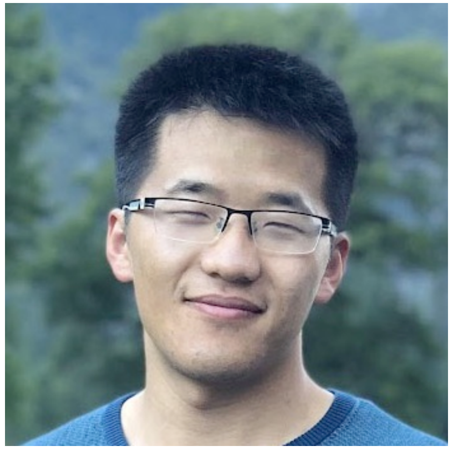
Distribution Shift Problems in Scientific Domains

Presented by Shikun Liu
Georgia Institute of Technology
Department of Electrical and Computer Engineering

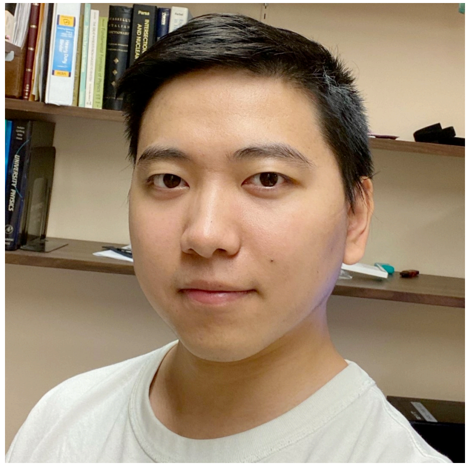
Accelerated AI Algorithms for Data-Driven Discovery

Introduction

- **PI:** Pan Li



- **A3D3 Trainees:**



Siqi Miao

Shikun Liu

GCoM

Georgia Institute of Technology
Department of Electrical and Computer Engineering

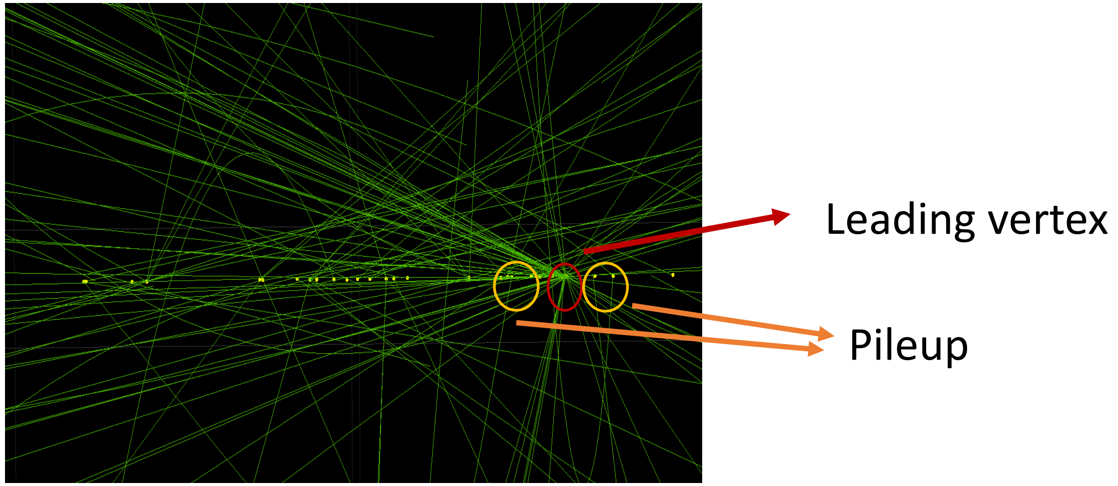
Focus: Interpretable and generalizable graph machine learning for scientific applications

Content

- Problems: Various distribution shifts in scientific applications
- A detailed example: Pileup Mitigation
- A principled solution: StruRW algorithm
- Future Works

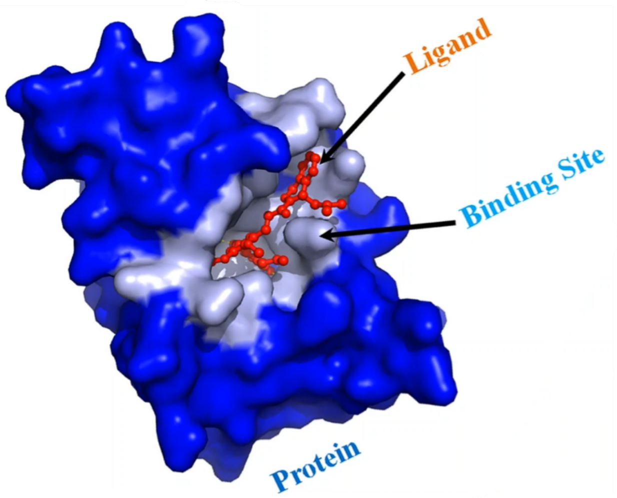
GNN for Science Applications

- Pileup Mitigation in HEP



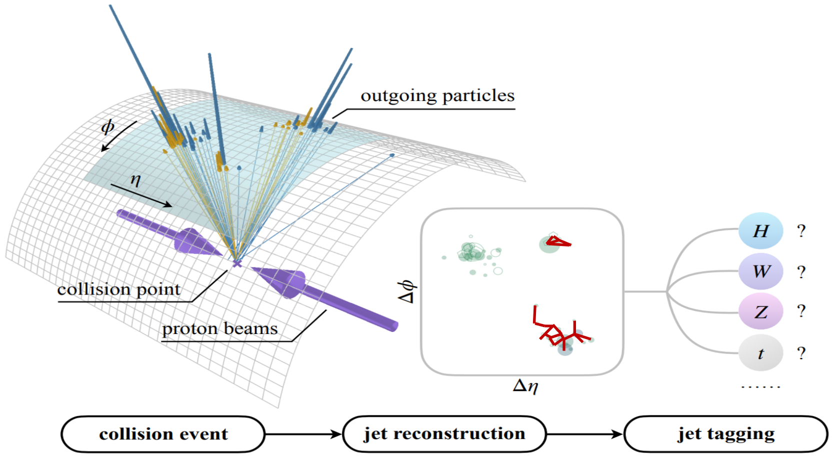
[Li et al., EPJC, 2023]

- Protein binding affinity prediction



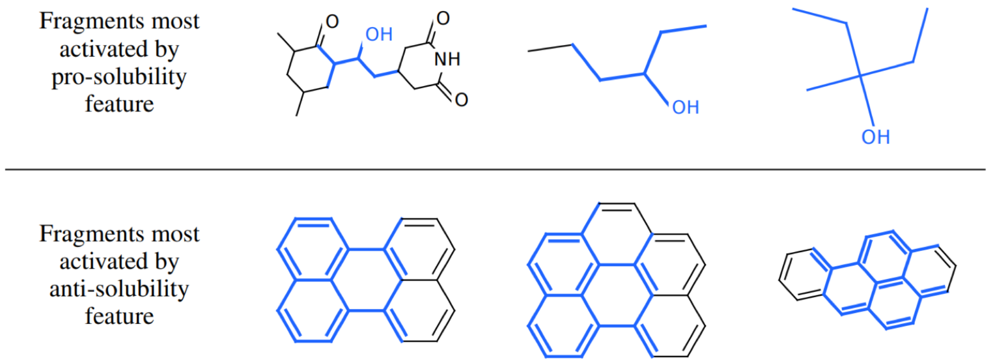
[Karimi et al., 2019]

- Jet Tagging in HEP



[Duvenaud et al., NeurIPS 2015]

- Molecular Property Prediction



Refined based on [Qu, Li, Qian, 2022]

Distribution Shift Problems

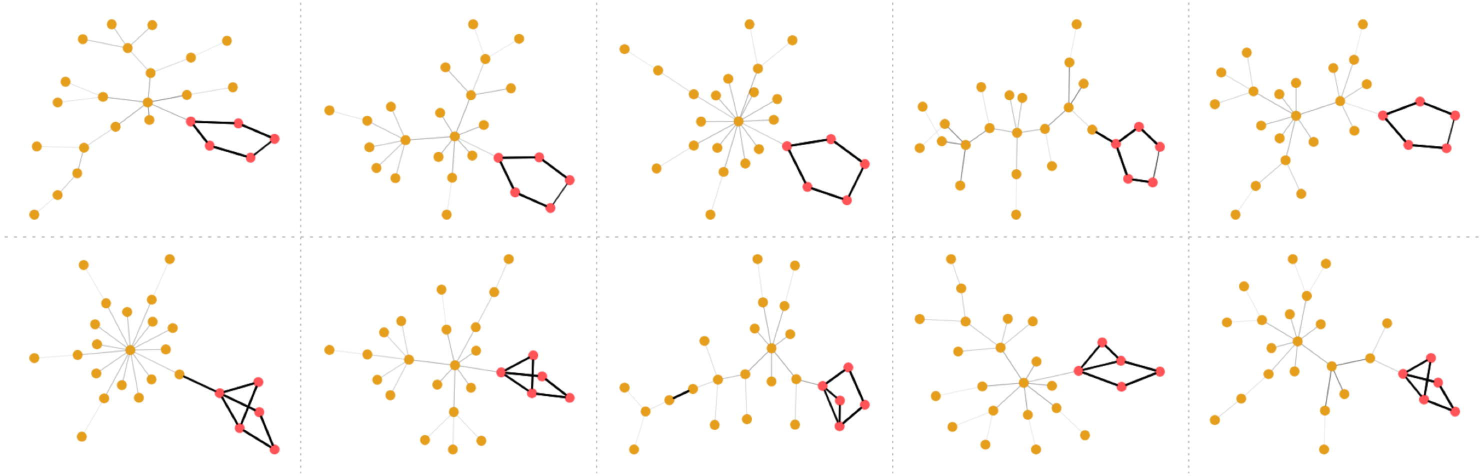
- **What is distribution shift ?**
  - Training data and testing data distribution are different

Trained model can not generalize to the testing phase

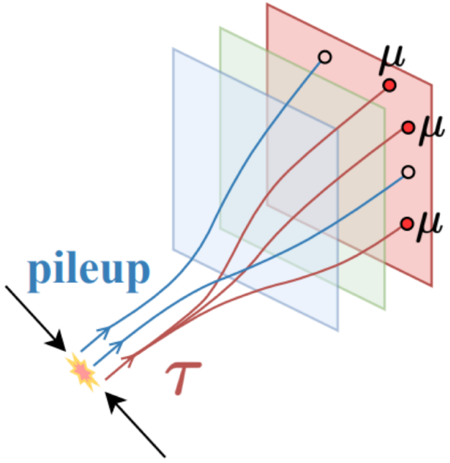
- **Exist widely in many scientific applications**
  - Synthetic data training vs. real data testing
  - Data obtained varies in different conditions
    - Time period
    - Experimental settings (location, environment, noise, …)
    - Measurement standards
  - Require task-specific generalization
  - ……

Distribution Shift Problems

- Exist widely in many scientific applications
  - Synthetic data training vs. real data testing
  - Data obtained varies in different conditions
  - Require task-specific generalization



Molecular property prediction



Detecting signal in HEP

Content

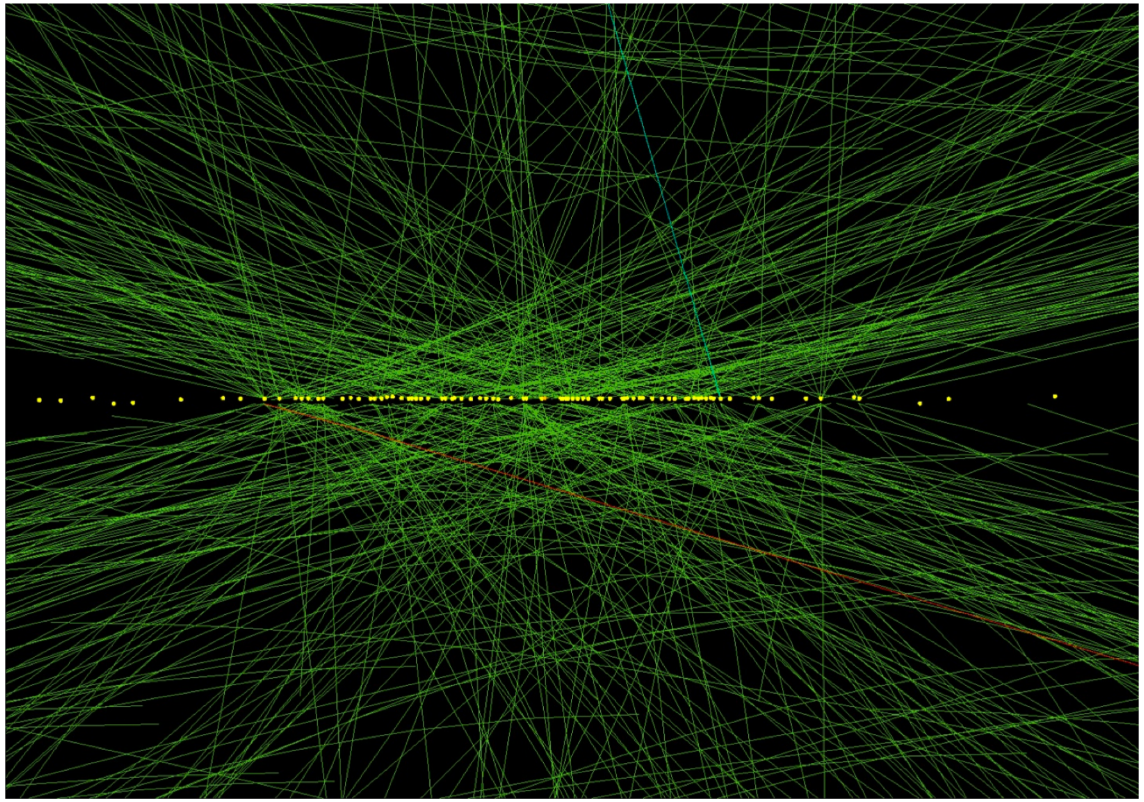
- Problems: Various distribution shifts in scientific applications
- A detailed example: Pileup Mitigation
- A principled solution: StruRW algorithm
- Future Works

Semi-supervised graph neural networks for pileup noise removal

Tianchun Li*, Shikun Liu*, Yongbin Feng*, Garyfallia Paspalaki, Nhan V. Tran, Miaoyuan Liu, Pan Li

Li, T., Liu, S., Feng, Y. *et al.* Semi-supervised graph neural networks for pileup noise removal. *Eur. Phys. J. C* **83**, 99 (2023).
https://doi.org/10.1140/epjc/s10052-022-11083-5

Example: Pileup Mitigation



- **Leading Vertex (LV):** Signal of interest from primary interactions
- **Pileup (PU):** Additional proton-proton interactions in the same or nearby bunch crossings
- **Task:** Identify whether a particle is from the LV or PU
- **Challenge:** Easy to retrieve labels for Charged particles; No truth information for Neutral particles

Example: Pileup Mitigation

How we handle this challenge?

A: Let the model train on Charged particles with given labels, then infer on Neutral particles

Distribution Shift Occurs

- **Intuition:** Make training charged particles look like testing neutral particles

- **Approach:** Masking strategy

✓ Assume the shared features have similar distribution

✓ Mask the unshared features

Example: Pileup Mitigation

How we handle this challenge?

A: Let the model train on Charged particles with given labels, then infer on Neutral particles

Distribution Shift Occurs

- **Intuition:** Make training charged particles look like testing neutral particles

- **Approach:** Masking strategy

✓ Assume the shared features have similar distribution

✓ Mask the unshared features

Example: Pileup Mitigation

How we handle this challenge?

A: Let the model train on Charged particles with given labels, then infer on Neutral particles

Distribution Shift Occurs

- **Intuition:** Make training charged particles look like testing neutral particles

- **Approach:** Masking strategy

✓ Assume the shared features have similar distribution

✓ Mask the unshared features

Example: Pileup Mitigation

How we handle this challenge?

A: Let the model train on Charged particles with given labels, then infer on Neutral particles

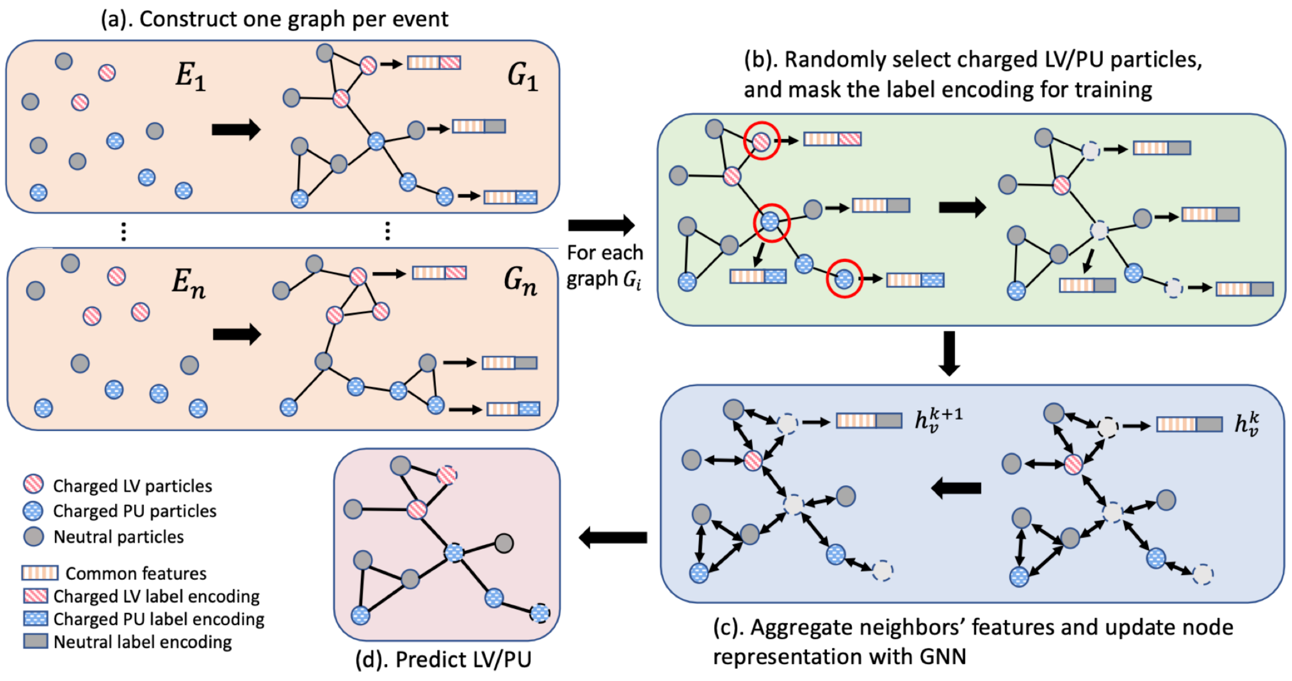
Distribution Shift Occurs

- **Intuition:** Make training charged particles look like testing neutral particles

- **Approach:** Masking strategy

✓ Assume the shared features have similar distribution

✓ Mask the unshared features



However…

- The shared features may exhibit different distributions
- There could be additional graph structure shift
- Additional generalization cases are needed
  - Shift across synthetic and real datasets
  - Shift across different pileup level
  - Shift over particles within different locations of detector

More principled and general methodology is needed

However…

- The shared features may exhibit different distributions
- There could be additional graph structure shift
- Additional generalization cases are needed
  - Shift across synthetic and real datasets
  - Shift across different pileup level
  - Shift over particles within different locations of detector

More principled and general methodology is needed

Content

- Problems: Various distribution shifts in scientific applications
- A detailed example: Pileup Mitigation
- A principled solution: StruRW algorithm
- Future Works

Structural Re-weighting Improves Graph Domain Adaptation

Shikun Liu, Tianchun Li, Yongbin Feng, Nhan Tran, Han Zhao, Qiu Qiang, Pan Li

Liu, Shikun, et al. "Structural Re-weighting Improves Graph Domain Adaptation." *International Conference on Machine Learning*. PMLR, 2023.

Problem Formulation

- **Categories:** Out of distribution generalization (OOD) and domain adaptation (DA)
- **Difference:**
  - OOD: no access to target / testing data
  - DA: have access to target / testing data
- **Similar goal:** Want the model to generalize well on target data
- **Focus on:** Unsupervised domain adaptation (have access to target feature but no label information)

$$P_S(X,Y) \neq P_T(X,Y)$$

Problem Formulation

- **Categories:** Out of distribution generalization (OOD) and domain adaptation (DA)
- **Difference:**
  - OOD: no access to target / testing data
  - DA: have access to target / testing data
- **Similar goal:** Want the model to generalize well on target data
- **Focus on:** Unsupervised domain adaptation (have access to target feature but no label information)

$$\boxed{\boldsymbol{P_S(X,Y) \neq P_T(X,Y)}}$$

Problem Formulation

$$\boxed{\boldsymbol{P_S(X,Y) \neq P_T(X,Y)}}$$

- **Assumption:**
  - Covariate Shift: $P_S\,(X) \neq P_T(X)$ and $P_S\,(Y|X) = P_T(Y|X)$
  - Label Shift: $P_S\,(Y) \neq P_T(Y)$ and $P_S\,(X|Y) = P_T(X|Y)$
  - Conditional Shift: $P_S\,(X|Y) \neq P_T\,(X|Y)$
- **Common Methodology:** Invariant representation learning

$$\boxed{\boldsymbol{Z = \phi(X)\,;\, P_S(Z) = P_T(Z)}}$$

Problem Formulation

$$\boxed{\boldsymbol{P_S(X,Y) \neq P_T(X,Y)}}$$

- **Assumption:**
  - Covariate Shift: $P_S\,(X) \neq P_T(X)$ and $P_S\,(Y|X) = P_T\,(Y|X)$
  - Label Shift: $P_S\,(Y) \neq P_T(Y)$ and $P_S\,(X|Y) = P_T\,(X|Y)$
  - Conditional Shift: $P_S\,(X|Y) \neq P_T\,(X|Y)$
- **Common Methodology:** Invariant representation learning

$$\boxed{\boldsymbol{Z = \phi(X)\,;\ P_S(Z) = P_T(Z)}}$$

Principled solution: StruRW – Problem

$$\boxed{Z = \phi(X)\,;\; P_S(Z) = P_T(Z)}$$

- **When extending this idea to graph structured data (GDA) …**
- **Assumption:** $P_S\,(Y) = P_T(Y)$

$$\boxed{Z = \phi(A, X)\,;\; P_S(Z) = P_T(Z)}$$

A: Adjacency matrix; X: Node features

Suboptimality of invariant representation learning in GDA

Principled solution: StruRW – Problem

$$\boxed{Z = \phi(X)\,;\ P_S(Z) = P_T(Z)}$$

- **When extending this idea to graph structured data (GDA) …**
- **Assumption:** $P_S\,(Y) = P_T(Y)$

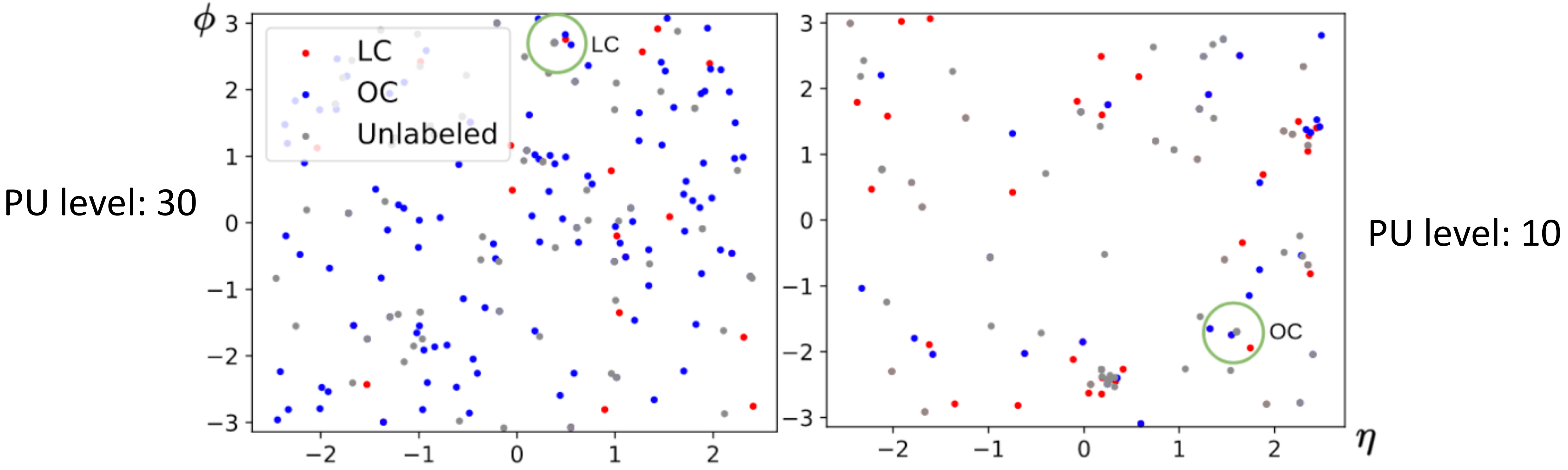
$$\boxed{Z = \phi(A, X)\,;\ P_S(Z) = P_T(Z)}$$

A: Adjacency matrix; X: Node features

Suboptimality of invariant representation learning in GDA

Principled solution: StruRW – Problem

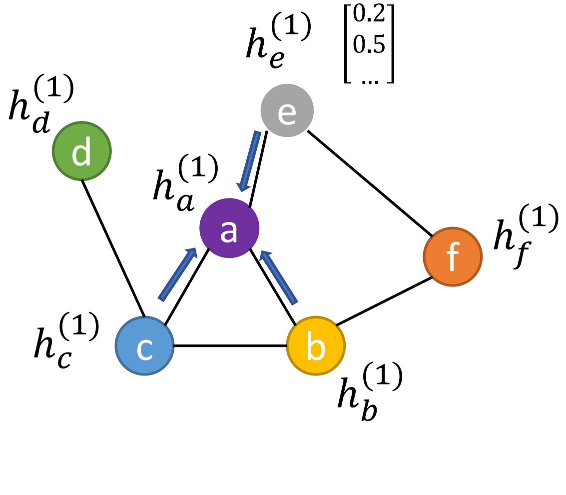
Conditional Structure Shift (CSS)

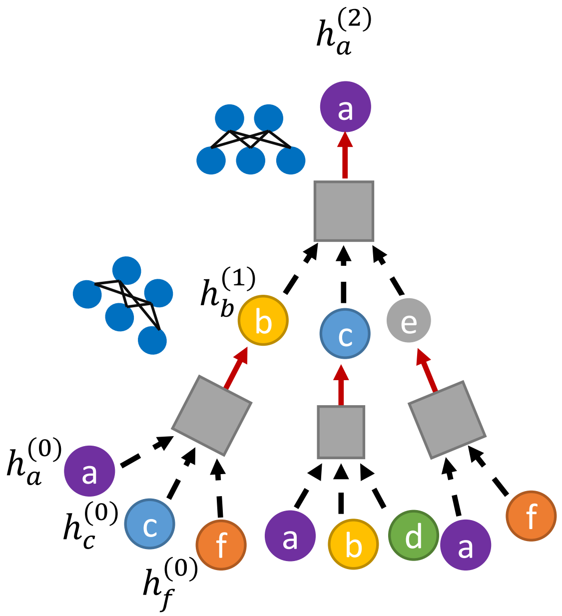


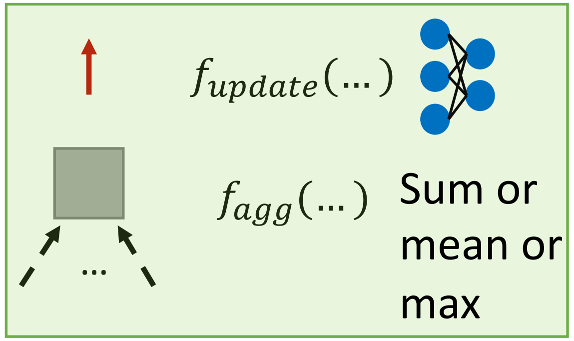
$$P_S(A|Y) \neq P_T(A|Y)$$

Principled solution: StruRW – Methodology

Graph Neural Network (GNN)







$$h_v^{(t+1)} = f_{update}\left(h_v^{(t)}, f_{agg}\left(\left\{h_u^{(t)} \mid u \in N_v\right\}\right)\right)$$

$$h_G = \mathrm{POOL}\left(\left\{h_v^{(L)} \mid v \in V\right\}\right)$$

Principled solution: StruRW – Methodology

Consider the one layer Message Passing as aggregating neighborhood representations to form a multiset

Source	Different cardinality and distribution of elements in multisets	Target
$\{h_0, h_1, h_0, h_1, h_1, \ldots\}$		$\{h_0, h_0, h_0, h_0, h_1, \ldots\}$
$\{h_0, h_1, h_1, h_1, h_1, \ldots\}$		$\{h_1, h_1, h_0, h_0, h_0, \ldots\}$
$\vdots$		$\vdots$

Goal: Downsample / resample the elements in multiset to let the source multiset distribution approximate target multiset distribution

Principled solution: StruRW – Methodology

Consider the one layer Message Passing as aggregating neighborhood representations to form a multiset

Source		Target
$\{h_0, h_1, h_0, h_1, h_1, \ldots\}$	Different cardinality and distribution of elements in multisets	$\{h_0, h_0, h_0, h_0, h_1, \ldots\}$
$\{h_0, h_1, h_1, h_1, h_1, \ldots\}$		$\{h_1, h_1, h_0, h_0, h_0, \ldots\}$
$\vdots$		$\vdots$

Goal: Downsample / resample the elements in multiset to let the source multiset distribution approximate target multiset distribution

Principled solution: StruRW – Methodology

Consider the one layer Message Passing as aggregating neighborhood representations to form a multiset

Source		Target
$\{h_0, h_1, h_0, h_1, h_1, \ldots\}$	Different cardinality and distribution of elements in multisets	$\{h_0, h_0, h_0, h_0, h_1, \ldots\}$
$\{h_0, h_1, h_1, h_1, h_1, \ldots\}$		$\{h_1, h_1, h_0, h_0, h_0, \ldots\}$
$\vdots$		$\vdots$

Goal: Downsample / resample the elements in multiset to let the source multiset distribution approximate target multiset distribution

Methodology: StruRW

Goal: Downsample / resample the elements in multiset to let the source multiset distribution approximate target multiset distribution

In Practice …

- Graph structure describes as the edge connection probability
  - a $k \times k$ edge connection probability matrix **B** with k classes
  - For a class-i node, nB_{ij} many class-j attributes for $j \in [k]$ in the multiset
- GNN pooling layer in aggregating information in multisets

Transforms as edge weights from class-j nodes to class-i nodes with B_{ij}^T / B_{ij}^S on source graph

$$B_{ij} = \frac{|\{e_{uv} \in \mathcal{E} | y_u = i, y_v = j\}|}{|\{v \in \mathcal{V} | y_v = i\}| \times |\{v \in \mathcal{V} | y_v = j\}|}.$$

Principled solution: StruRW – Methodology

Goal: Downsample / resample the elements in multiset to let the source multiset distribution approximate target multiset distribution

In Practice …

- Estimate the edge connection probability
- GNN pooling layer in aggregating information in multisets

Transforms as edge weights from class-j nodes to class-i nodes with B_{ij}^T / B_{ij}^S on source graph

$$B_{ij} = \frac{|\{e_{uv} \in \mathcal{E} | y_u = i, y_v = j\}|}{|\{v \in \mathcal{V} | y_v = i\}| \times |\{v \in \mathcal{V} | y_v = j\}|}.$$

B_{ij}^T can be approximated with pseudo-labels

Hyperparameter: $\lambda + (1-\lambda) B_{ij}^T / B_{ij}^S$

Principled solution: StruRW – Methodology

Goal: Downsample / resample the elements in multiset to let the source multiset distribution approximate target multiset distribution

In Practice …

- Estimate the edge connection probability
- GNN pooling layer in aggregating information in multisets

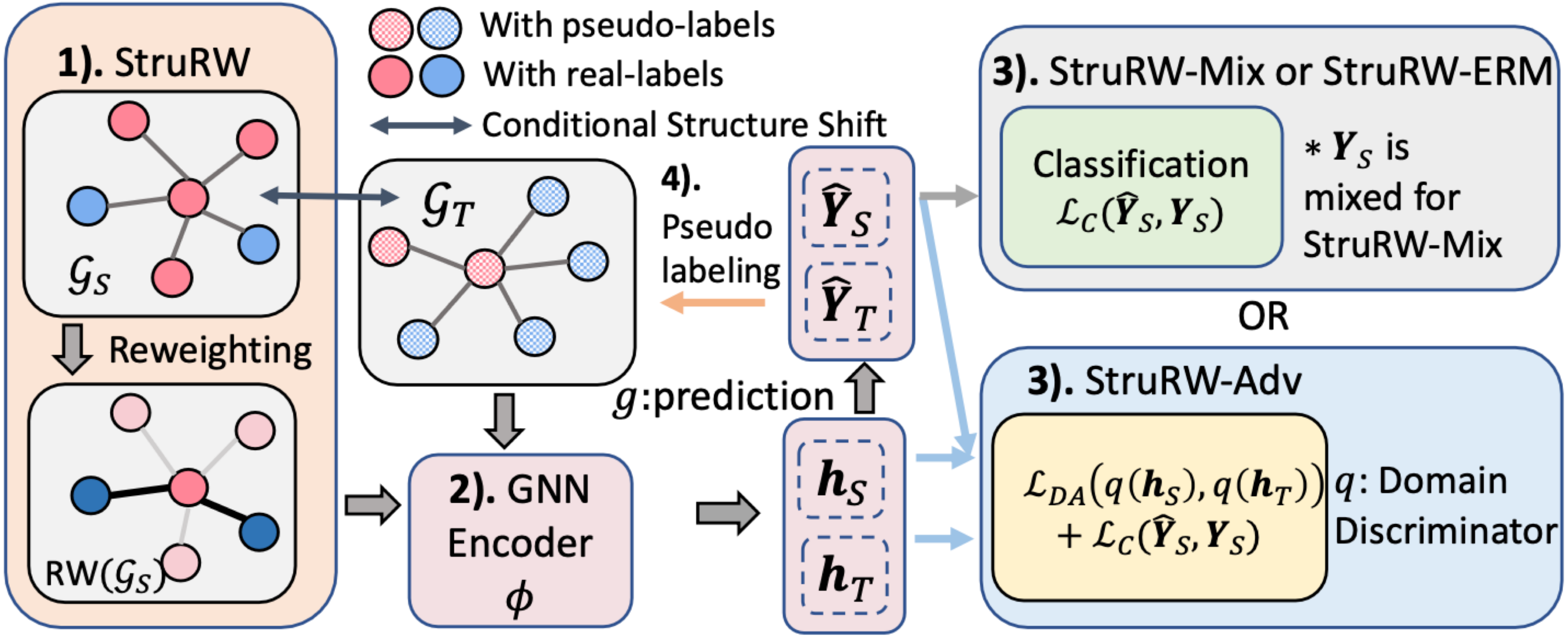
Transforms as edge weights from class-j nodes to class-i nodes with B_{ij}^{T}/B_{ij}^{S} on source graph

$$B_{ij} = \frac{|\{e_{uv} \in \mathcal{E} | y_u = i, y_v = j\}|}{|\{v \in \mathcal{V} | y_v = i\}| \times |\{v \in \mathcal{V} | y_v = j\}|}.$$

B_{ij}^{T} can be approximated with pseudo-labels

Hyperparameter: $\lambda + (1-\lambda)B_{ij}^{T}/B_{ij}^{S}$

Principled solution: StruRW – Methodology



Principled solution: StruRW – Experiments

Table 2: Synthetic CSBM results. The **bold** font and the underline indicate the first and second best model respectively, † indicates the significant improvement, where the mean-1*std of a method > the mean of its corresponding backbone model.

	$q = 0.016$	$q = 0.014$	$q = 0.012$	$q = 0.01$	$q = 0.006$	$q = 0.001$
ERM	36.52 ± 3.76	41.62 ± 5.92	48.66 ± 6.31	57.29 ± 5.28	89.72 ± 2.62	100 ± 0
DANN	64.25 ± 5.69	72.56 ± 8.54	79.63 ± 6.84	86.29 ± 8.14	96.88 ± 1.35	100 ± 0
CDAN	67.53 ± 4.98	75.38 ± 7.46	82.51 ± 6.95	89.73 ± 7.44	97.03 ± 1.09	100 ± 0
UDAGCN	51.98 ± 1.31	57.83 ± 3.05	59.74 ± 1.52	65.97 ± 1.66	98.25 ± 0.52	100 ± 0
EERM	57.36 ± 4.52	65.88 ± 3.09	70.12 ± 10.26	72.87 ± 13.70	95.01 ± 3.88	100 ± 0
MIXUP	62.54 ± 2.77	69.21 ± 2.03	74.92 ± 1.56	82.87 ± 3.45	96.89 ± 0.38	100 ± 0
STRURW-ERM	$85.24^{\dagger} \pm 1.63$	$87.92^{\dagger} \pm 1.77$	$90.26^{\dagger} \pm 1.05$	$93.84^{\dagger} \pm 0.98$	$98.28^{\dagger} \pm 0.14$	$\mathbf{100} \pm 0$
STRURW-ADV	$\underline{86.37}^{\dagger} \pm 3.92$	$\underline{89.22}^{\dagger} \pm 1.83$	$\underline{91.53}^{\dagger} \pm 2.41$	$\underline{94.08}^{\dagger} \pm 0.98$	$\mathbf{98.40}^{\dagger} \pm 0.34$	$\mathbf{100} \pm 0$
STRURW-MIX	$\mathbf{88.48}^{\dagger} \pm 1.93$	$\mathbf{89.76}^{\dagger} \pm 1.15$	$\mathbf{92.08}^{\dagger} \pm 1.13$	$\mathbf{94.26}^{\dagger} \pm 0.99$	$\underline{98.35}^{\dagger} \pm 0.23$	$\mathbf{100} \pm 0$

- Performance decreases with increase in CSS (from smaller q to larger q)
- StruRW-based methods significantly outperform other baselines especially under large CSS

Principled solution: StruRW – Experiments

Table 4: HEP dataset with different PU conditions and Physical process. The **bold** font indicate the best model, † indicates the significant improvement, where the mean-1*std of a method > the mean of its corresponding backbone model.

	PU CONDITIONS				PHYSICAL PROCESSES	
DOMAINS	PU30 → 10	PU10 → 30	PU140 → 50	PU50 → 140	$gg \to Z(\nu\nu)$	$Z(\nu\nu) \to gg$
ERM	69.83 ± 0.43	70.73 ± 0.46	68.70 ± 0.56	68.28 ± 0.65	63.09 ± 0.48	66.53 ± 1.04
DANN	70.14 ± 0.52	71.29 ± 0.58	69.01 ± 0.42	68.98 ± 0.63	63.15 ± 0.66	66.24 ± 0.97
STRURW-ERM	$71.35^{\dagger} \pm 0.76$	$71.95^{\dagger} \pm 0.24$	$69.43^{\dagger} \pm 0.65$	69.05 ± 0.36	63.55 ± 0.40	$\mathbf{67.73} \pm 0.93$
STRURW-ADV	$\mathbf{70.77}^{\dagger} \pm 0.52$	$\mathbf{71.96} \pm 0.73$	$\mathbf{69.88}^{\dagger} \pm 0.71$	$\mathbf{70.54} \pm 0.84$	$\mathbf{64.36}^{\dagger} \pm 0.58$	66.91 ± 0.67

- StruRW-based methods perform better than baselines
- The smaller gap may be due to the physics task itself being:
  - Binary classification
  - Multigraph training and testing process

Content

- Problems: Various distribution shifts in scientific applications
- A detailed example: Pileup Mitigation
- A principled solution: StruRW algorithm
- Future Works

Principled solution: Future Works

- **Scientific Application:**
  - Extend the Pileup Mitigation project to real data setting
  - Currently try with some DA techniques to boost pileup mitigation performance
  - Seek for more applications that exist distribution shifts
- **ML on distribution shift**
  - Elaborate on the StruRW project with more general assumptions
  - An OOD benchmark project on various scientific applications (HEP, biology, material science)

Conclusion

- Distribution shifts are critical problems in scientific domain
- We have proposed series of work to handle distribution shifts
  - Masking in Pileup Mitigation project
  - More principled StruRW algorithm
- If are interested in our work

Pileup Mitigation Paper

StruRW Paper

StruRW Code

Thank you
Q & A

Thanks to NSF and A3D3 (OAC-2117997) for funding our research

Pileup Mitigation Paper

StruRW Paper

StruRW Code