Meta-learning on Heterogeneous Information Networks for Cold-start Recommendation

Yuanfu Lu[1], **Yuan Fang[2]**, Chuan Shi[1]

[1]Beijing University of Posts and Telecommunications

[2]Singapore Management University

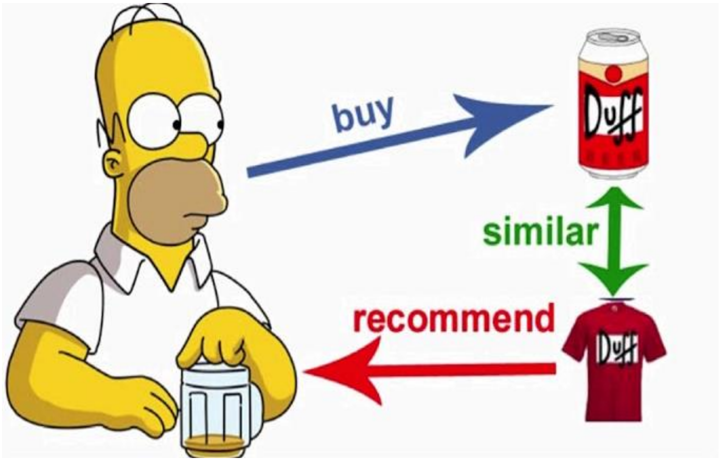
Outline

- Motivation
- MetaHIN
- Experiments
- Conclusions

Outline

- **Motivation**
- MetaHIN
- Experiments
- Conclusions

Cold-start Recommendation



Recommender System

- collaborative filtering
- content-based filtering
- …

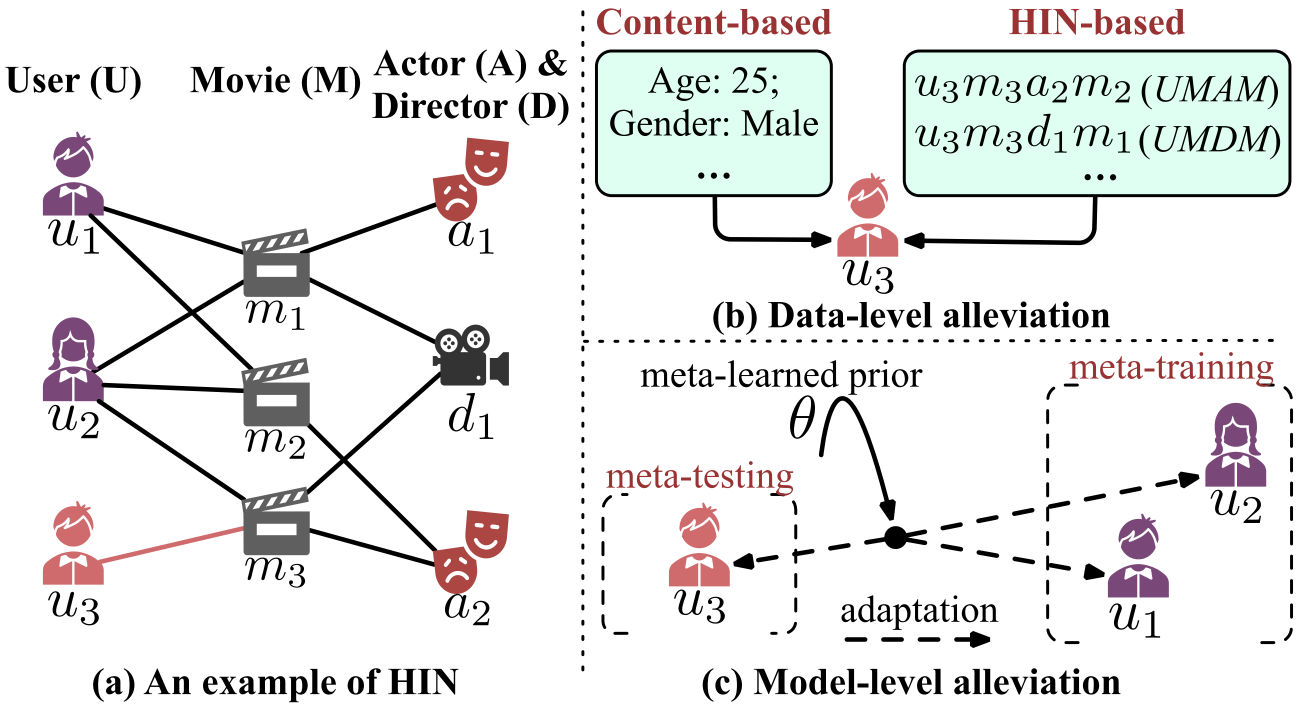
What about a new user or a new item?

Cold-start Problem

- New users or new items
- The interaction data is very sparse

Images from the Internet.

Existing Methods



Existing alleviations

- Data level
  - Content-based
  - HIN-based
- Model level
  - Meta-learning

Our Idea

Address the cold-start problem
at both data and model levels?

Exploit the power of both meta-learning at the model level and HINs at the data level

NON-TRIVIAL !

Challenges

C1: How to model HINs in the meta-learning setting?

- *Existing methods model HINs under traditional supervised or unsupervised learning settings*

C2: How to model the general knowledge across tasks?

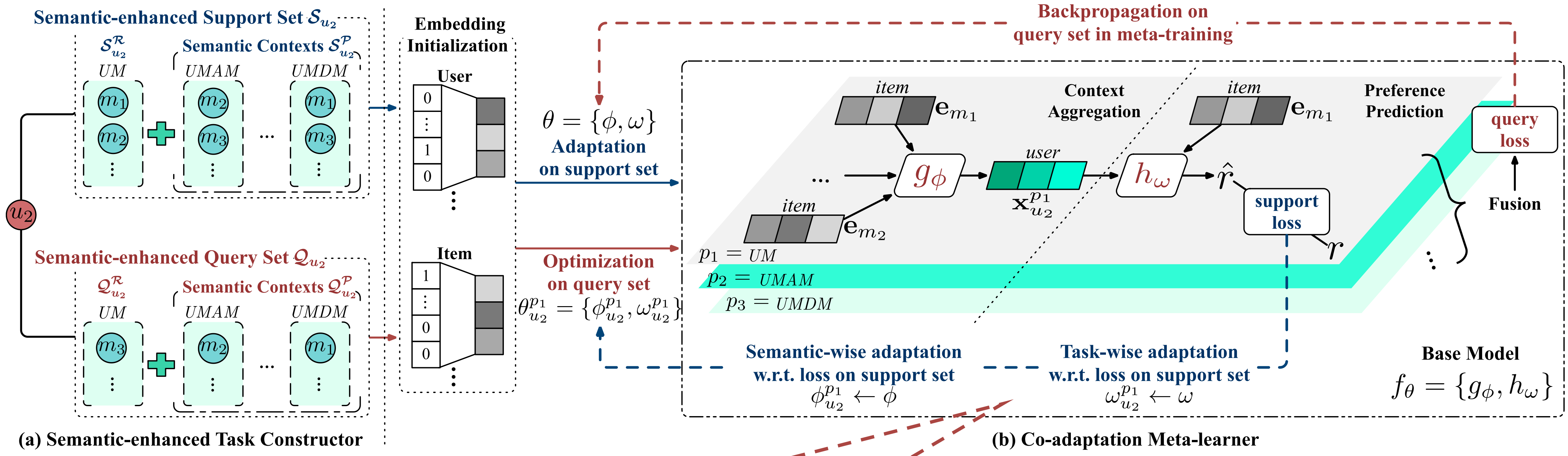
- *Previous work: Only adapt to new tasks (e.g., new users) from a globally shared prior*
- *Our work: there exist multifaceted semantics brought by HINs*

Outline

- Motivation
- MetaHIN
- Experiments
- Conclusions

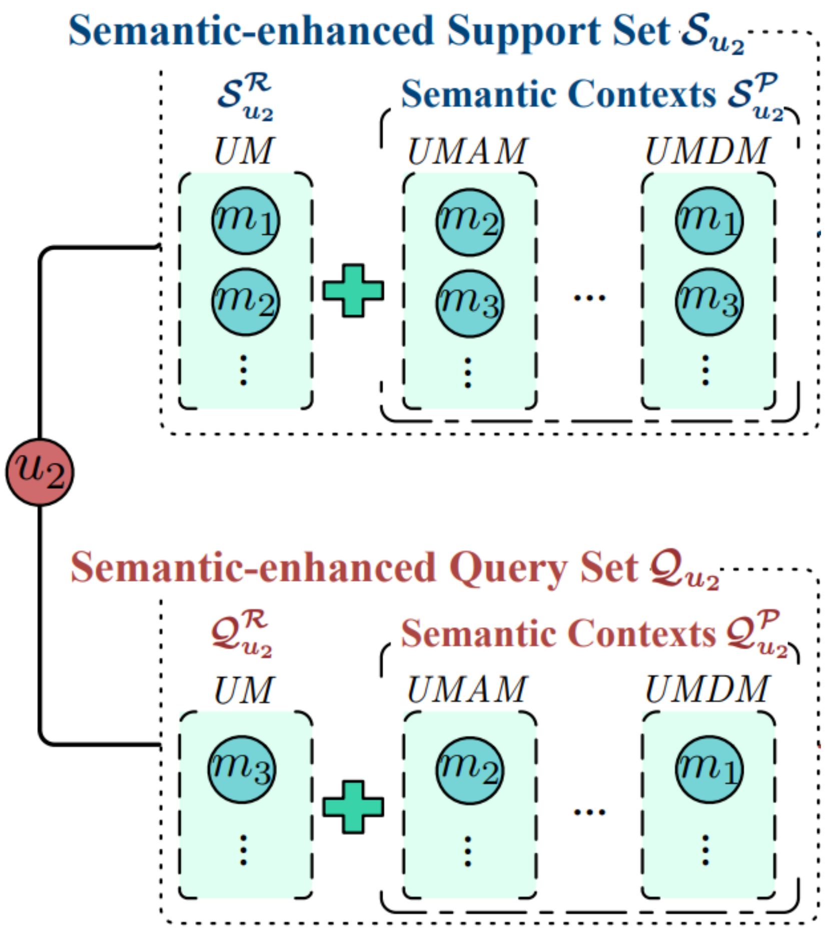
Overall Framework of MetaHIN

C1:
Task construction augmented with semantic contexts



C2:
Semantic and task-wise adaptations

Semantic-enhanced Task Constructor



(a) Semantic-enhanced Task Constructor

Support set of u $\mathcal{S}_u = (\mathcal{S}^{\mathcal{R}}_u, \mathcal{S}^{\mathcal{P}}_u)$

rated items

semantically related items

↓

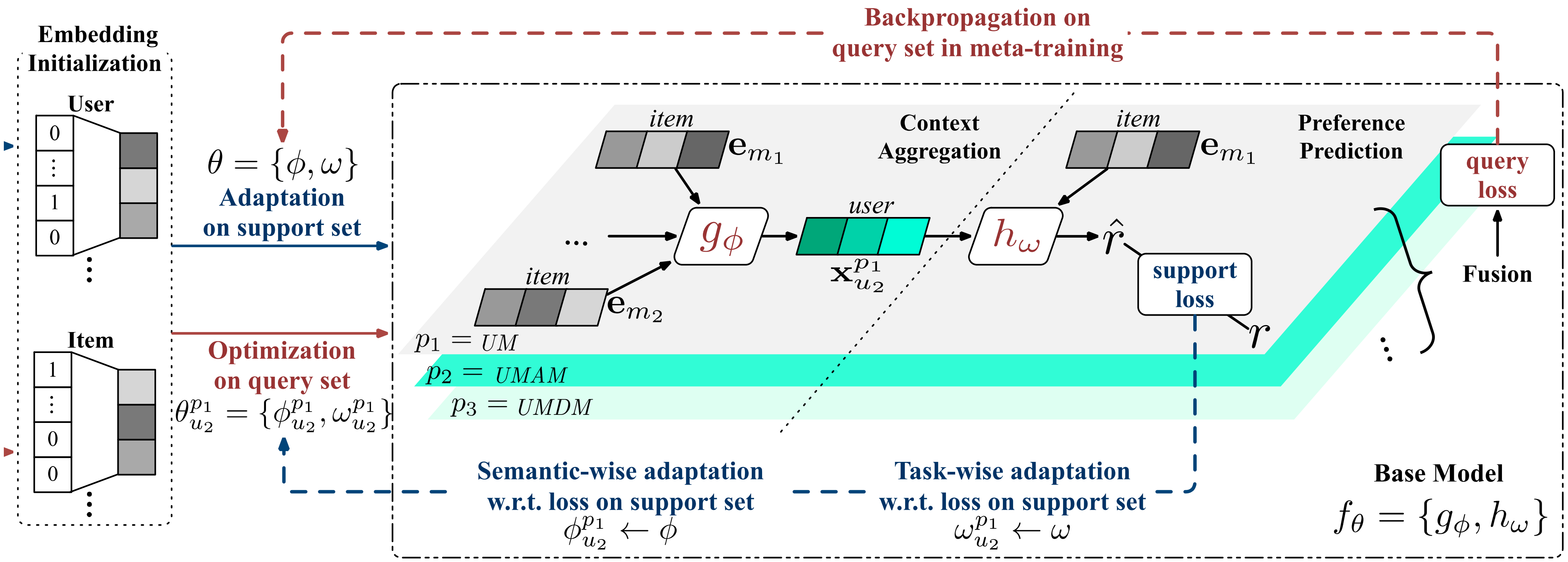
meta-path reachable items

$$\mathcal{S}^{\mathcal{P}}_u = (\mathcal{S}^{p_1}_u, \mathcal{S}^{p_2}_u, \ldots, \mathcal{S}^{p_n}_u)$$

$$\mathcal{S}^{p}_u = \bigcup_{i \in \mathcal{S}^{\mathcal{R}}_u} C^{p}_{u,i}$$

$$C^{p}_{u,i} = \{j : j \in \text{items reachable along } p \text{ starting from } u\text{–}i\}$$

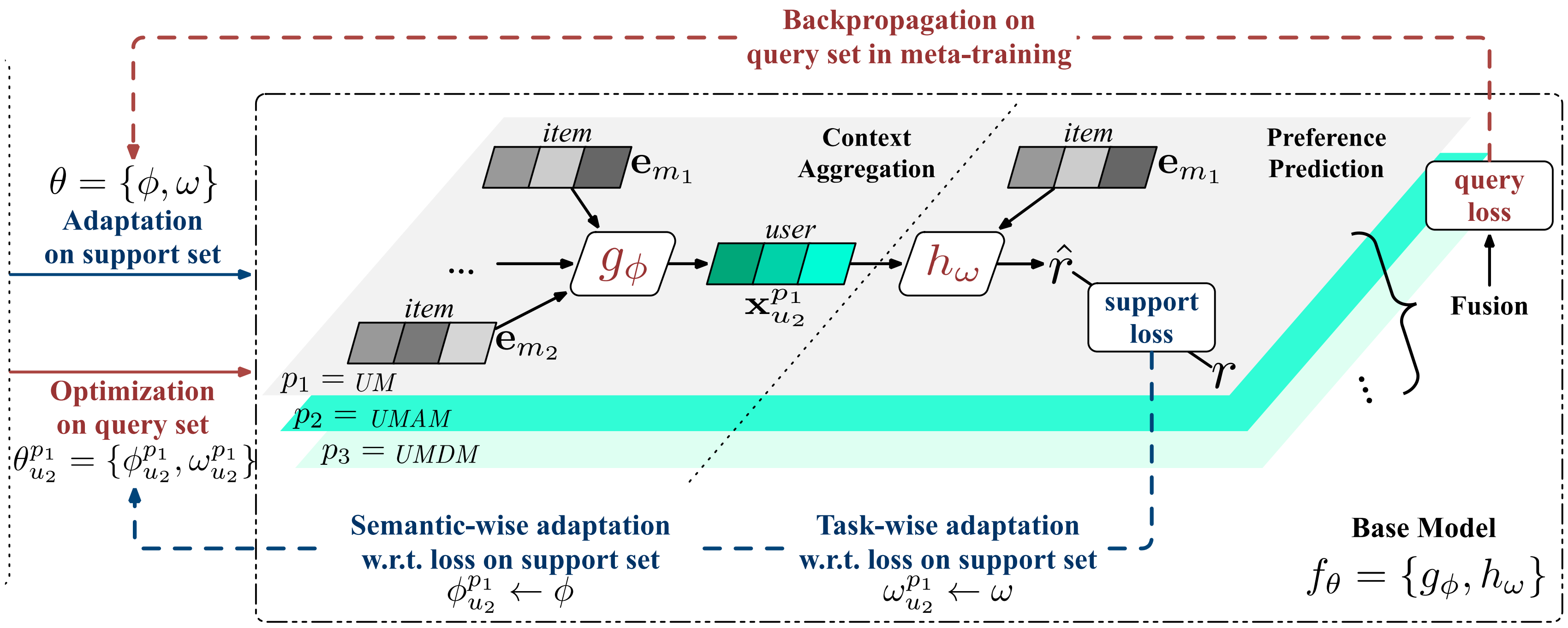
Co-adaptation Meta-learner



(b) Co-adaptation Meta-learner

- **Base Model** $f_\theta = (g_\phi, h_\omega)$ parameterized by $\theta = \{\phi, \omega\}$
- $\mathbf{x}_u = g_\phi(u, C_u) = \sigma(\mathrm{MEAN}(\{\mathbf{W}e_j + \mathbf{b}: j \in C_u\}))$
- $\hat{r}_{ui} = h_\omega(\mathbf{x}_u, \mathbf{e}_i) = \mathrm{MLP}(\mathbf{x}_u \oplus \mathbf{e}_i)$

Co-adaptation Meta-learner



(b) Co-adaptation Meta-learner

- semantic-wise adaptation $\phi_u^p = \phi - \alpha \dfrac{\partial \mathcal{L}_{\mathcal{T}_u}(\omega, \mathbf{x}_u^p, \mathcal{S}_u^{\mathcal{R}})}{\partial \phi}$
- task-wise adaptation $\omega_u^p = \omega^p - \beta \dfrac{\partial \mathcal{L}_{\mathcal{T}_u}(\omega^p, \mathbf{x}_u^{p\langle \mathcal{S} \rangle}, \mathcal{S}_u^{\mathcal{R}})}{\partial \omega^p}$

Optimization

- Objective function to optimize global prior $\theta = \{\phi, \omega\}$

rating prediction model adapted to *u*

user preferences (embeddings) adapted to *u*

$$\min_{\theta} \sum\nolimits_{\mathcal{T}_u \in \mathcal{T}^{tr}} \mathcal{L}_{\mathcal{T}_u}(\omega_u, \mathbf{x}_u, \mathcal{Q}_u^{\mathcal{R}})$$

where

ground truth

predicted rating

$$\mathcal{L}_{\mathcal{T}_u}(\omega_u, \mathbf{x}_u, \mathcal{Q}_u^{\mathcal{R}}) = \sum\nolimits_{i \in Q_u^R} (r_{ui} - \hat{r}_{ui})^2$$

Outline

- Motivation
- MetaHIN
- **Experiments**
- Conclusions

Experiments

- How does MetaHIN perform compared to state-of-the-art approaches?
- How does MetaHIN benefit from the multifaceted semantic contexts and co-adaptation meta-learner?
- How is MetaHIN impacted by its hyper-parameters?

Setup

- **Datasets**
  - Dbook: #node: 42,070, #edge: 839,465
  - MovieLens: #node: 20,137, #edge: 1,019,817
  - Yelp: #node: 86,874, #edge: 1,429,218
- **3+1 scenarios**
  - Three cold-start scenarios:
    - (UC) User Cold-start, i.e., recommendation of existing items for new users;
    - (IC) Item Cold-start, i.e., recommendation of new items for existing users;
    - (UIC) User-Item Cold-start, i.e., recommendation of new items for new users
  - One traditional scenario
    - recommendation of existing items for existing users

Performance Comparison (RQ1)

Table 2: Experimental results in four recommendation scenarios and on three datasets. A smaller MAE or RMSE value, and a larger nDCG@5 value indicate a better performance. The best method is bolded, and second best is underlined.

Scenario	Model	DBook			MovieLens			Yelp		
		MAE ↓	RMSE ↓	nDCG@5 ↑	MAE ↓	RMSE ↓	nDCG@5 ↑	MAE ↓	RMSE↓	nDCG@5 ↑
Existing items for new users (User Cold-start or UC)	FM	0.7027	0.9158	0.8032	1.0421	1.3236	0.7303	0.9581	1.2177	0.8075
	NeuMF	0.6541	0.8058	0.8225	0.8569	1.0508	0.7708	0.9413	1.1546	0.7689
	GC-MC	0.9061	0.9767	0.7821	1.1513	1.3742	0.7213	0.9321	1.1104	0.8034
	mp2vec	0.6669	0.8391	0.8144	0.8793	1.0968	0.8233	0.8972	1.1613	0.8235
	HERec	0.6518	0.8192	0.8233	0.8691	0.9916	0.8389	0.8894	1.0998	0.8265
	DropoutNet	0.8311	0.9016	0.8114	0.9291	1.1721	0.7705	0.8557	1.0369	0.7959
	MeteEmb	0.6782	0.8553	0.8527	0.8261	1.0308	0.7795	0.8988	1.0496	0.7875
	MeLU	0.6353	0.7733	0.8793	0.8104	0.9756	0.8415	0.8341	1.0017	0.8275
	MetaHIN	**0.6019**	**0.7261**	**0.8893**	**0.7869**	**0.9593**	**0.8492**	**0.7915**	**0.9445**	**0.8385**
New items for existing users (Item Cold-start or IC)	FM	0.7186	0.9211	0.8342	1.3488	1.8503	0.7218	0.8293	1.1032	0.8122
	NeuMF	0.7063	0.8188	0.7396	0.9822	1.2042	0.6063	0.9273	1.1009	0.7722
	GC-MC	0.9081	0.9702	0.7634	1.0433	1.2753	0.7062	0.8998	1.1043	0.8023
	mp2vec	0.7371	0.9294	0.8231	1.0615	1.3004	0.6367	0.7979	1.0304	0.8337
	HERec	0.7481	0.9412	0.7827	0.9959	1.1782	0.7312	0.8107	1.0476	0.8291
	DropoutNet	0.7122	0.8021	0.8229	0.9604	1.1755	0.7547	0.8116	1.0301	0.7943
	MeteEmb	0.6741	0.7993	0.8537	0.9084	1.0874	0.8133	0.8055	0.9407	0.8092
	MeLU	0.6518	0.7738	0.8882	0.9196	1.0941	0.8041	0.7567	0.9169	0.8451
	MetaHIN	**0.6252**	**0.7469**	**0.8902**	**0.8675**	**1.0462**	**0.8341**	**0.7174**	**0.8696**	**0.8551**
New items for new users (User-Item Cold-start or UIC)	FM	0.8326	0.9587	0.8201	1.3001	1.7351	0.7015	0.8363	1.1176	0.8278
	NeuMF	0.6949	0.8217	0.8566	0.9686	1.2832	0.8063	0.9860	1.1402	0.7836
	GC-MC	0.7813	0.8908	0.8003	1.0295	1.2635	0.7302	0.8894	1.1109	0.7923
	mp2vec	0.7987	1.0135	0.8527	1.0548	1.2895	0.6687	0.8381	1.0993	0.8137
	HERec	0.7859	0.9813	0.8545	0.9974	1.1012	0.7389	0.8274	0.9887	0.8034
	DropoutNet	0.8316	0.8489	0.8012	0.9635	1.1791	0.7617	0.8225	0.9736	0.8059
	MeteEmb	0.7733	0.9901	0.8541	0.9122	1.1088	0.8087	0.8285	0.9476	0.8188
	MeLU	0.6517	0.7752	0.8891	0.9091	1.0792	0.8106	0.7358	0.8921	0.8452
	MetaHIN	**0.6318**	**0.7589**	**0.8934**	**0.8586**	**1.0286**	**0.8374**	**0.7195**	**0.8695**	**0.8521**
Existing items for existing users (Non-cold-start)	FM	0.7358	0.9763	0.8086	1.0043	1.1628	0.6493	0.8642	1.0655	0.7986
	NeuMF	0.6904	0.8373	0.7924	0.9249	1.1388	0.7335	0.7611	0.9731	0.8069
	GC-MC	0.8056	0.9249	0.8032	0.9863	1.2238	0.7147	0.8518	1.0327	0.8023
	mp2vec	0.6897	0.8471	0.8342	0.8788	1.1006	0.7091	0.7924	1.0191	0.8005
	HERec	0.6794	0.8409	0.8411	0.8652	1.0007	0.7182	0.7911	0.9897	0.8101
	DropoutNet	0.7108	0.7991	0.8268	0.9595	1.1731	0.7231	0.8219	1.0333	0.7394
	MeteEmb	0.7095	0.8218	0.7967	0.8086	1.0149	0.8077	0.7677	0.9789	0.7740
	MeLU	0.6519	0.7834	0.8697	0.8084	0.9978	0.8433	0.7382	0.9028	0.8356
	MetaHIN	**0.6393**	**0.7704**	**0.8859**	**0.7997**	**0.9491**	**0.8499**	**0.6952**	**0.8445**	**0.8477**

w.r.t. MAE

- **Dbook:** **3.05-5.26%**
- **MovieLens:** **2.89-5.55%**
- **Dbook:** **2.22-5.19%**

Performance Comparison (RQ1)

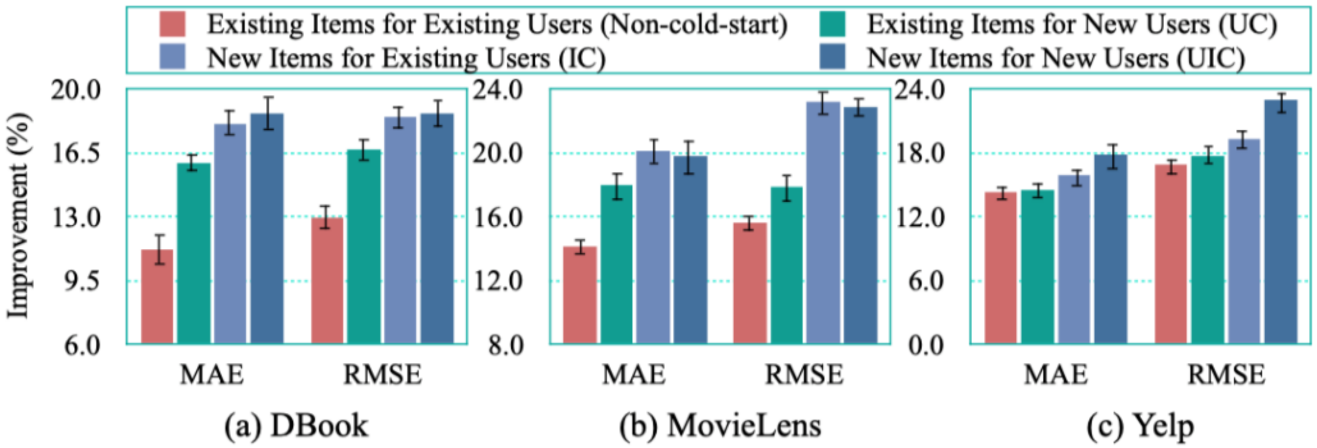


Figure 3: Performance improvement of MetaHIN.

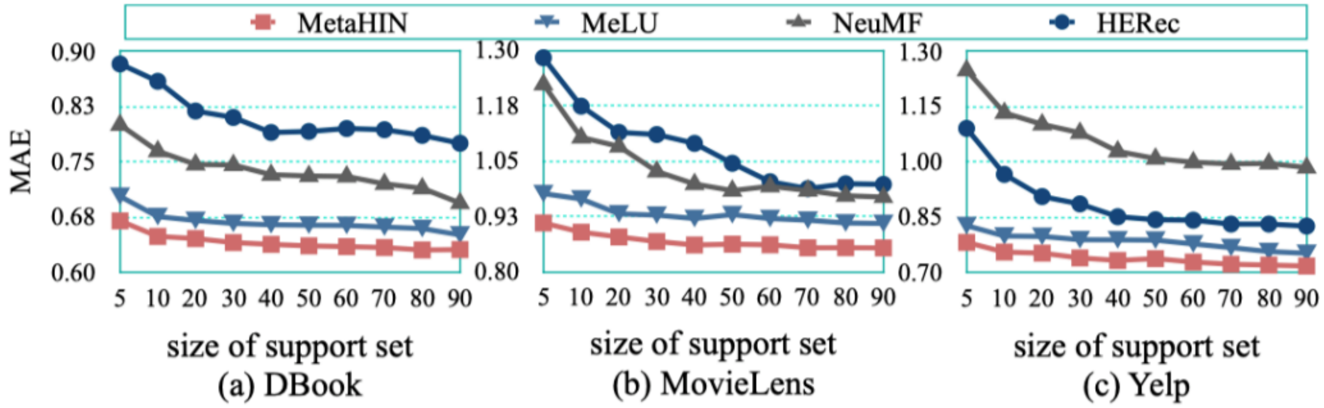


Figure 4: Impact of the size of support sets in UIC scenario.

- **improvement**

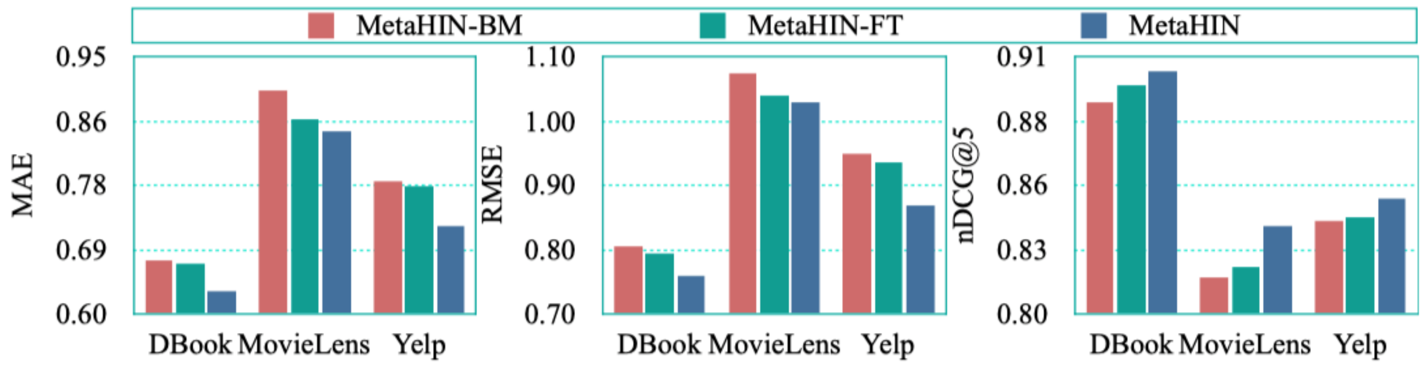
  non-cold-start < UC ~ IC < UIC

- **support set**

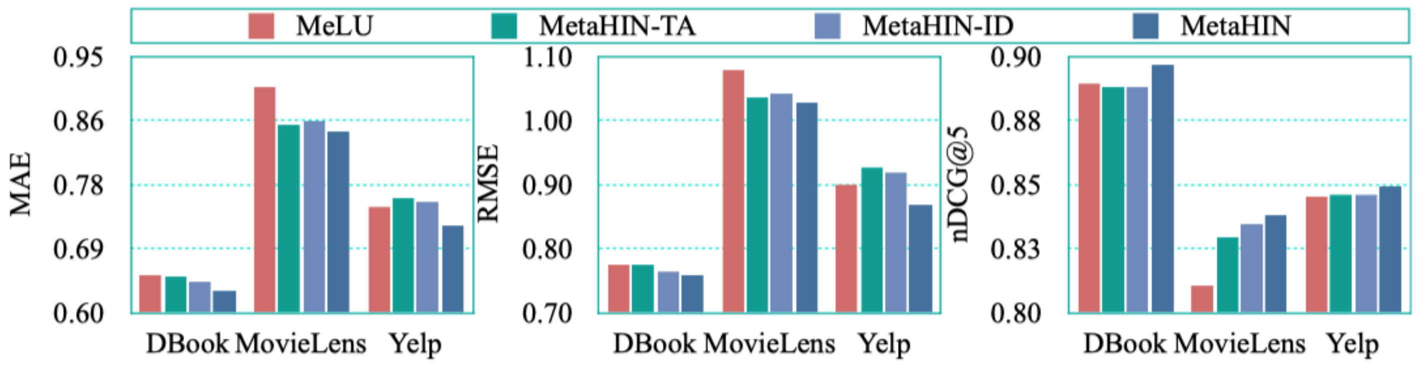
  the larger the set, the better the performance;

  MetaHIN is robust to set size.

Model Analysis (RQ2)



(a) Effect of meta-learning.



(b) Effect of semantic contexts and co-adaptation.

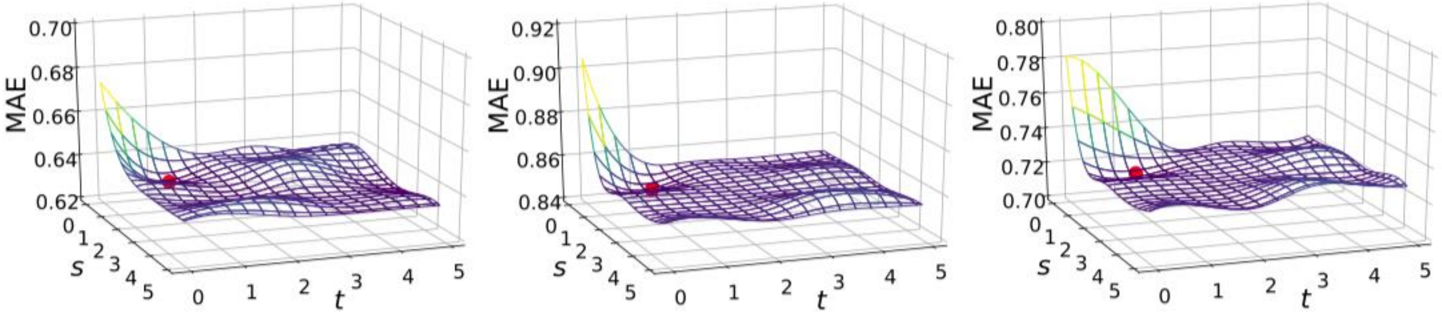
- **MetaHIN-BM**
  base model without meta-learning
- **MetaHIN-FT**
  fine-tune the base model

- **MetaHIN-TA**
  only task-wise adaptation
- **MetaHIN-ID**
  independently adopts task-wise adaptation

Parameter Analysis (RQ3)

Number of Co-adaptations

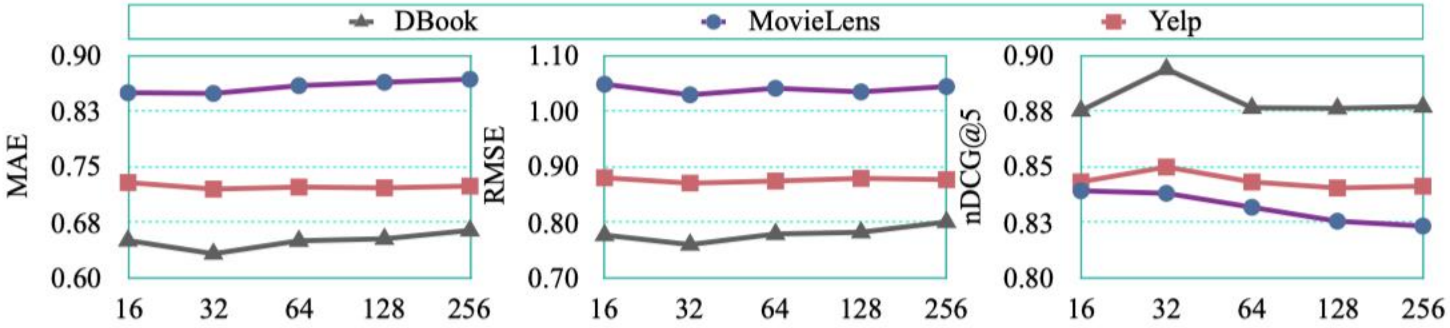
s and *t* are the number of semantic- and task-wise adaption step



(a) Impact of semantic-wise (s) and task-wise adaptation steps (t).

Embedding Dimensions

d is the dimensions of user embeddings



(b) Impact of user embedding dimensions.

Outline

- Motivation
- MetaHIN
- Experiments
- **Conclusions**

Conclusions

- **MetaHIN** alleviates the cold-start problem at both data and model levels.
- A **semantic-enhanced task constructor** to explore rich semantics on HINs in the meta-learning setting.
- A **co-adaptation meta-learner** with semantic- and task-wise adaptions to cope with different semantic facets within each task.
- Extensive experiments on three datasets.

Thank you !

Q&A

More materials in
http://shichuan.org
http://www.yfang.site
https://yuanfulu.github.io

北京郵電大學
Beijing University of Posts and Telecommunications
SMU
SINGAPORE MANAGEMENT UNIVERSITY