{X} - \boldsymbol{F} \right\|^2 \right] \leq \mathbb{E}_{\boldsymbol{A},\boldsymbol{X}} \left\| f(\boldsymbol{A}, \boldsymbol{X}) - \boldsymbol{X} \right\|^2 + 2\sigma |V| \, \mathbb{E}_J \left[\frac{\mathbb{E}_{\boldsymbol{A},\boldsymbol{X}} \left\| f_J(\boldsymbol{A}, \boldsymbol{X}) - f_J(\boldsymbol{A}, \boldsymbol{X}_{J^c}) \right\|^2}{|J|} \right]^{1/2}$$

Xie *et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

LaGraph Objectives

Node-level representation learning

Corollary 2.2. *Let $G = (\boldsymbol{A}, \boldsymbol{X})$ be a given graph, $G_{\mathcal{I}} = (\boldsymbol{A}, \boldsymbol{F})$ be its latent graph, $\mathcal{E}$ and $\mathcal{D}$ be a graph encoder and a prediction head (decoder) consisting of fully-connected layers. If the prediction head $\mathcal{D}$ is ℓ-Lipschitz continuous with respect to l_2-norm, we further have the following inequality,*

$$\mathbb{E}\left[\,\|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{F}\|^2 + \|\boldsymbol{X} - \boldsymbol{F}\|^2\,\right] \leq \mathbb{E}\,\|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{X}\|^2 + 2\sigma|V|\ell\,\mathbb{E}_J\left[\frac{\mathbb{E}\,\|\boldsymbol{H}_J - \boldsymbol{H}'_J\|^2}{|J|}\right]^{1/2}, \tag{3}$$

where $\boldsymbol{H} = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X})$ and $\boldsymbol{H}' = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X}_{J^c})$ denote the node embedding of the given graph and the masked graph, respectively, and $\boldsymbol{H}_J := \boldsymbol{H}[J, :]$ selects rows with indices in J.

Graph-level representation learning

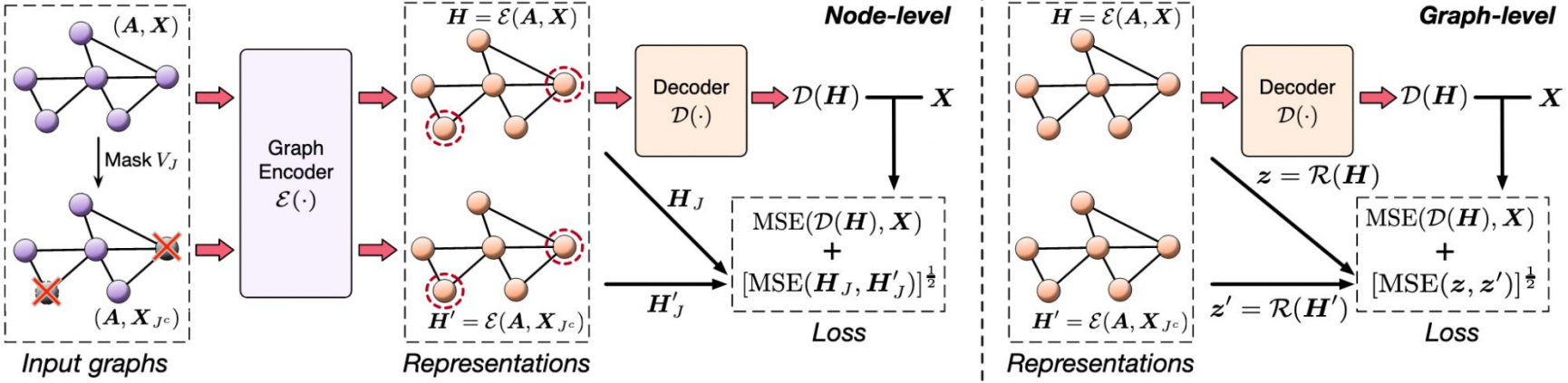
Corollary 2.3. *Let $G = (\boldsymbol{A}, \boldsymbol{X})$ be a given graph, $G_{\mathcal{I}} = (\boldsymbol{A}, \boldsymbol{F})$ be its hidden latent graph, $\mathcal{E}$ be a graph encoder, $\mathcal{R}$ be a readout function satisfying k-Bilipschitz continuity with respect to l_2-norm, and $\mathcal{D}$ be a prediction head (decoder). If the prediction head $\mathcal{D}$ is ℓ-Lipschitz continuous with respect to l_2-norm, we have the following inequality,*

$$\mathbb{E}\left[\,\|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{F}\|^2 + \|\boldsymbol{X} - \boldsymbol{F}\|^2\,\right] \leq \mathbb{E}\,\|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{X}\|^2 + 2\sigma|V|k\ell\,\mathbb{E}_J\left[\frac{\mathbb{E}\,\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{|J|}\right]^{1/2}, \tag{4}$$

where $\boldsymbol{z} = \mathcal{R}(\boldsymbol{H})$ and $\boldsymbol{z}' = \mathcal{R}(\boldsymbol{H}')$ denote the graph-level representations of the given graph and the masked graph, respectively.

***Xie** et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

The LaGraph Framework



***Xie** et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

The Information Bottleneck Principle

Intuition: a good representation (compression) of data should be

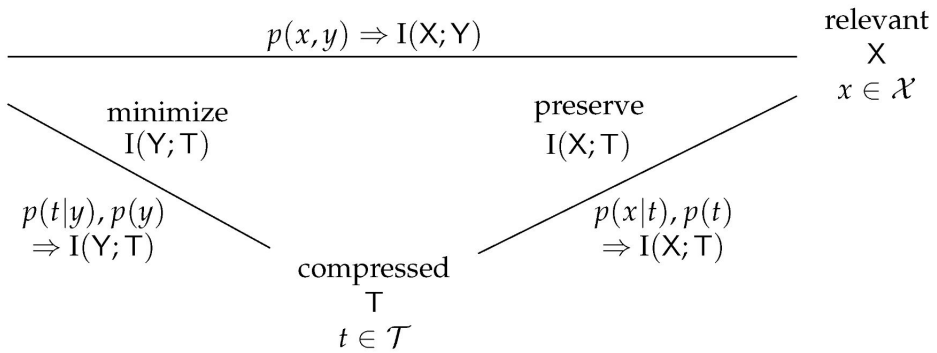
I. As informative as possible

II. Robust to noise in the observed data

$$\boldsymbol{T}^* = \arg\min_{\boldsymbol{T}} I(\boldsymbol{T};\boldsymbol{Y}) - \beta I(\boldsymbol{T};\boldsymbol{X})$$

$$\boldsymbol{T}^* = \arg\min_{\boldsymbol{T}} \big[H(\boldsymbol{T}) - H(\boldsymbol{T}|\boldsymbol{Y})\big] - \beta\big[H(\boldsymbol{T}) - H(\boldsymbol{T}|\boldsymbol{X})\big]$$

$$= \arg\min_{\boldsymbol{T}} H(\boldsymbol{T}|\boldsymbol{X}) - \lambda H(\boldsymbol{T}),$$



***Xie** et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

The Information Bottleneck Principle

Reconstruction term: Computed between original node features X, to ensure the learning of ***informative*** *representations.*

Invariance term: Computed between representations of original graph and masked graph respectively, to ensure the learning of ***non-trivial*** *representations (e.g., to avoid identical mapping).*

Corollary 2.2. *Let $G = (\boldsymbol{A}, \boldsymbol{X})$ be a given graph, $G_{\mathcal{I}} = (\boldsymbol{A}, \boldsymbol{F})$ be its latent graph, $\mathcal{E}$ and $\mathcal{D}$ be a graph encoder and a prediction head (decoder) consisting of fully-connected layers. If the prediction head $\mathcal{D}$ is ℓ-Lipschitz continuous with respect to l_2-norm, we further have the following inequality,*

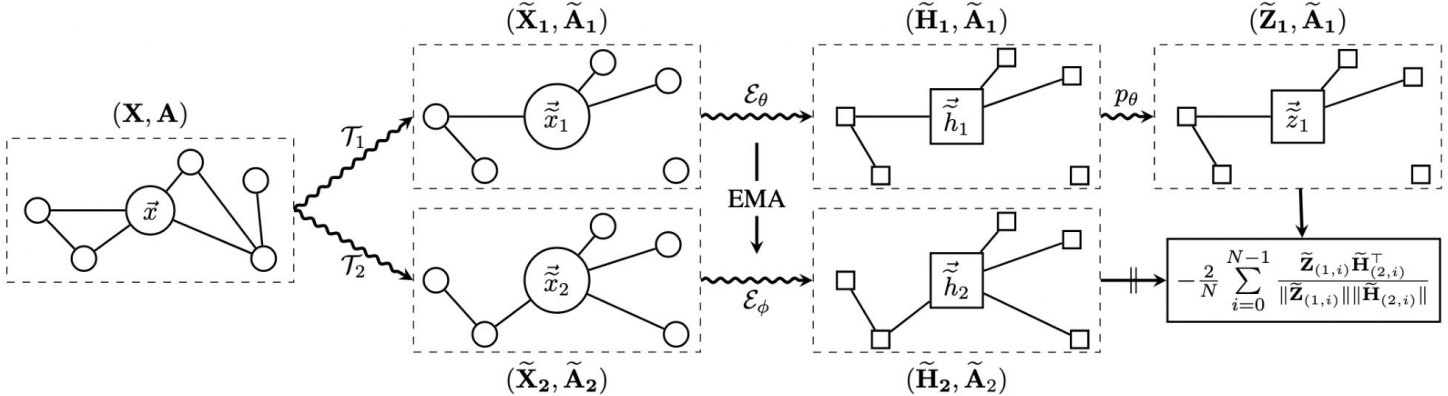
$$\mathbb{E}\left[\|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{F}\|^2 + \|\boldsymbol{X} - \boldsymbol{F}\|^2\right] \leq \mathbb{E}\|\mathcal{D}(\boldsymbol{H}) - \boldsymbol{X}\|^2 + 2\sigma|V|\ell\,\mathbb{E}_J\left[\frac{\mathbb{E}\|\boldsymbol{H}_J - \boldsymbol{H}'_J\|^2}{|J|}\right]^{1/2}, \tag{3}$$

where $\boldsymbol{H} = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X})$ and $\boldsymbol{H}' = \mathcal{E}(\boldsymbol{A}, \boldsymbol{X}_{J^c})$ denote the node embedding of the given graph and the masked graph, respectively, and $\boldsymbol{H}_J := \boldsymbol{H}[J, :]$ selects rows with indices in J.

Xie *et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

BGRL/BYOL

- Both include an invariance term to regularize the difference between representations of two versions of a graph.
- BGRL draws intuition from contrastive learning, requires engineering and tricks to guarantee non-trivial solutions. Not much theoretical justification.



Grill et al., Bootstrap your own latent: A new approach to self-supervised Learning. NeurIPS 2020.

Results: Node-level Tasks

Table 2. Performance on node-level datasets, 20 runs averaged. Results of SSL methods with the best performance are highlighted in bold numbers. *Left*: Mean classification accuracy on transductive datasets, with baseline results from Thakoor et al. (2021). *Right*: Micro-averaged F1 scores on larger-scale inductive datasets, with baseline results from Thakoor et al. (2021) and Jiao et al. (2020).

Transductive	Am.Comp.	Am.Pht.	Co.CS	Co.Phy
Raw features	73.8±0.0	78.5±0.0	90.4±0.0	93.6±0.0
DeepWalk	85.7±0.1	89.4±0.1	84.6±0.2	91.8±0.2
GAE	87.7±0.3	92.7±0.3	92.4±0.2	95.3±0.1
VGAE	88.1±0.3	92.8±0.3	92.5±0.2	95.3±0.1
Supervised	86.5±0.5	92.4±0.2	93.0±0.3	95.7±0.2
DGI	84.0±0.5	91.6±0.2	92.2±0.6	94.5±0.5
GMI	82.2±0.3	90.7±0.2	OOM	OOM
MVGRL	87.5±0.1	91.7±0.1	92.1±0.1	95.3±0.0
GRACE	87.5±0.2	92.2±0.2	92.9±0.0	95.3±0.0
GCA	88.9±0.2	92.5±0.2	93.1±0.0	95.7±0.0
BGRL	**89.7±0.3**	92.9±0.3	93.2±0.2	95.6±0.1
LaGraph	88.0±0.3	**93.5±0.4**	**93.3±0.2**	**95.8±0.1**

Inductive	PPI	Flickr	Reddit
Raw feat.	42.5±0.3	20.3±0.2	58.5±0.1
GAE	75.7±0.0	50.7±0.2	OOM
VGAE	75.8±0.0	50.4±0.2	OOM
Super-GCN	51.5±0.6	48.7±0.3	93.3±0.1
Super-GAT	97.3±0.2	OOM	OOM
GraphSAGE	46.5±0.7	36.5±1.0	90.8±1.1
DGI	63.8±0.2	42.9±0.1	94.0±0.1
GMI	65.0±0.0	44.5±0.2	95.0±0.0
SUBG-CON	66.9±0.2	48.8±0.1	**95.2±0.0**
BGRL-GCN	69.6±0.2	50.0±0.3*	OOM*
BGRL-GAT	70.5±0.1	44.2±0.1*	OOM*
LaGraph	**74.6±0.0**	**51.3±0.1**	**95.2±0.0**

***Xie** et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

Results: Graph-level Tasks

Table 1. Performance on graph-level classification tasks, scores are averaged over 5 runs. Bold and underlined numbers highlight the top-2 performance. OOM indicates running out-of-memory on a 56GB Nvidia A6000 GPU.

	NCI1	PROTEINS	DD	MUTAG	COLLAB	RDT-B	RDT-M5K	IMDB-B
GL	–	–	–	81.7±2.1	–	77.3±0.2	41.0±0.2	65.9±1.0
WL	80.0±0.5	72.9±0.6	–	80.7±3.0	–	68.8±0.4	46.1±0.2	72.3±3.4
DGK	80.3±0.5	73.3±0.8	–	87.4±2.7	–	78.0±0.4	41.3±0.2	67.0±0.6
Node2Vec	54.9±1.6	57.5±3.6	75.1±0.5	72.6±10.2	55.7±0.2	73.8±0.5	34.1±0.4	50.0±0.8
Sub2Vec	52.8±1.5	53.0±5.6	73.6±1.5	61.1±15.8	62.1±1.4	71.5±0.4	36.7±0.4	55.3±1.5
Graph2Vec	73.2±1.8	73.3±2.1	76.2±0.1	83.2±9.3	59.9±0.0	75.8±1.0	47.9±0.3	71.1±0.5
GAE	73.3±0.6	74.1±0.5	77.9±0.5	84.0±0.6	56.3±0.1	74.8±0.2	37.6±1.6	52.1±0.2
VGAE	73.7±0.3	74.0±0.5	77.6±0.4	84.4±0.6	56.3±0.0	74.8±0.2	39.1±1.6	52.1±0.2
InfoGraph	76.2±1.1	<u>74.4±0.3</u>	72.9±1.8	89.0±1.1	70.7±1.1	82.5±1.4	53.5±1.0	73.0±0.9
GraphCL	<u>77.9±0.4</u>	74.4±0.5	**78.6±0.4**	86.8±1.3	71.4±1.2	<u>89.5±0.8</u>	<u>56.0±0.3</u>	71.1±0.4
MVGRL	75.1±0.5	71.5±0.3	OOM	<u>89.7±1.1</u>	OOM	84.5±0.6	OOM	**74.2±0.7**
LaGraph	**79.9±0.5**	**75.2±0.4**	<u>78.1±0.4</u>	**90.2±1.1**	**77.6±0.2**	**90.4±0.8**	**56.4±0.4**	<u>73.7±0.9</u>

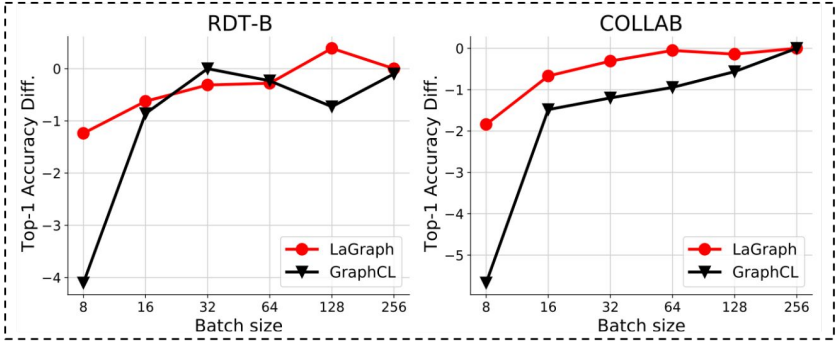
***Xie** et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

Scalability and Robustness

	# nodes sampled	100	1,000	2,500	5,000	10,000	all
Flickr	% nodes sampled	0.22%	2.24%	5.60%	11.20%	22.41%	100.00%
	F1-score - *LaGraph*	6.07	51.12	51.12	51.27	51.29	51.26
	Memory - *LaGraph*	1389MB	1465MB	1553MB	1725MB	2065MB	4211MB
	F1-score - GraphCL	45.27	45.27	45.27	45.38	45.45	45.48
	Memory - GraphCL	1647MB	2599MB	4137MB	6741MB	11905MB	47939MB
Reddit	% nodes sampled	0.07%	0.65%	1.63%	3.25%	6.50%	100.00%
	F1-score - *LaGraph*	5.76	95.05	95.06	95.08	95.09	95.22
	Memory - *LaGraph*	1403MB	1475MB	1585MB	1783MB	2161MB	16933MB
	F1-score - GraphCL	93.24	93.24	93.25	93.31	93.32	OOM
	Memory - GraphCL	4199MB	6117MB	6687MB	9297MB	14495MB	OOM

⬆ *Model scalability on large-scale graphs: less memory usage during training.*

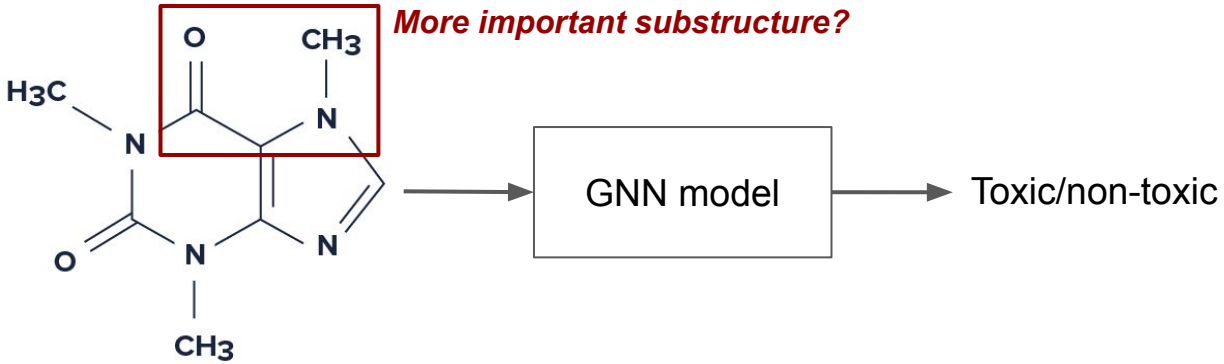
Model robustness to small mini-batch sizes: ➡ *less performance losses when memory is limited.*



***Xie** et al., Self-Supervised Representation Learning via Latent Graph Prediction. ICML'22.*

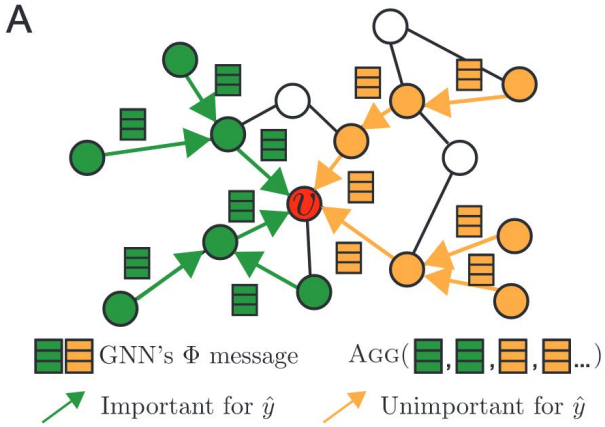
Graph Explanations

- Given a well-trained GNN and a graph instance, figure out the substructure that lead (contribute the most) to the prediction results.



Graph Explanations

- Perturbation-based explanation (e.g., GNNExplainer, PGExplainer).
- **Key idea**: to learn masks on given graph edges (and node attributes), s.t., when unimportant edges/attributes are masked-out, the prediction results of the GNN should be similar.
- Requires the entire GNN model to learn an explainer.



Ying et al. GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS 2019.

Graph Explanation in 2-Stage Training Setting

- The 2-stage training is common in industrial applications.
- Issues: undefined downstream task and unavailable downstream models during stage 1, current explainers:
  - Unable to explain the GNN encoder without downstream tasks.
  - Unable to explain GNN encoder with non-differentiable downstream models.
  - Have to repeatedly train explainers when downstream models are updated.

***Xie** et al., Task-agnostic Graph Explanations. NeurIPS'22.*

Graph Explanation in Multi-task Setting

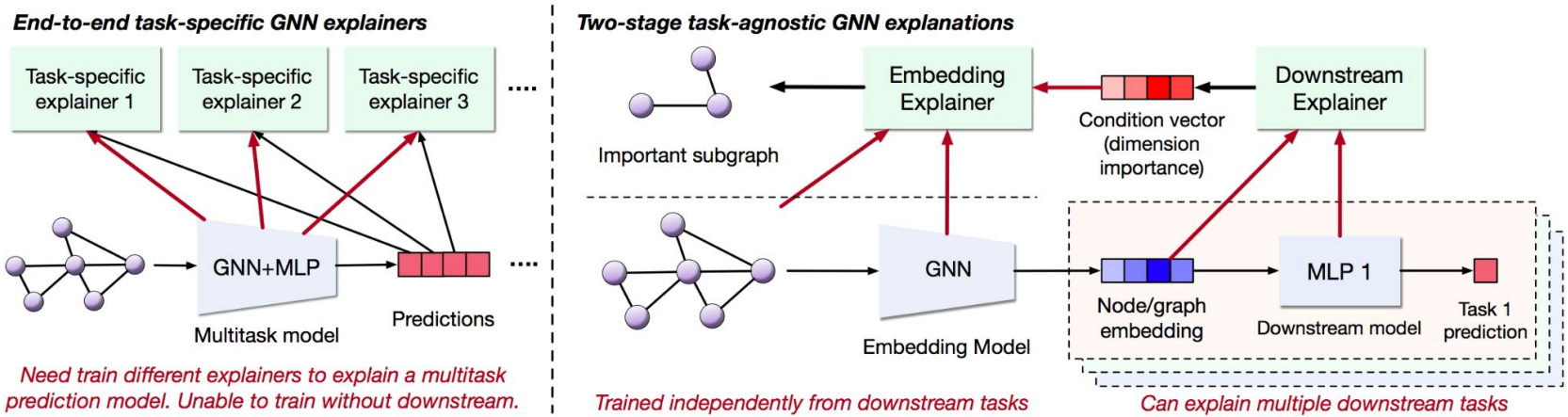
- Multi-task prediction is common for graph machine learning, e.g., to predict multiple chemical properties in drug discovery for a molecular graph.
  - ToxCast from MoleculeNet has 167 graph-level prediction tasks.
  - PPI has 121 node-level binary classification tasks.

	Learning	Inductive	Task-agnostic	# explainers required
Gradient- & Rule-based	No	-	-	1
GNNExplainer (Ying et al., 2019)	Yes	No	No	$M * N$
SubgraphX (Yuan et al., 2021)	Yes	No	No	$M * N$
PGExplainer (Luo et al., 2020)	Yes	Yes	No	M
Task-agnostic explainers	Yes	Yes	Yes	1

***Xie** et al., Task-agnostic Graph Explanations. NeurIPS'22.*

The Task-Agnostic Graph Explanations

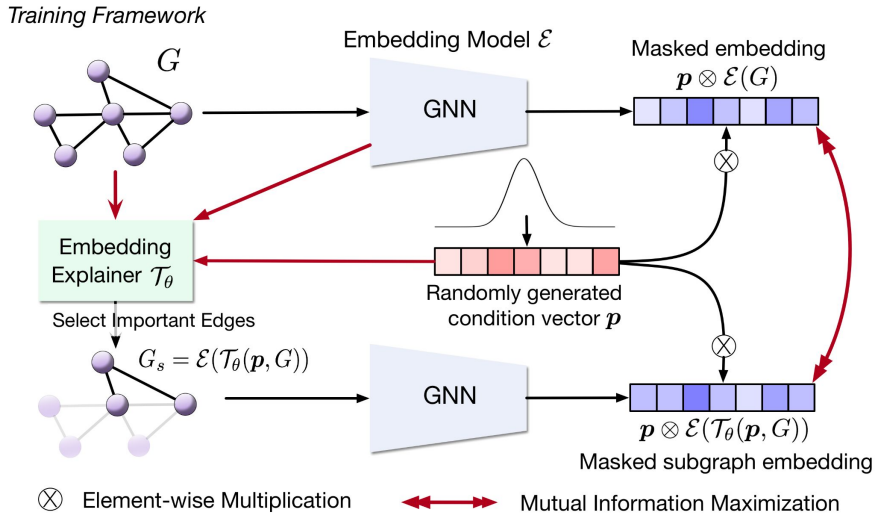
- To construct embedding explainer and downstream explainer respectively.



***Xie** et al., Task-agnostic Graph Explanations. NeurIPS'22.*

The Embedding Explainer & Training

- Inputs: the graph data, the GNN encoder, a condition vector.
- Outpus: the importance score for each edge.



Objective:

$$\max_\theta \mathrm{E}_{\boldsymbol{p}}[\mathbf{MI}(\boldsymbol{p} \otimes \mathcal{E}(G), \boldsymbol{p} \otimes \mathcal{E}(\mathcal{T}_\theta(\boldsymbol{p}, G)))]$$

Practically:

$$\min_\theta -\frac{1}{N}\sum_{i=1}^{N}\left[\log \frac{\exp\{(\boldsymbol{p} \otimes \boldsymbol{z}_i)^T(\boldsymbol{p} \otimes \boldsymbol{z}_{i,\theta})\}}{\sum_{j \neq i}\exp\{(\boldsymbol{p} \otimes \boldsymbol{z}_i)^T(\boldsymbol{p} \otimes \boldsymbol{z}_{j,\theta})\}}\right]$$

***Xie** et al., Task-agnostic Graph Explanations. NeurIPS'22.*

The Downstream Explainer

- A simple gradient-based explainer, since the downstream models are usually based on fully-connected layers and are less challenging to explain.

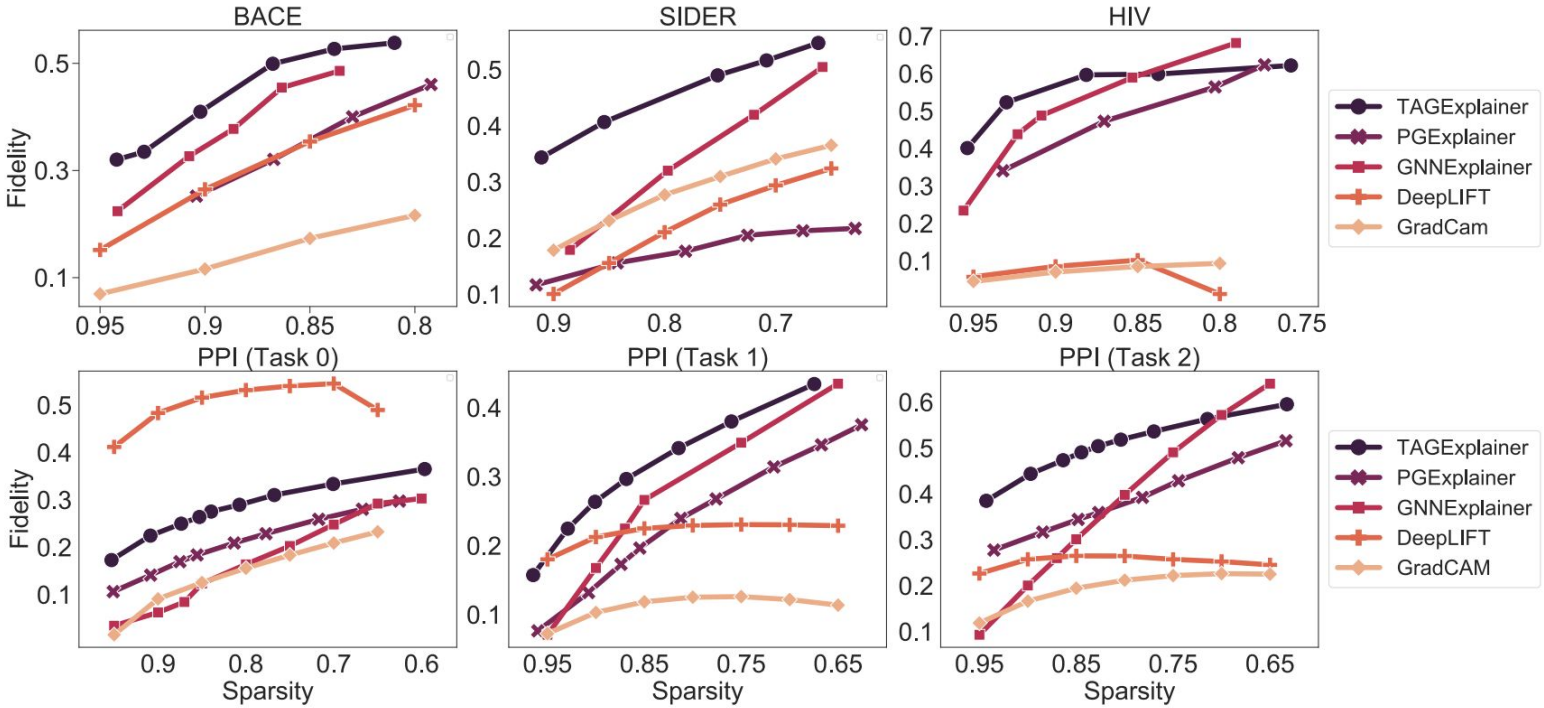
$$\boldsymbol{g} = \frac{\partial \max_{c \leq C} \mathcal{D}(\boldsymbol{z})[c]}{\partial \boldsymbol{z}} \in \mathbb{R}^{1 \times d}$$

$$\boldsymbol{p} = \mathrm{ReLU}(\mathrm{norm}(\boldsymbol{g}^T))$$

- Value in a dimension with higher gradient indicates that the dimension is more important for a certain downstream task

***Xie** et al., Task-agnostic Graph Explanations. NeurIPS'22.*

Experimental Studies



***Xie** et al., Task-agnostic Graph Explanations. NeurIPS'22.*

Experimental Studies

Table 6. Fidelity scores with controlled sparsity on the E-commerce product dataset. Each column corresponds to one explainer model trained on different tasks or without downstream task. Underlines highlight the best explanation quality in terms of fidelity, on the same level of sparsity.

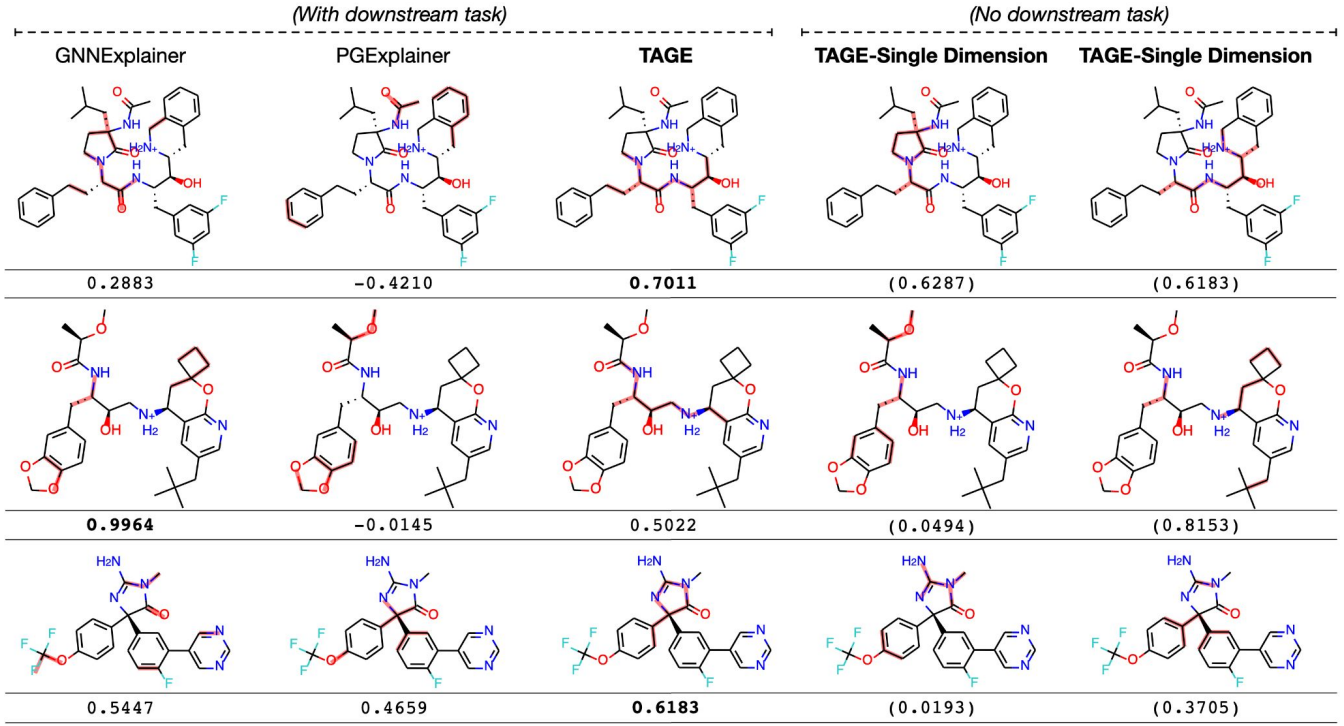
	PGExplainer (trained on)			TAGE
Eval on	Buyers	Sellers	Reviews	w/o downstream
Buyers	**0.2009 ±0.2233**	0.1731 ±0.3774	0.1740 ±0.4463	**0.2713 ±0.1834**
Sellers	0.5465 ±0.4773	**0.3246 ±0.4026**	0.1128 ±0.3019	**0.6515 ±0.3426**
Reviews	0.4178 ±0.3683	0.1258 ±0.3492	**0.2310 ±0.4178**	**0.5692 ±0.4214**

Table 3. Comparison of computational time cost among three learning-based GNN explainers on the PPI dataset. The left two columns record time cost breakdown for T downstream tasks. The fourth column estimates the total time cost for explaining all 121 tasks of PPI. The last row shows the speedup times compared to GNNExplainer and PGExplainer, respectively.

Time cost	Training (s)	Inference (s)	Total time (T=1) (s)	Est. total for 121 tasks
GNNExplainer	20040.1*T	–	20040.1	28 d
PGExplainer	7117.0*T	**427.2*T**	7604.2	10.7 d
TAGE	**1405.3**	582.7*T	**1988.0**	**0.83 d**
Speedup	**14.3*T× / 5.1*T×**	– / 0.73×	**10.1× / 3.8×**	**33.7× / 12.9×**

***Xie** et al., Task-agnostic Graph Explanations. NeurIPS'22.*

Visualizations: Drug discovery



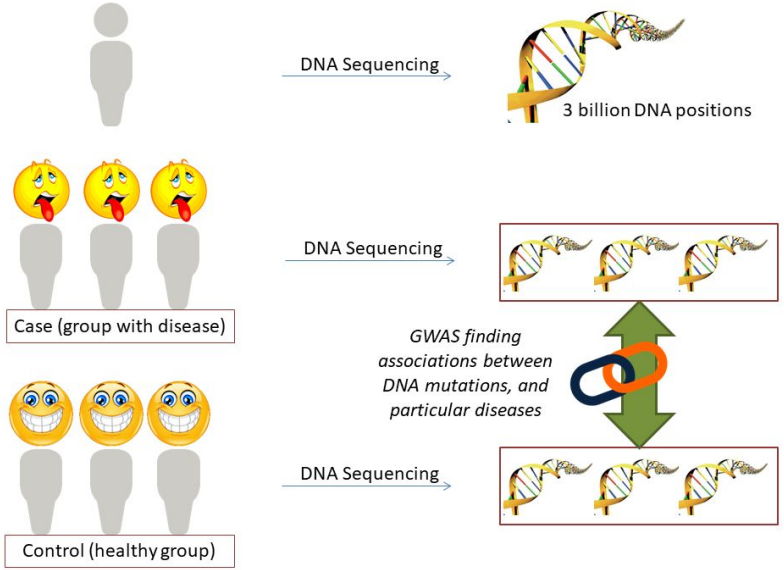
***Xie** et al., Task-agnostic Graph Explanations. NeurIPS'22.*

TEXAS A&M UNIVERSITY
Engineering

Future Directions of SSL

Future Directions of SSL

- Self-supervised learning to solve genome-wide association study (GWAS) problems
  - Difficulties on high-dimensional data.
  - Requires encoders and explainers both trained under self-supervision.



Source: https://www.esat.kuleuven.be/cosic/blog/privacy-preserving-gwas-practical/

Future Directions of SSL

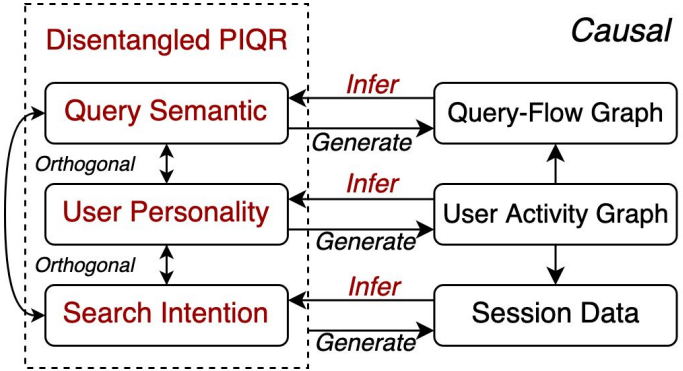
- Self-supervised learning to solve genome-wide association study (GWAS) problems

		# Independent Loci			
		All	Unique	MI (JSE)	Heritability h^2
	Random Init	14	1	1.2165	0.0756 ± 0.0656
Predictive	Autoencoder (Patel et al., 2022)	26	1	1.3120	0.3121 ± 0.0769
	Autoencoder-attention	23	4	1.3124	0.2984 ± 0.0773
	GenPred	10	0	1.2412	0.0918 ± 0.1110
Contrastive	Barlow Twins (Zbontar et al., 2021)	11	1	1.2996	0.0814 ± 0.0636
	SimCLR (Chen et al., 2020)	15	1	1.2397	0.1448 ± 0.1128
	SimCLR-JSE	17	7	1.3044	0.1604 ± 0.1151
Trans-Modal Contrastive	InfoNCE (ContIG, Taleb et al. (2022))	11	0	1.2299	0.1334 ± 0.0588
	Decorrelated InfoNCE	13	3	1.2382	0.0527 ± 0.0349
	GIM (Ours)	**40**	**15**	**1.3681**	**0.3723 ± 0.0305**

- Next steps: self-supervised explainability on associations, causal relationships between gene and phenotypes, etc.

Future Directions of SSL

- Identify and use causal effects when performing self-supervised representation learning
  - Causal effects are among latent variables, who do not have labels.
  - Better generalization and adaptation.



An example with industrial searching data.

Future Directions of SSL

- Broader scientific discovery (AI4Science) problems
  - Material design, PDE, Quantum mechanics, etc.
- How to incorporate [prior/domain knowledge, physical constraints, etc.] into SSL?
- What self-supervision can we use given different forms of data for scientific discovery?

List of Publications

Self-supervised learning

- **Yaochen Xie**, Sumeet Katariya, Xianfeng Tang, Edward Huang, Nikhil Rao, Karthik Subbian, Shuiwang Ji, “Task-Agnostic Graph Explanations.” The 36th Annual Conference on Neural Information Processing Systems (**NeurIPS**), 2022
- **Yaochen Xie***, Zhao Xu*, and Shuiwang Ji, “Self-Supervised Representation Learning via Latent Graph Prediction.” International Conference on Machine Learning (**ICML**), 24460-24477, 2022.
- **Yaochen Xie**, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji, “Self-supervised Learning of Graph Neural Networks: A Unified Review.” IEEE Transactions on Pattern Analysis and Machine Intelligence (**TPAMI**), 2022.
- Xinyi Xu, Cheng Deng, **Yaochen Xie**, Shuiwang Ji, “Group Contrastive Self-Supervised Learning on Graphs.” IEEE Transactions on Pattern Analysis and Machine Intelligence (**TPAMI**), 2022.
- **Yaochen Xie**, Zhengyang Wang, and Shuiwang Ji, “Noise2Same: Optimizing A Self-Supervised Bound for Image Denoising.” The 34th Neural Information Processing Systems (**NeurIPS**), 20320-20330, 2020.

List of Publications

Self-supervised learning (cont'd)

- Liu, Meng*, Youzhi Luo*, Limei Wang*, **Yaochen Xie***, Hao Yuan*, Shurui Gui, Haiyang Yu et al. "DIG: A Turnkey Library for Diving into Graph Deep Learning Research." Journal of Machine Learning Research (**JMLR**) 22 (2021): 1-9.

Self-attention mechanism for augmented microscopy

- **Yaochen Xie**, Yu Ding, Shuiwang Ji, "Augmented Equivariant Attention Networks for Microscopy Image Transformation." IEEE Transactions on Medical Imaging (**TMI**), 2022.
- Zhengyang Wang*, **Yaochen Xie***, and Shuiwang Ji, "Global Voxel Transformer Networks for Augmented Microscopy." **Nature Machine Intelligence**, 3: 161-171, 2021.

List of Publications

Drug discovery, quantum chemistry, etc.

- Zhengyang Wang*, Meng Liu*, Youzhi Luo*, Zhao Xu*, **Yaochen Xie***, Limei Wang*, Lei Cai*, Qi Qi, Zhuoning Yuan, Tianbao Yang, Shuiwang Ji, “Advanced graph and sequence neural networks for molecular property prediction and drug discovery” **Bioinformatics** , 38 (9), 2579-2586, 2022.
  - First place in the MIT AI Cures Challenge for COVID-19 drug discovery.
- Liu, Meng*, Cong Fu*, Xuan Zhang, Limei Wang, **Yaochen Xie**, Hao Yuan, Youzhi Luo, Zhao Xu, Shenglong Xu, and Shuiwang Ji. "Fast Quantum Property Prediction via Deeper 2D and 3D Graph Networks." AI for Science: Mind the Gaps workshop at **NeurIPS** 2021.
  - Runner-up award in the KDD-Cup 2021. First place among teams from academia.

TEXAS A&M UNIVERSITY

Engineering

Thank you!