Hierarchical Structure Sharing Empowers Multi-task Heterogeneous GNNs for Customer Expansion

Xinyue Feng, Shuxin Zhong, Jinquan Hang, Wenjun Lyv,
Yuequn Zhang, Guang Yang, Haotian Wang, Desheng Zhang, Guang Wang

Problem Definition

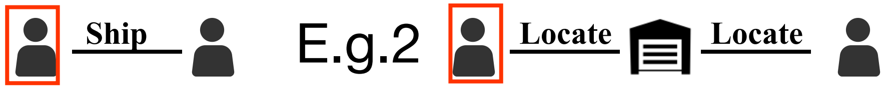
- **Customer Expansion**

  To ensure stable revenue streams -> Seek **high-value customers** willing to **sign long-term contracts** with our logistics company.

State-of-the-Arts Customer Expansion Methods

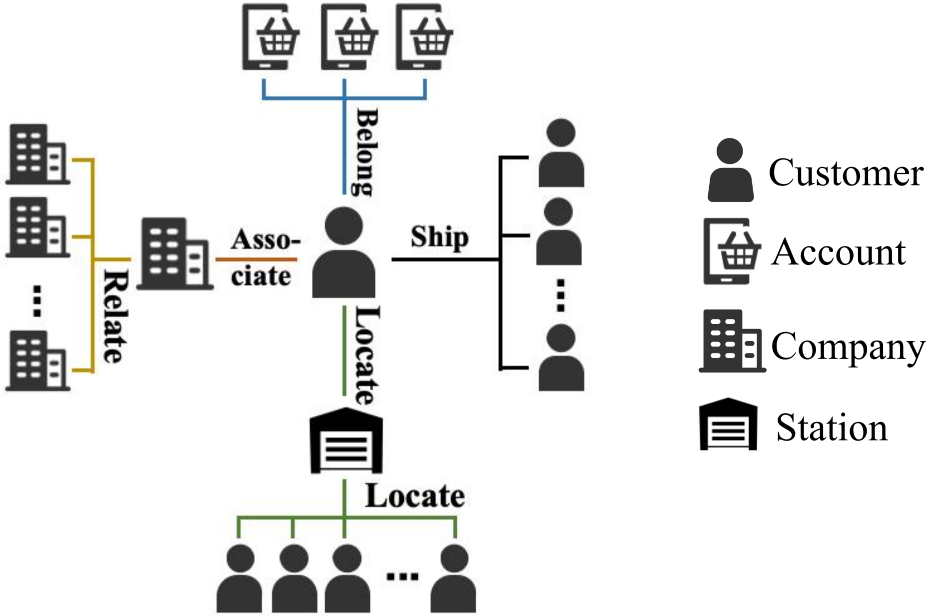
Learn **complex structural patterns** surrounding a customer by heterogeneous graph learning. [KDD22, CIKM23]

E.g.1 E.g.2



Predict

Whether is our target customer.



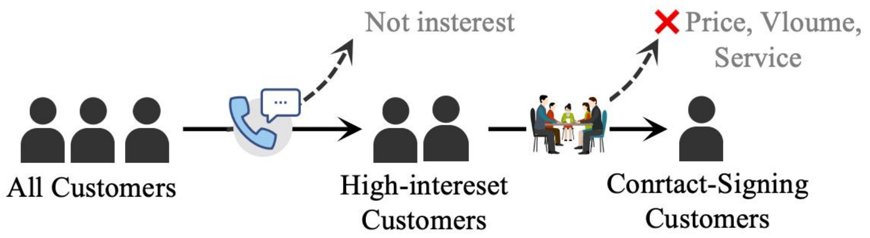
State-of-the-Arts Methods

- **Limitation**

When positive label (i.e., the number of contracted customers) is **extremely sparse**

Limit models' ability to learn meaningful structural patterns.

Compromising predictive performance



Opportunity

Correlation

(Task 1)
High Interest
Customer Predicion

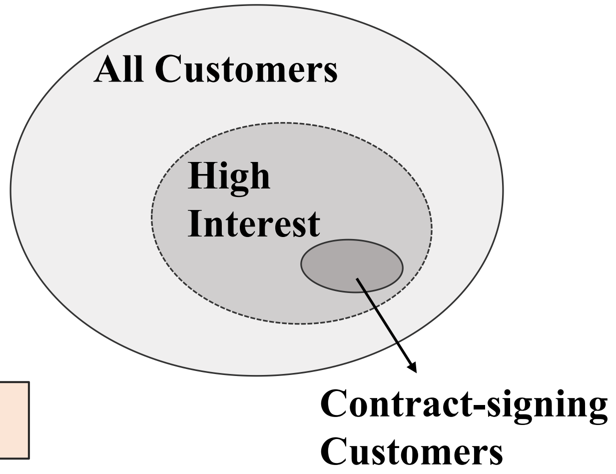
Abundant positive labels

Knowledge Sharing

by
Multi-Task Learning
(MTL)

(Task 2)
Contract-signing
Customer Predicion

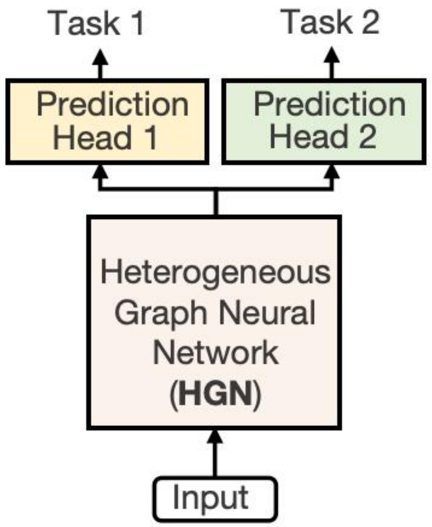
Scarce positive labels



Challenge

- **The most polular MTL framework results in performance degradation**

E.g. ***Customer-ship-Customer* (*C-C*)**



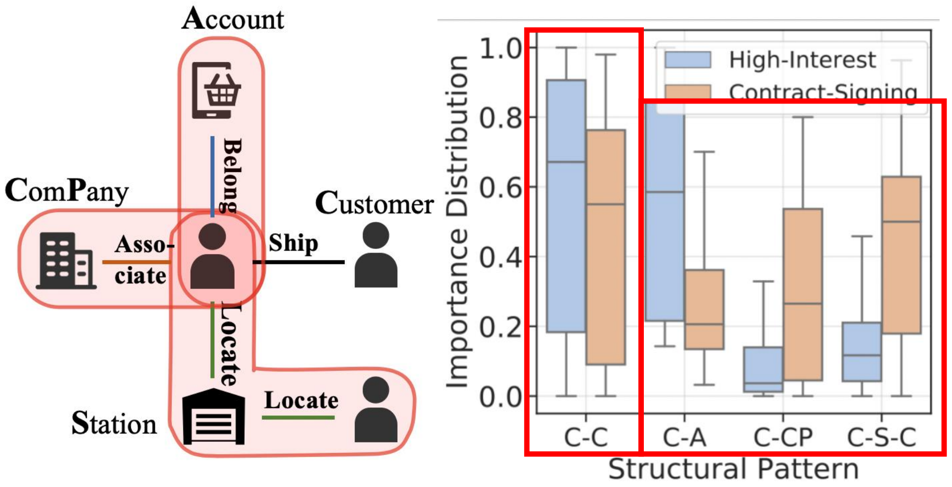
Shared-backbone Framework [KDD21, SIGIR22, KDD23]

Logistics	AUC	
	Task 1	Task 2
Single-task Learning	61.56	58.91
Shared-backbone MTL	60.38	56.46

Reason: Some learned structural patterns benefit for all tasks, while others are task-specific.

E.g. ***C-A, C-CP, C-S-C***

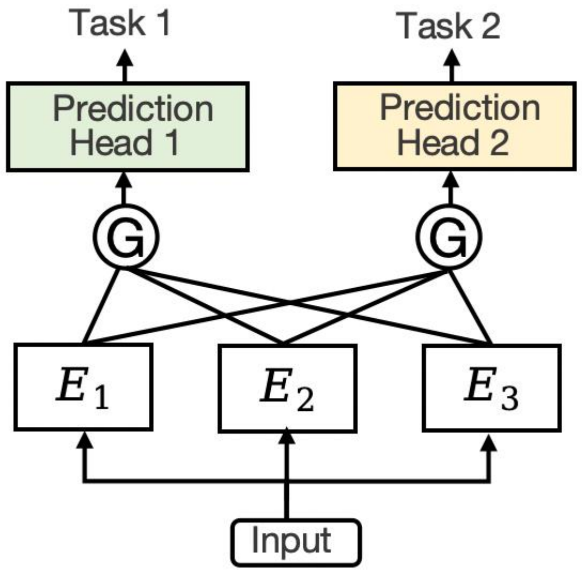
Visualization of the Learned Structural Pattern Importance



However, such shared-backbone design indiscriminately shares all structural patterns.

Challenge

- **State-of-the-Art MTL framework still struggle to solve the issue**



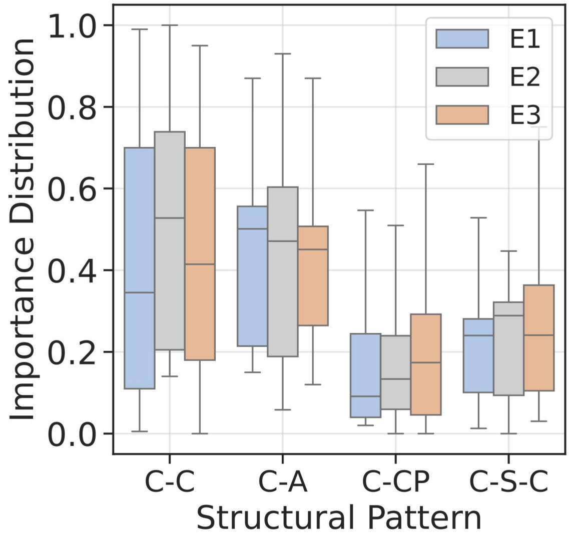
Mixture-of-Experts(MoE)-based MTL Framework [KDD18, SIGIR23]

Logistics	AUC	
	Task 1	Task 2
Single-task Learning	61.56	58.91
MoE-based MTL	60.74	57.36

Method: employ multiple backbones to **implicitly** learn task-specific or task-shared patterns separately.

Ideally,
E1/E3: learn task-specific structural pattern
E2: learn shared structural pattern

Reality:
E1/E2/E3 learn similar structural patterns.

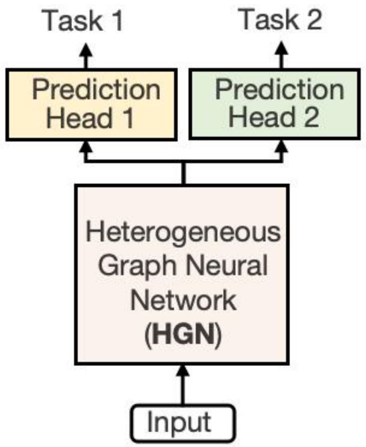


Though it is effective for non-graph scenarios, these **implicit** discrimination methods struggle with structural patterns in heterogeneous graph, which involves multi-layer aggregation of multi-type relations.

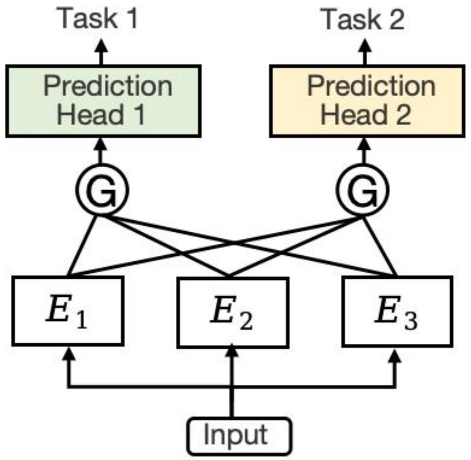
Our Method: Structure-Aware Hierarchical Information Sharing Framework (SrucHIS)

SoA MTL Framework

Shared-backbone



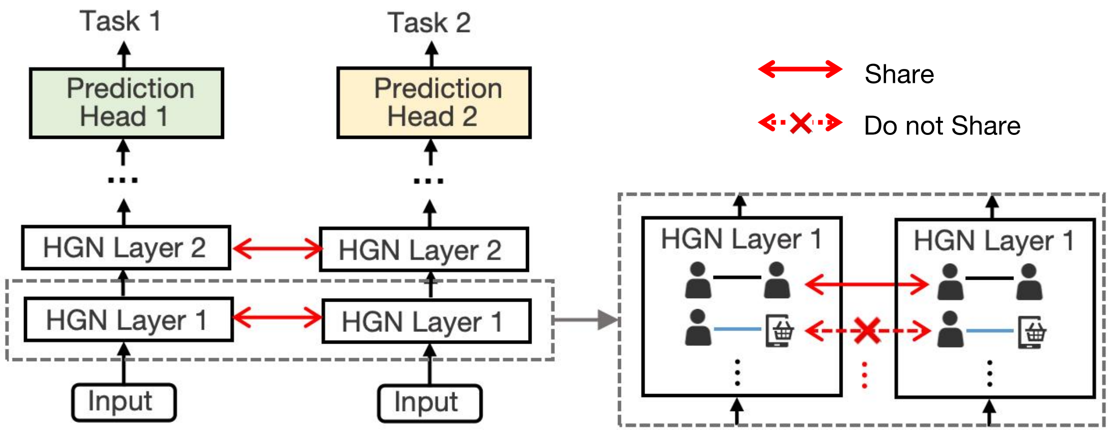
MoE-based



Indiscriminately shares all structural patterns

Implicitly regulate structure sharing

Our StrucHIS



Explicitly regulate structure sharing

- ➢ Hierarchically decomposes the structure learning process into **distinct stages (relation-wise & layer-wise).**
- ➢ Selective share structural patterns at each stage.

3 StrucHIS Framework

- **Logistics Graph Construction**

Nodes: *customer, company, account, and station.*
Edges: *relations between entities.*
Each node and edge is enriched with unique **feature** representations.

- **Feature Pre-processing**

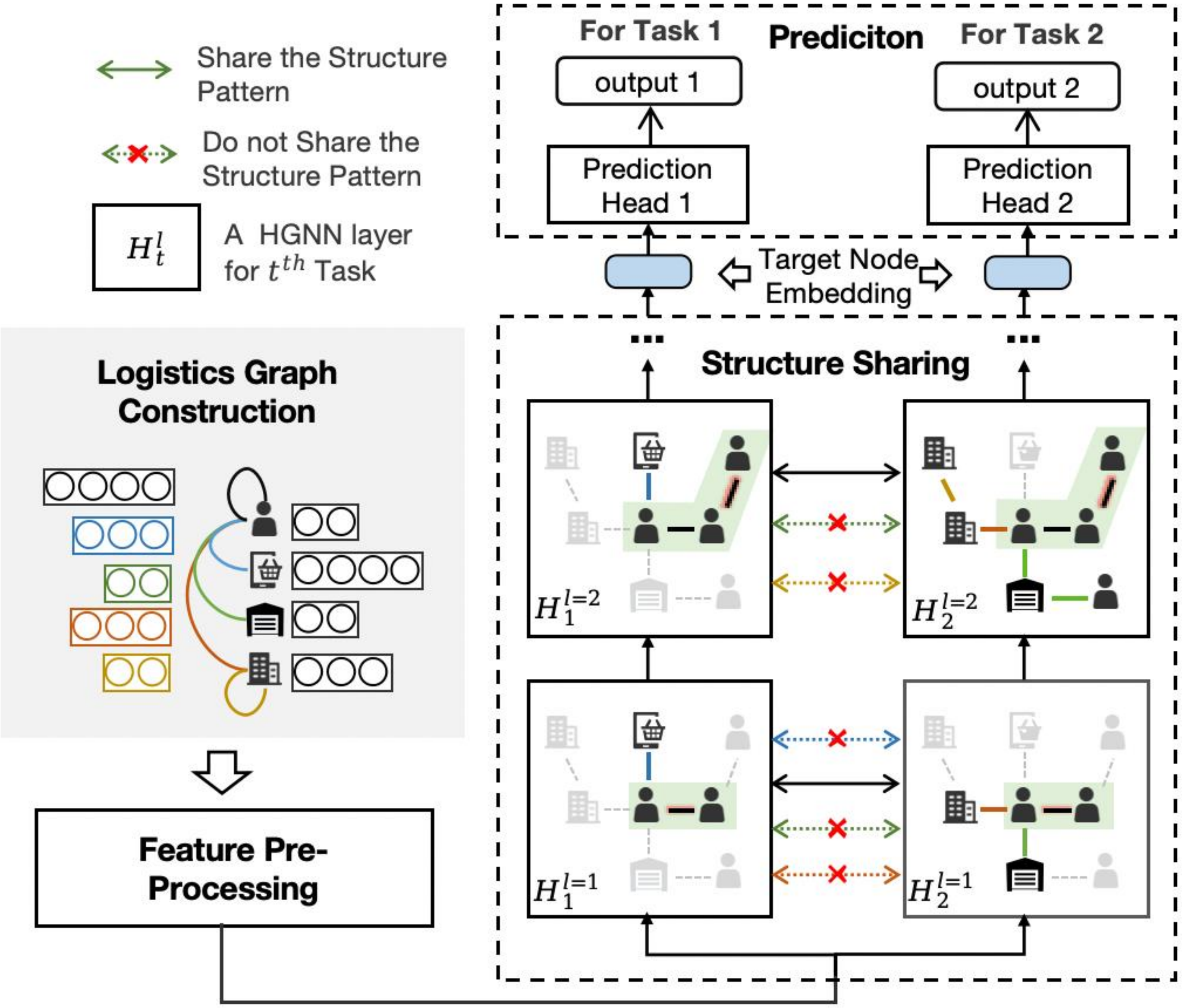
Project all types of features into the **identical** embedding length.

- **Structure Sharing**

The core of our framework

- **Prediction**

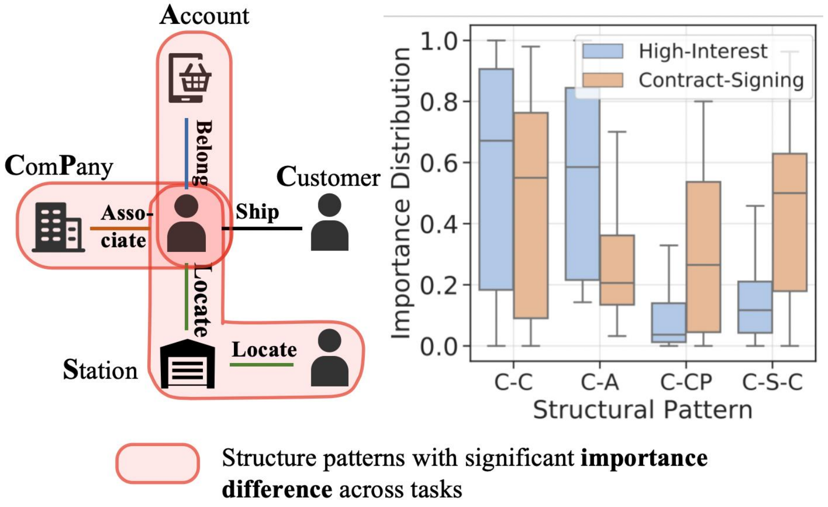
Task-specific prediction after L layers of structural learning and selective sharing.



3 Structure Sharing within a Single Layer

- **Motivation**

Consider a single-layer HGNN where a customer node aggregate information only from it 1-hop neighbors



Its 1-hop structural patterns may hold different importance for different tasks.

E.g.1 Customer-(ship)-Customer (C-C) **Share**

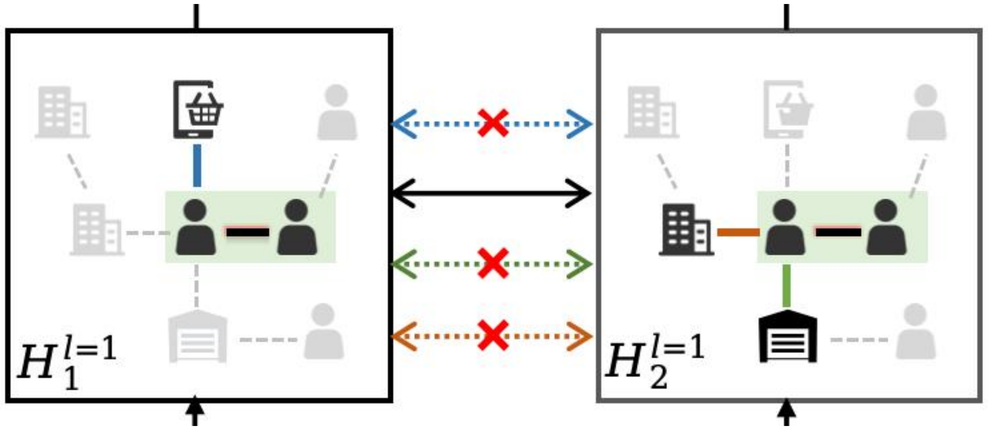
Important for both tasks.

E.g.2 Customer -(relate)-Account(C-A)

Important for task 1.

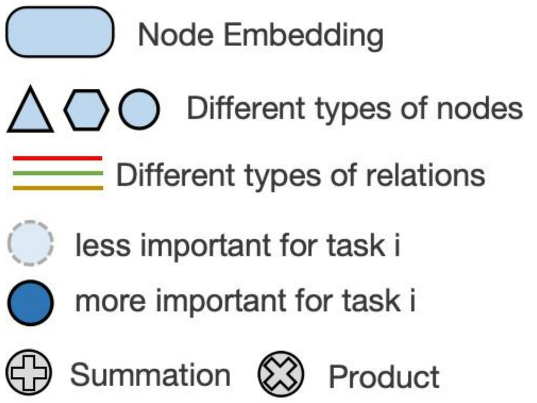
Less important for task2.

Do not Share



It is impratical to **maunally** decide what to share.
How to **automatically** control it?

3 Structure Sharing within a Single Layer



- **Step1: Relation-wise Information Aggregation**

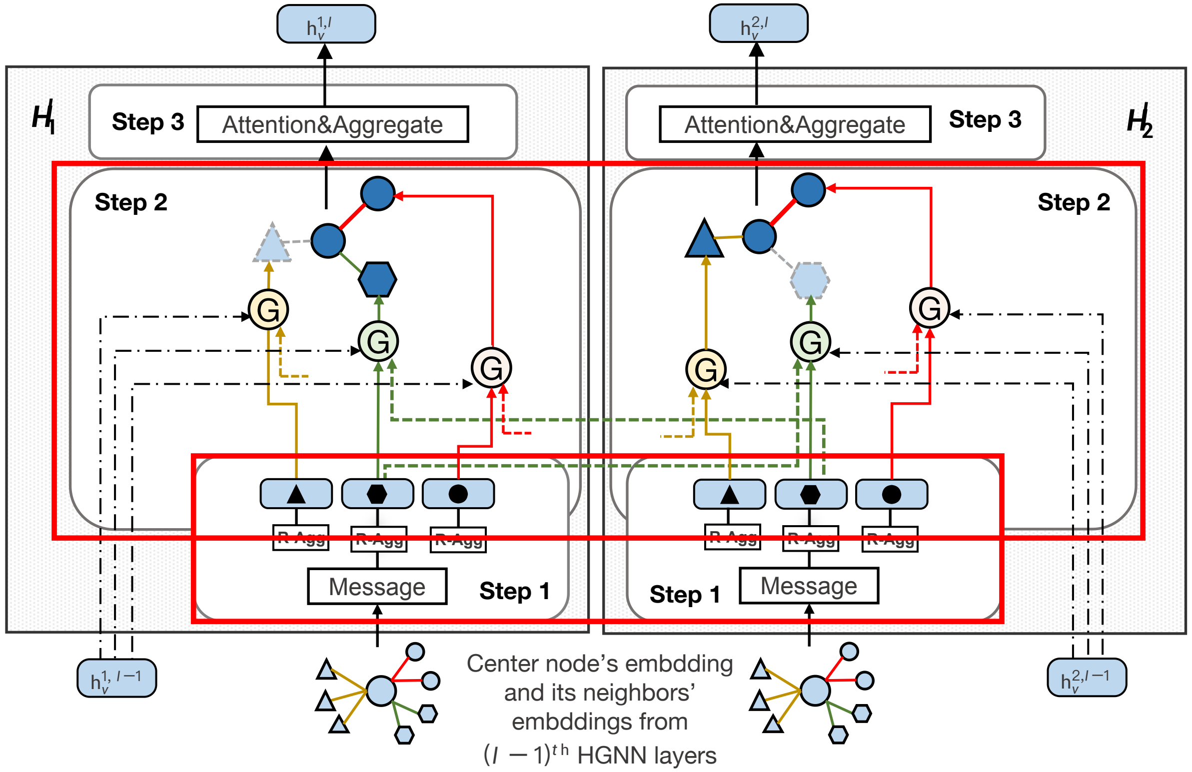
  Goal: extracts and **aggregates structural features** from neighboring nodes based on their specific relations.

- **Step2: Selective Information Sharing**

  Goal: selectively **share structural features** across tasks.

We introduce **Structure Aware Gates** G

The gate can automatically learn to decide the contribution of structural features from other tasks for the target task.



3 Structure Sharing within a Single Layer

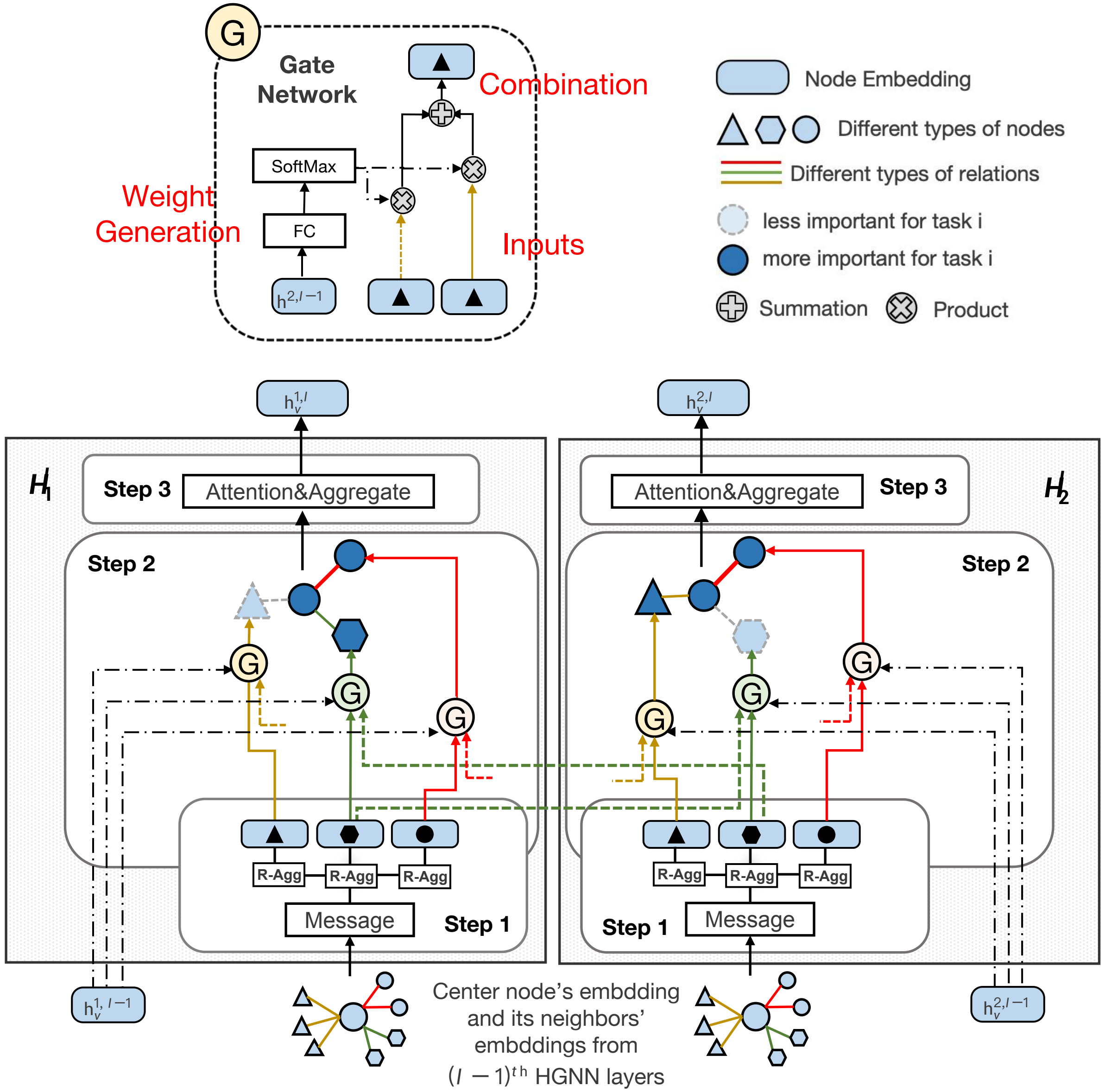
- **Step2: Selective Information Sharing**

Each gate is executed in three steps:

- **Inputs:** gather structural features from other tasks
- **Weight Generation:** Generate different weights for these features.
- **Combination:** Fuse these features based on different weights.

- **Step3: Aggregation Across Relations**

Aggreagte structural features experienced information sharing into the central node.



3 Apply Structure Sharing at Each Layer

- **Motivation**

Given a central customer node, rather than its **1-hop** structural patterns, its **n-hop** structural patterns may also hold different importance for different tasks.

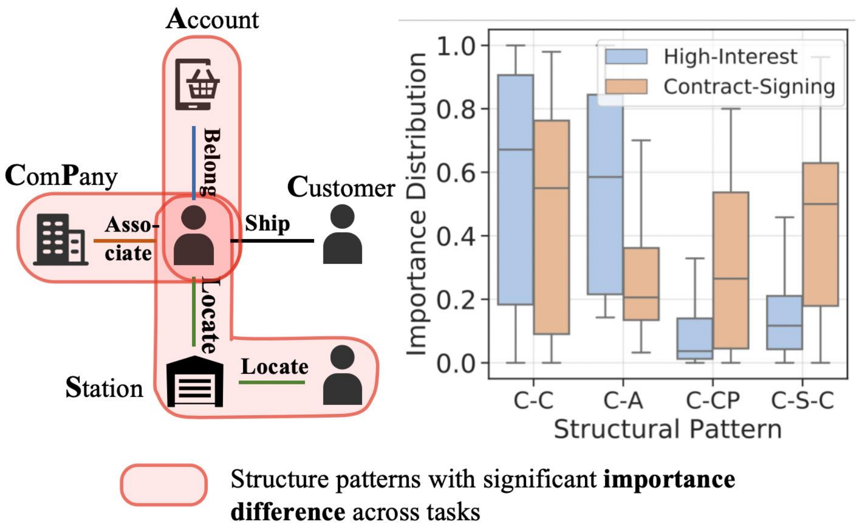
E.g. Customer-(Locate)-Station-(Locate)-Customer

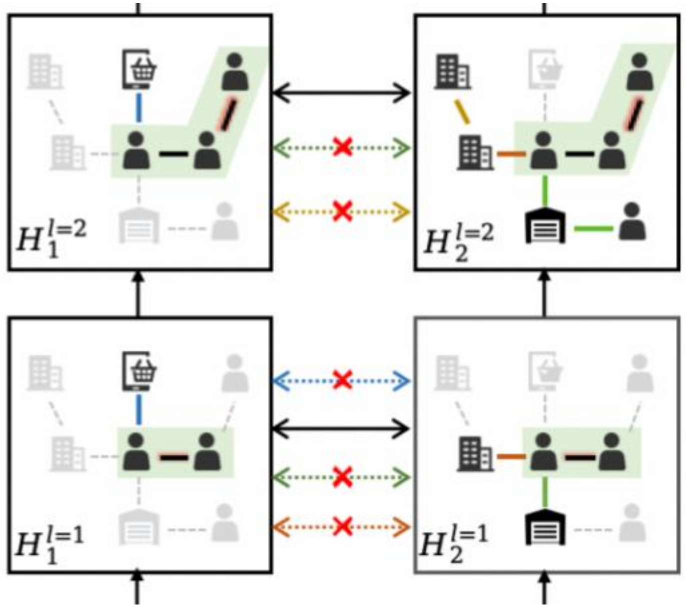
Important for task 2.

Less important for task 1.

n-hop structural pattern is learned at the $\boldsymbol{n}^{th}$ layer.

Thus, we apply Step1/2/3 not only at the first HGNN layer, but also at each HGNN layers,





4 Experiments

Dataset	# Nodes	# Node Types	# Edge	# Edge Types	# Task
DBLP	26,108	3	223,978	6	2
Aminer	28,253	2	139,831	4	2
Logistic	6,142,839	4	11,592,443	5	2

- **Datasets:**
  - (Private) **Logistic Dataset**
    - task1: High-interest Customer prediction
    - task2: Contract-Signing Customer prediction
  - (Public) **Aminer** - Citation dataset
    - task1: Author research areas classification
    - task2: Paper research fileds classification
  - (Public) **DBLP** - Citation dataset
    - task1: Author research areas classification
    - task2: Paper publciation venue prediction

- **Evaluation Metric**
  - Micro F1 & Macro F1: for DBLP and Aminer. (follow previous works)
  - AUC (area under ROC curve) & AP (Average Precision): for Logistic Datset

4 Experiments

- **Baselines:**
  - **Single-Task Learning (STL) Methods**
    - *STL-HGT [WWW20], STL-HGB [KDD21]*: select two widely-known **Heterogeneous Graph Neural Networks (HGNN)** models as our base models.
    - Setting: each task is independently trained.
  - **Shared-Backbone MTL Methods**
    - *MTL-HGT, MTL-HGB*
    - Setting: A shared HGNN backbone for all tasks + Separate prediction heads for each task
  - **MoE-based MTL Methods:**
    - *MMoE* [KDD18], *PLE* [RecSys20], *MultiSFS* [SIGIR23].
    - Setting: We implement their experts using same HGNN to ensure an equitable comparison with other methods.

4 Experiments

Our experiments focuses on answer 4 research questions:

- **RQ1:** Does StrucHIS solve the **Structural Interference Issue** in Logistics Dataset?
- **RQ2**: Can StrucHIS **generalize** to other **public** graph datasets?
- **RQ3: Are every modules** in StrucHIS effective?
- **RQ4**: How does the model perform in **Real-world Deployment?**

4 Experiments

RQ1: *Does StrucHIS solve the Structural Interference Issue in Logistics Dataset?*

Dataset	Logistics Dataset			
Tasks	High-Interest		Contract-Signing	
Metric	AUC	AP	AUC	AP
STL-HGT	$60.23_{\pm 0.17}$	$4.27_{\pm 0.04}$	$56.11_{\pm 0.19}$	$1.02_{\pm 0.03}$
STL-HGB	$61.56_{\pm 0.25}$	$4.51_{\pm 0.08}$	$58.91_{\pm 0.21}$	$1.42_{\pm 0.02}$
Shared-HGT	$59.09_{\pm 0.40}$	$4.11_{\pm 0.11}$	$56.10_{\pm 0.34}$	$1.07_{\pm 0.09}$
Shared-HGB	$60.38_{\pm 0.33}$	$4.31_{\pm 0.09}$	$56.45_{\pm 0.28}$	$1.12_{\pm 0.06}$
MMOE	$60.46_{\pm 0.74}$	$4.38_{\pm 0.17}$	$56.98_{\pm 0.53}$	$1.25_{\pm 0.12}$
PLE	$60.51_{\pm 0.88}$	$4.42_{\pm 0.21}$	$57.36_{\pm 0.87}$	$1.31_{\pm 0.17}$
MultiSFS	$60.74_{\pm 0.65}$	$4.49_{\pm 0.19}$	$57.32_{\pm 0.91}$	$1.29_{\pm 0.13}$
StrucHIS	$\mathbf{65.21_{\pm 0.49}}$	$\mathbf{5.67_{\pm 0.14}}$	$\mathbf{63.44_{\pm 0.43}}$	$\mathbf{2.15_{\pm 0.12}}$

Poor performance due to positive label sparsity

performance drop

solve

performance drop

solve

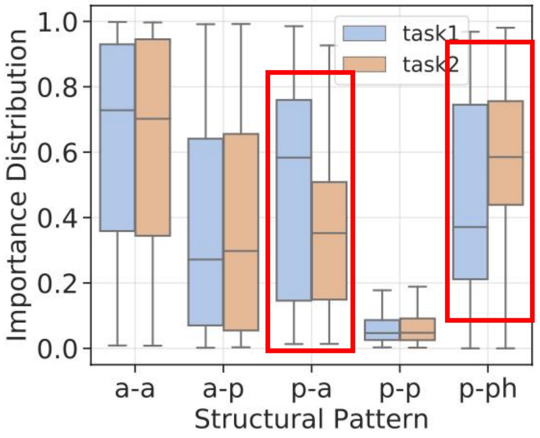
Multi-task learning methods

StrucHIS achieve a **51.41%** improvement in Average Precision on Contract-Signing Prediction Task.
(AP is a very important metric in real industry scenarios.)

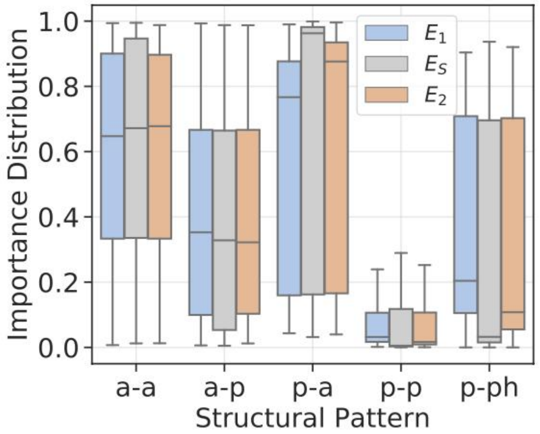
4 Experiments

***RQ2**: Can StrucHIS **generalize** to other **public** graph datasets?*

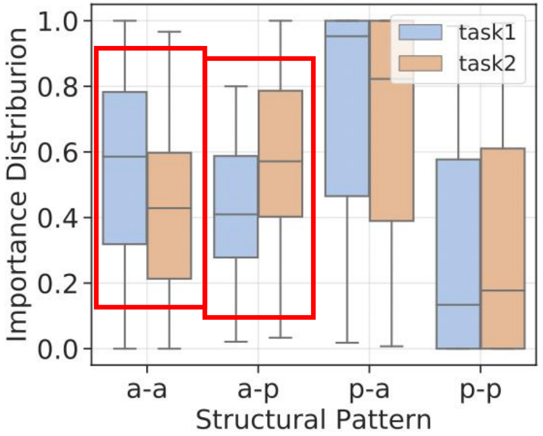
- ➢ Our visulization shows that, Structural Interference not only exists in our Logistics Dataset, but also **exists in two widely-used public graph datasets DBLP and Aminer**.
- ➢ On public datases, SoA MTL methods still struggle to solve the issue.



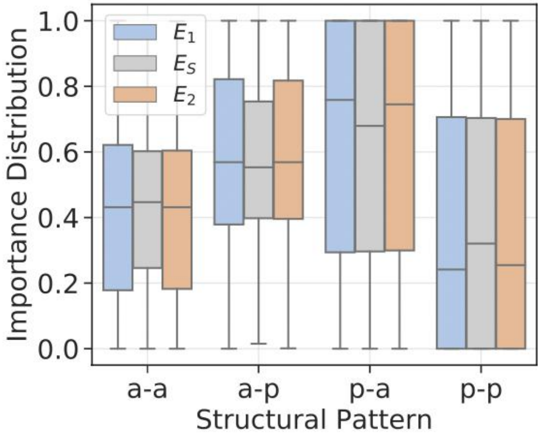
(a) STL Method on DBLP



(b) MoE MTL Method on DBLP



(a) STL Method on Aminer



(b) MoE MTL Method on Aminer

4 Experiments

***RQ2**: Can StrucHIS **generalize** to other **public** graph datasets?*

Dataset	Aminer				DBLP			
Tasks	task1		task2		task1		task2	
Metric	micro F1	macro F1	micro F1	macro F1	micro F1	macro F1	micro F1	macro F1
STL-HGT	$90.03_{\pm 0.50}$	$90.05_{\pm 0.49}$	$92.51_{\pm 0.17}$	$92.57_{\pm 0.21}$	$82.67_{\pm 0.76}$	$81.82_{\pm 0.87}$	$35.39_{\pm 0.47}$	$18.63_{\pm 1.07}$
STL-HGB	$91.16_{\pm 0.19}$	$91.64_{\pm 0.19}$	$92.58_{\pm 0.22}$	$92.62_{\pm 0.24}$	$85.51_{\pm 0.74}$	$84.67_{\pm 0.78}$	$37.39_{\pm 0.20}$	$19.10_{\pm 1.60}$
Shared-HGT	$93.35_{\pm 0.47}$	$93.78_{\pm 0.44}$	$93.38_{\pm 0.24}$	$93.41_{\pm 0.23}$	$84.71_{\pm 0.49}$	$83.59_{\pm 0.46}$	$32.90_{\pm 0.51}$	$12.86_{\pm 1.02}$
Shared-HGB	$93.89_{\pm 0.28}$	$94.28_{\pm 0.25}$	$93.41_{\pm 0.18}$	$93.48_{\pm 0.16}$	$91.02_{\pm 0.36}$	$90.34_{\pm 0.27}$	$36.34_{\pm 0.98}$	$14.21_{\pm 1.87}$
MMOE	$93.63_{\pm 0.66}$	$93.86_{\pm 0.64}$	$93.20_{\pm 0.28}$	$93.24_{\pm 0.28}$	$91.01_{\pm 0.23}$	$90.29_{\pm 0.25}$	$35.18_{\pm 0.57}$	$13.12_{\pm 0.75}$
PLE	$93.72_{\pm 0.13}$	$94.13_{\pm 0.14}$	$92.96_{\pm 0,33}$	$93.07_{\pm 0.35}$	$91.22_{\pm 0.30}$	$90.38_{\pm 0.33}$	$35.59_{\pm 0.98}$	$14.23_{\pm 1.29}$
MultiSFS	$93.47_{\pm 0.30}$	$94.17_{\pm 0.28}$	$93.19_{\pm 0.38}$	$93.16_{\pm 0.35}$	$91.10_{\pm 0.13}$	$90.47_{\pm 0.14}$	$36.14_{\pm 0.69}$	$14.56_{\pm 0.99}$
StrucHIS	$\mathbf{95.01_{\pm 0.41}}$	$\mathbf{95.34_{\pm 0.38}}$	$\mathbf{93.52_{\pm 0.19}}$	$\mathbf{93.54_{\pm 0.37}}$	$\mathbf{91.43_{\pm 0.28}}$	$\mathbf{90.56_{\pm 0.39}}$	$\mathbf{38.25_{\pm 0.87}}$	$\mathbf{20.92_{\pm 1.14}}$

Multi-task learning methods

performance drop

solve

further Improvement

StrucHIS significantly alleviate the performance drop caused by structural interference, achieving a **10.52%** macro F1 gain on public dataset DBLP.

4 Experiments

RQ3: *Is **every modules** in StrucHIS effective?*

➢ **(w/o R&L) Without Both Relation-wise and Layer-wise Sharing:**

Tasks use separate HGNN backbones with information sharing only at the final output embedding.

➢ **(w/o R) Without Relation-wise Sharing, Only use Layer-wise sharing**

Selective Information Sharing is applied only at layer-level.

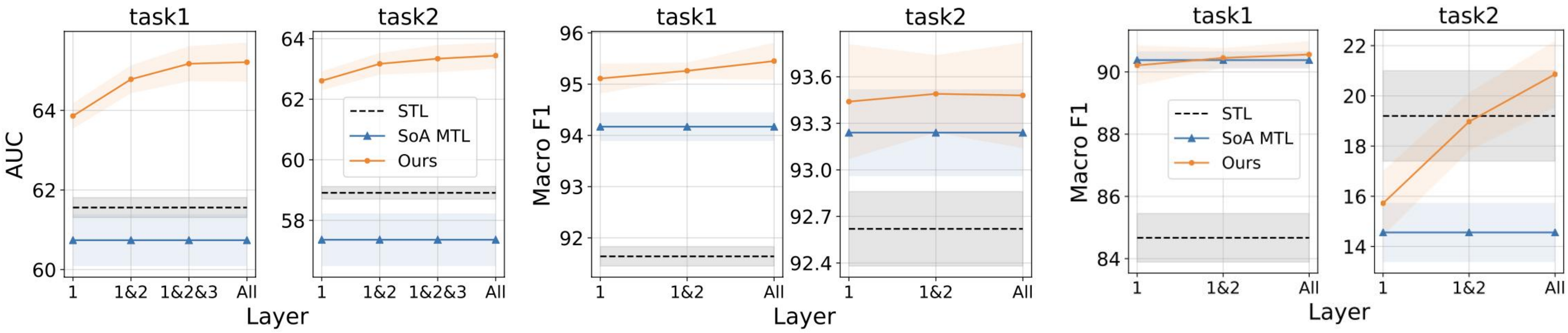
Dataset	Logistic		DBLP		Aminer	
Task	Task1	Task2	Task1	Task2	Task1	Task2
w/o R&L	60.43	56.82	91.11	35.76	93.64	93.01
w/o R	60.58	57.31	91.25	36.21	93.78	93.16
StrucHIS	**65.21**	**63.44**	**91.43**	**38.25**	**95.01**	**93.52**

Layer-wise sharing alone is insufficient and **necessitates relation-wise sharing** to mitigate structural interference.

4 Experiments

RQ3: *Is **every modules** in StrucHIS effective?*

➢ **The necessity of Layer-wise sharing (applying relation-wise sharing at each layer)**

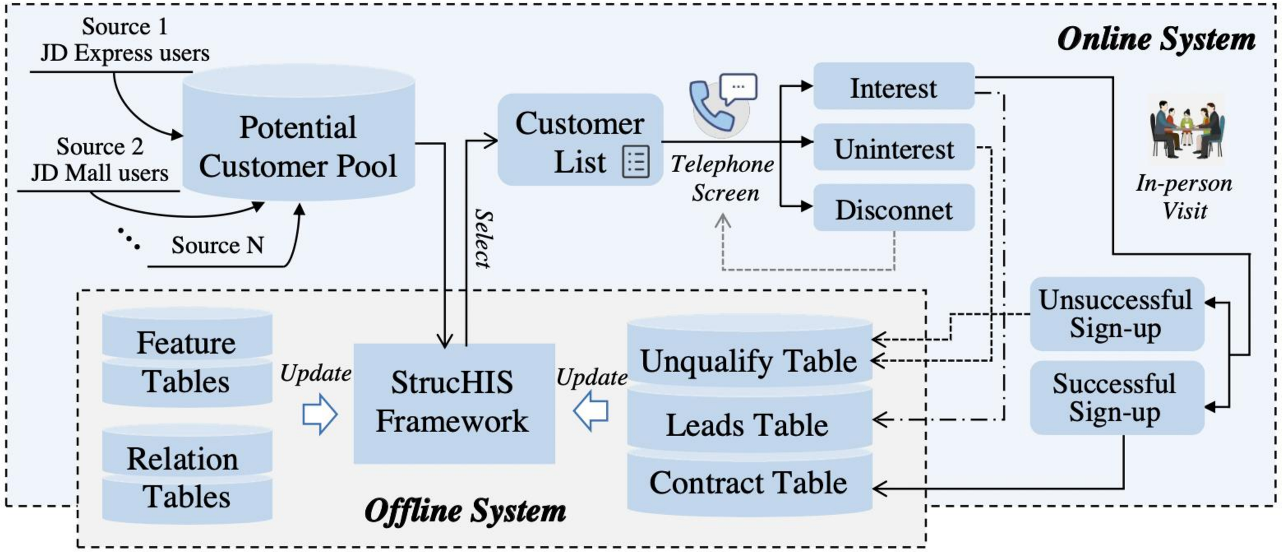


(a) Logistics (b) Aminer (c) DBLP

4 Experiments

RQ4: *How does the model perform in* ***Real-world Deployment?***

Deployment System



Online A/B Testing

Group	Method	# High-Interest	# Contract-Signing
Control	JD's Method	254	12
	STL-HGB	272 (+7.1%)	13 (+8.3%)
Treatment	StrucHIS	336 (+32.3%)	17 (+41.7%)

StrucHIS demonstrates a **41.67%** improvement in the success contract-signing rate over existing strategies, generating over **453K** new orders within just two months..

5 Contribution

- **Conceptually:**

We observe and analyze a critical yet previously under-researched issue based on our large-scale industry data: **cross-task structural interference** in multi-task heterogeneous graph neural network. We highlight that SoA MTL approaches struggle with such interference.

- **Technically:**

We introduce SrucHIS, a heterogeneous graph-based MTL framework that **explicitly regulates structural knowledge sharing** to mitigate cross-task interference..

- **Experimentally:**

➢ **JD Logistics** (one of the largest logistics companies in China): Average Precision **51.41%** .

➢ **Two public datasets**: Macro F1 scores **10.52%** .

➢ **Real-world Deployment**: Success Contract-signing Rate **41.67%** . New orders **453K** .

I'm looking for **2025/2026**
Full-time Internship !

LLM | GNN | Data Mining

fengxy26@gmail.com

For more about me, LinkedIn

THANK YOU!