ACM SIGKDD Conference on Knowledge Discovery and Data Mining
(KDD 2024)

Item-Difficulty-Aware Learning Path Recommendation: From a Real Walking Perspective

Haotian Zhang, Shuanghong Shen, Bihan Xu
Zhenya Huang, Jinze Wu, Jing Sha, Shijin Wang

认知智能全国重点实验室
STATE KEY LABORATORY OF COGNITIVE INTELLIGENCE

Reporter: Haotian Zhang

CONTENTS

KDD2024
BARCELONA, SPAIN

CONTENTS

KDD2024 BARCELONA, SPAIN

Background

➢ **Learning path recommendation** aims to provide learners with a reasonable order of items to achieve their learning goals.

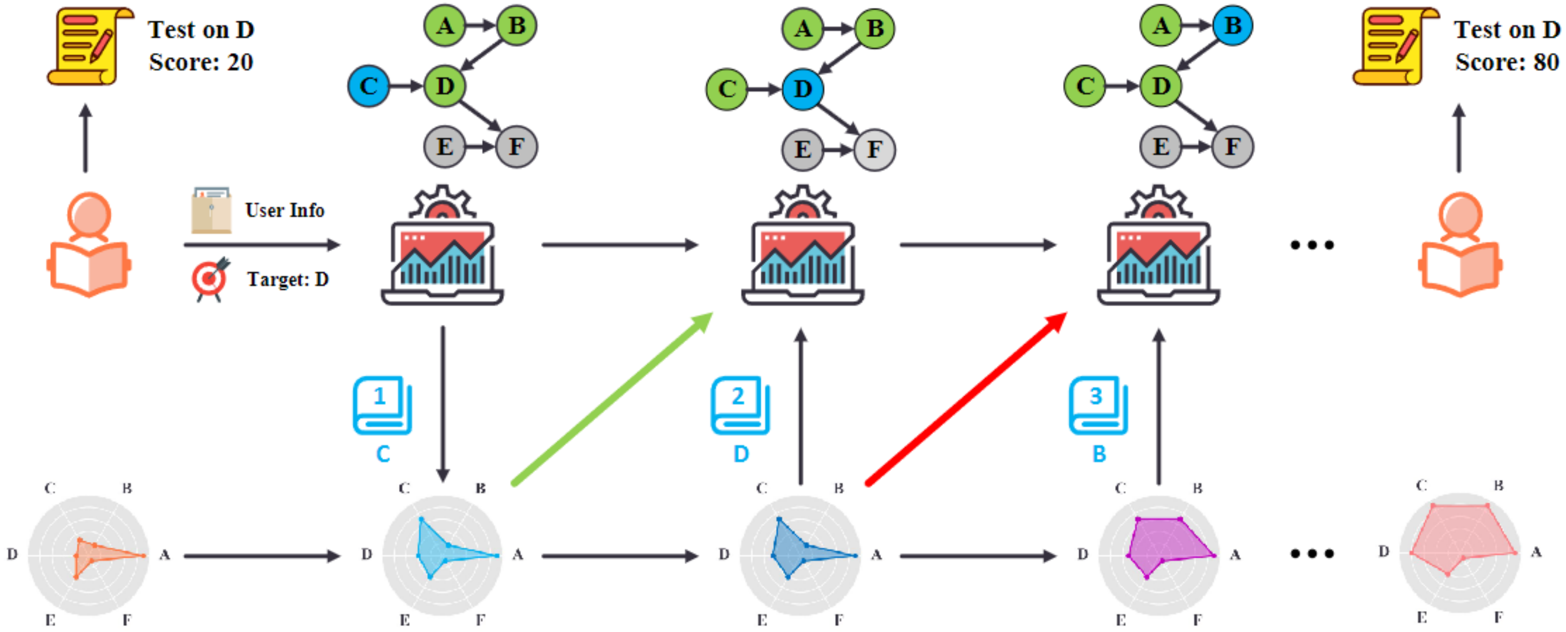
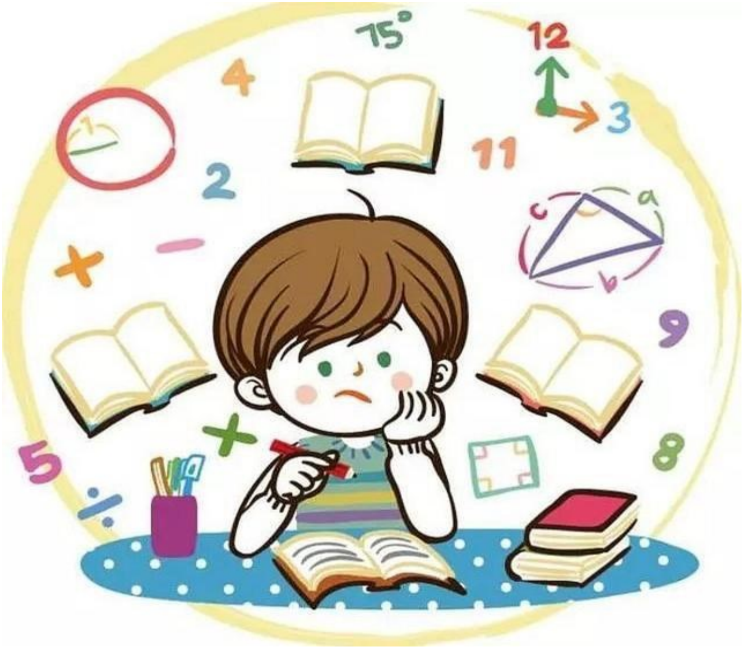


Figure1: Illustration of Learning Path Recommendatioin [1].

[1] Liu, Qi, et al. "Exploiting cognitive structure for adaptive learning." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.

Background

- Intuitively, the **learning process** on the learning path can be metaphorically **likened to walking**. Just as walking involves taking one step at a time to reach a destination, learning requires progressing through items methodically to achieve mastery.



Background

- Despite extensive efforts in this area, most previous methods mainly focus on the relationship among items but overlook the difficulty of items, which may raise two issues from a real walking perspective: **“rough” and “inefficient”**

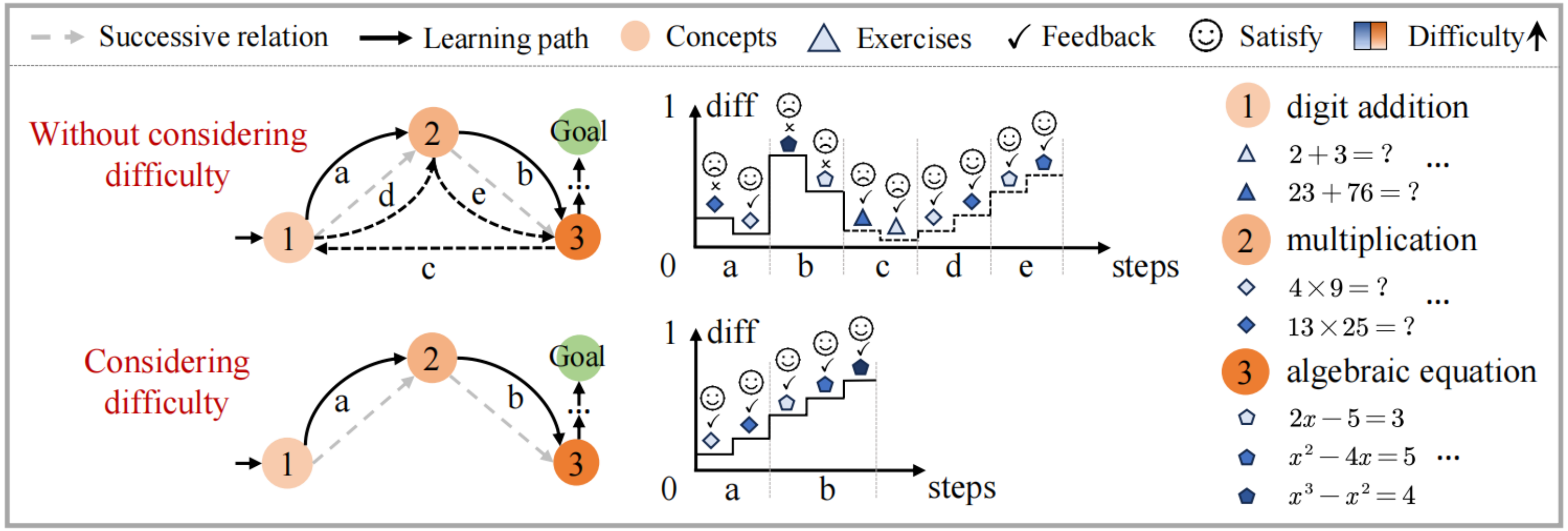


Figure 2: Contrasting Learning Path Recommendation: Considering vs. Without Considering Item Difficulty.

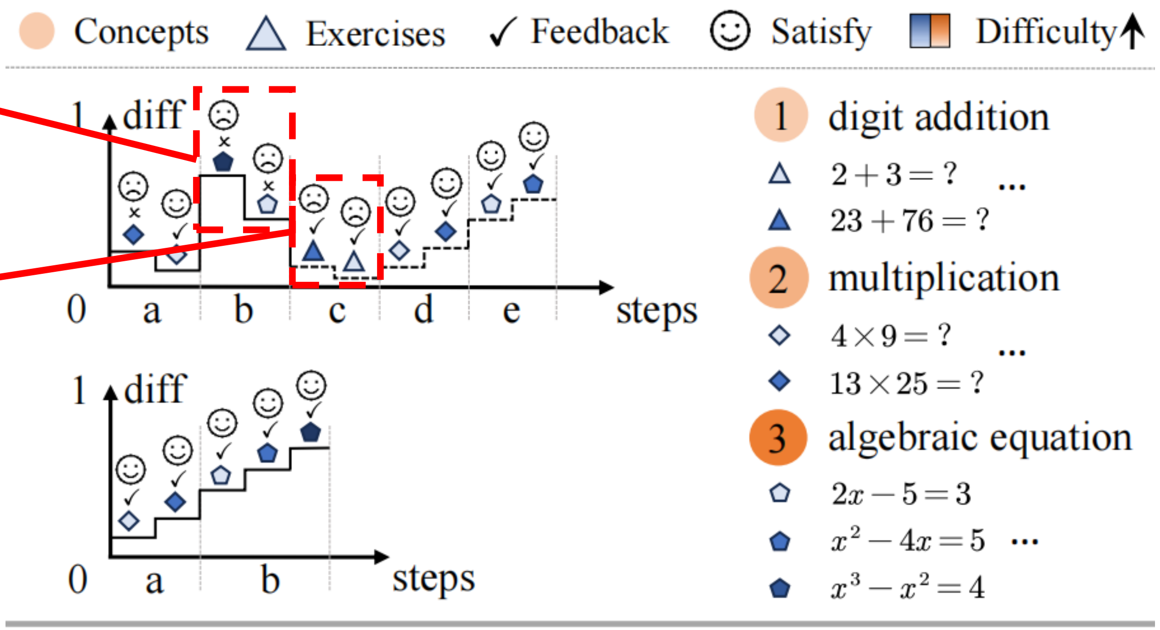
Background

➢ First, the path may be **rough**: When learners tread the path without considering item difficulty, it's akin to walking a dark, uneven road, making learning harder and dampening interest.

Recommend item $x^3 - x^2 = 4$ **before item** $2x - 5 = 3$ is correctly solved

Recommend items similar to $2+3=?$ **after item** $23+76=?$ has been solved successfully

Lead to a decrease in learners' learning interest and satisfaction



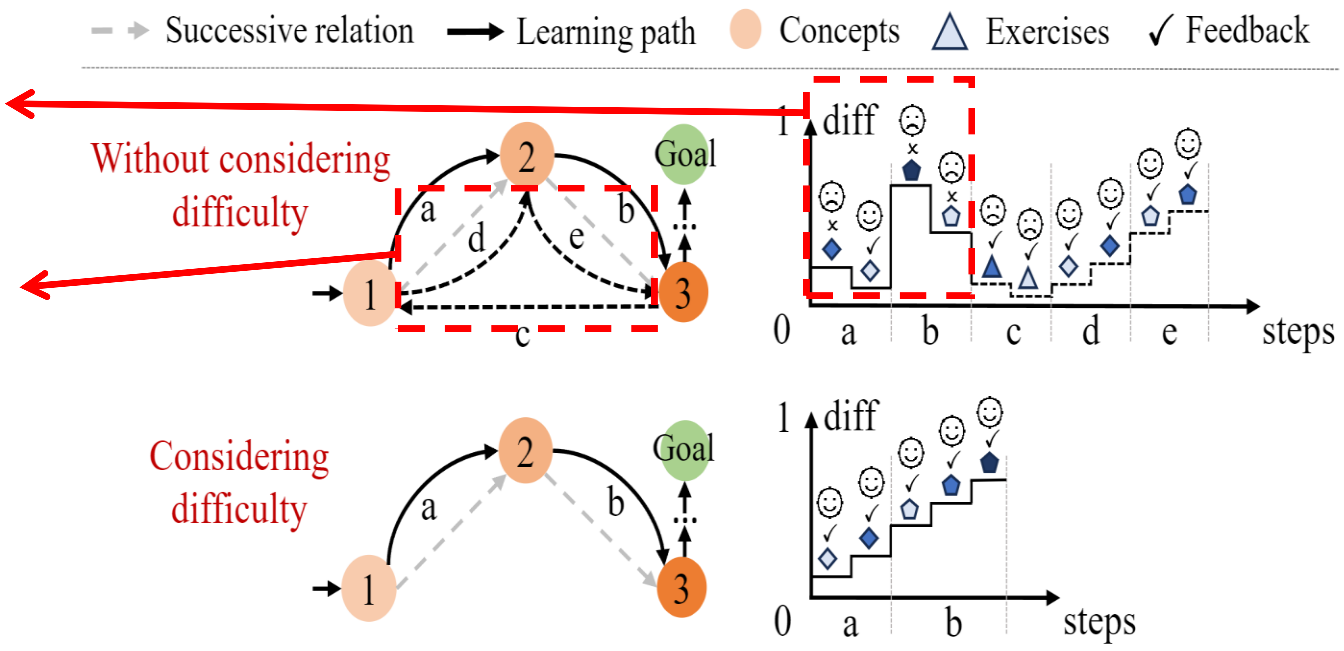
Background

- Second, the path may be **inefficient**: Treating different difficulty levels of items in the learning path with equal effort is akin to taking the same number of steps regardless of the distance of the journey.

Allowing learners only **a few attempts** on very challenging items before switching.

Persisting with a difficult item despite **numerous attempts** without mastery.

Result in inefficiencies in the learning journey.



CONTENTS

Problem Statement

- We focus on the issue of **step-by-step recommendations** for session-based learning paths based on real-time interactions.
  - **Given**
    - historical learning records **H**
    - learning goals **G**
    - item set **LI, PI**
  - **Output**
    - learning path **P**

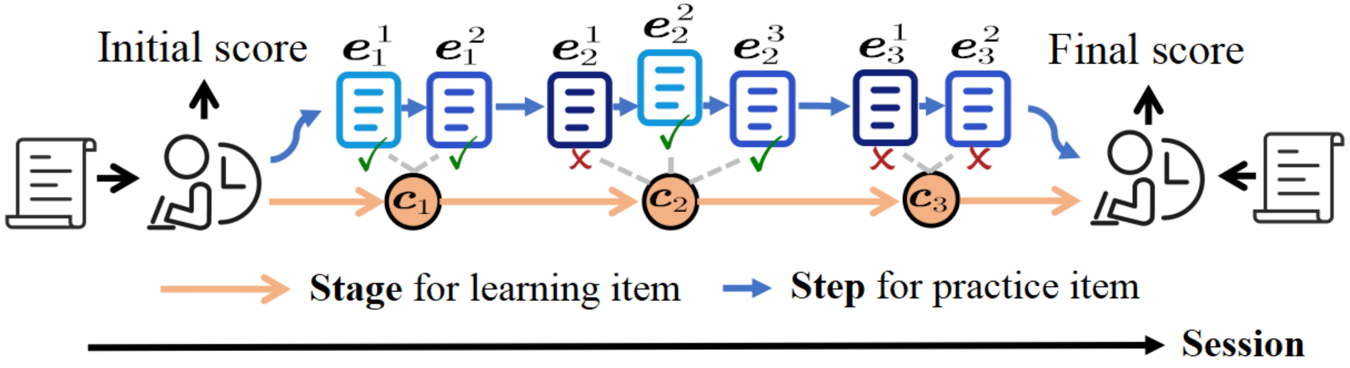


Figure 3: Illustration of Learning Process in One Session

$$E_p = \frac{E_e - E_s}{E_{sup} - E_s}$$

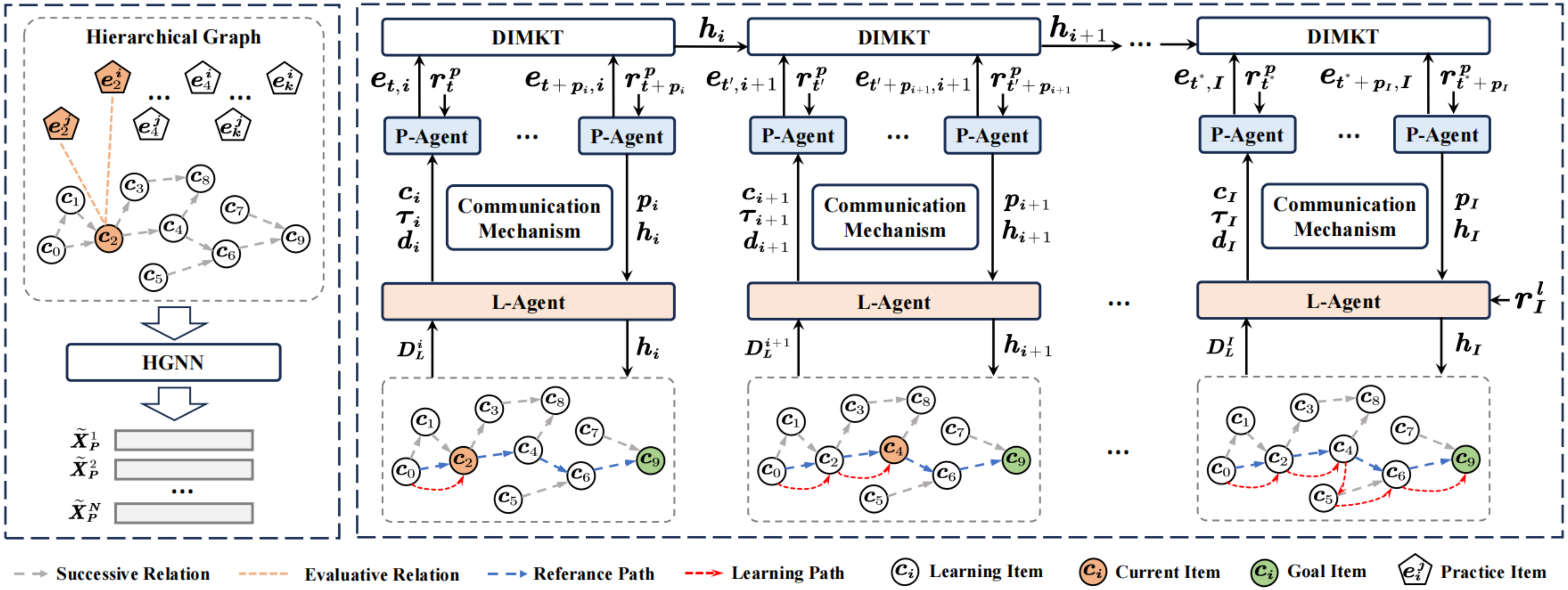
E_e : final score

E_s : initial score

E_{sup} : full score

Our Method

- To conquer the above limitations, we propose a novel method named **Difficulty-constrained Learning Path Recommendation (DLPR)**, which is aware of item difficulty.

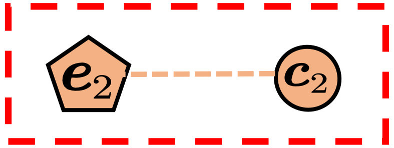


Our Method

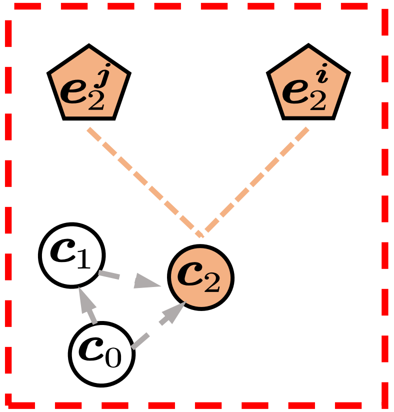
- Hierarchical Graph Enhenced Item Representation

Graph Construction

item



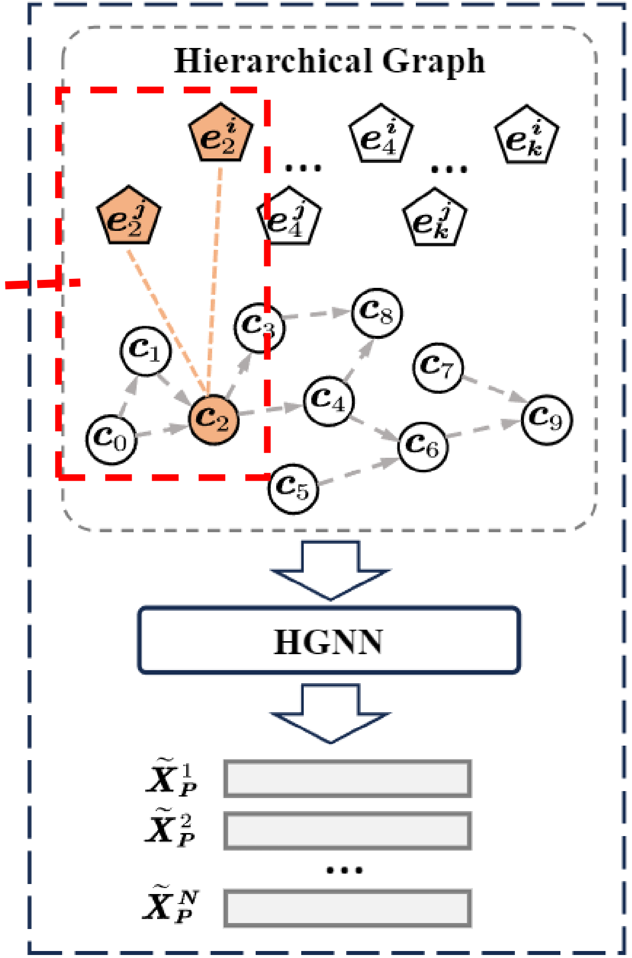
Previous Methods



practice item

learning item

Our Method



➢ Hierarchical Graph Enhenced Item Representation

Graph Representation

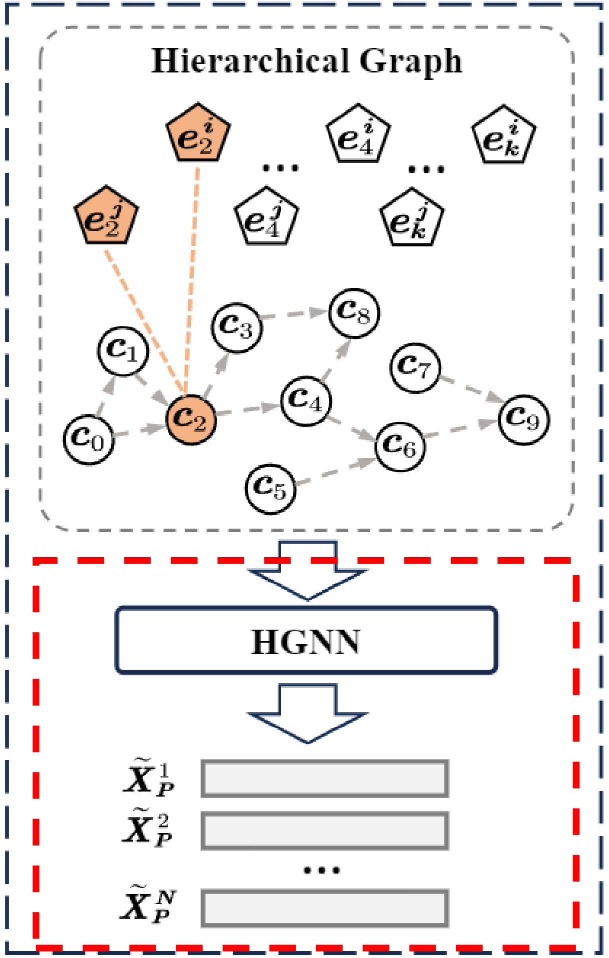
Difficulty calculation

Following some previous works, we calculate the difficulty based on statistical information as follows:

$$DP_j = \frac{\sum_{j=1}^{|S_j|} a_j == 0}{|S_j|} \times \lambda_P$$

The more learners get it wrong, the more difficult it is !

where S_j is the number of learners who answer the practice items e_j. $a_j == 0$ indicates learners that answered incorrectly and λ_P represents the predefined level of item difficulty.



Our Method

➢ Hierarchical Graph Enhenced Item Representation

Graph Representation

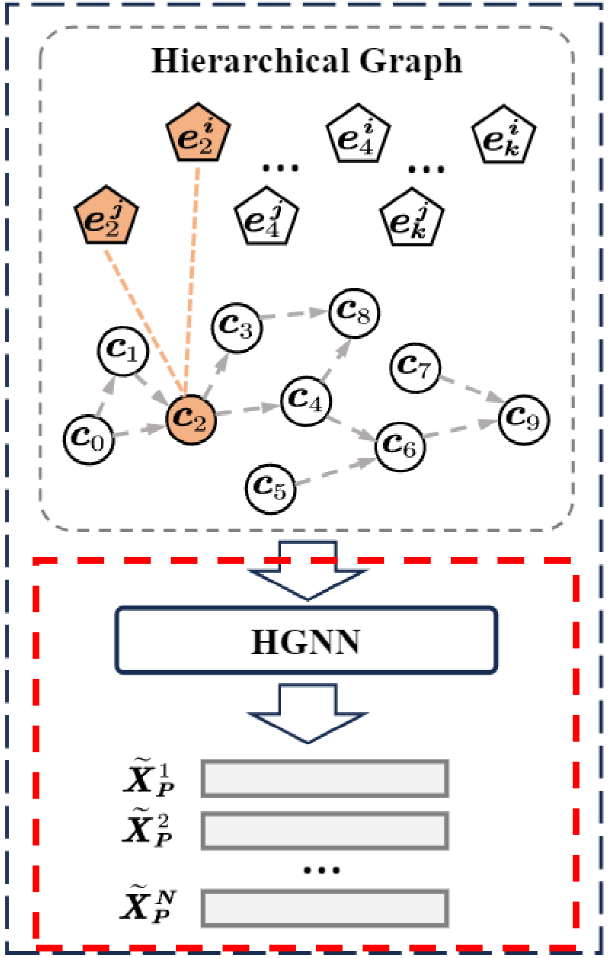
Item representation

We represent all of the items and their difficulty with embeddings.

items **difficulty** **simplified mean aggregation**

$$\mathbf{X}_L^i = \mathbf{W}_L^T[\mathbf{E}_l^i \oplus \mathbf{E}_{dl}^i] + \mathbf{b}_L,$$
$$\mathbf{X}_P^j = \mathbf{W}_P^T[\mathbf{E}_p^j \oplus \mathbf{E}_{dp}^j] + \mathbf{b}_P,$$

$$\tilde{\mathbf{X}}_P^j = \mathbf{X}_P^j \oplus \frac{1}{|\mathcal{N}_j|}\sum_{i \in \mathcal{N}_j} \mathbf{X}_L^i$$



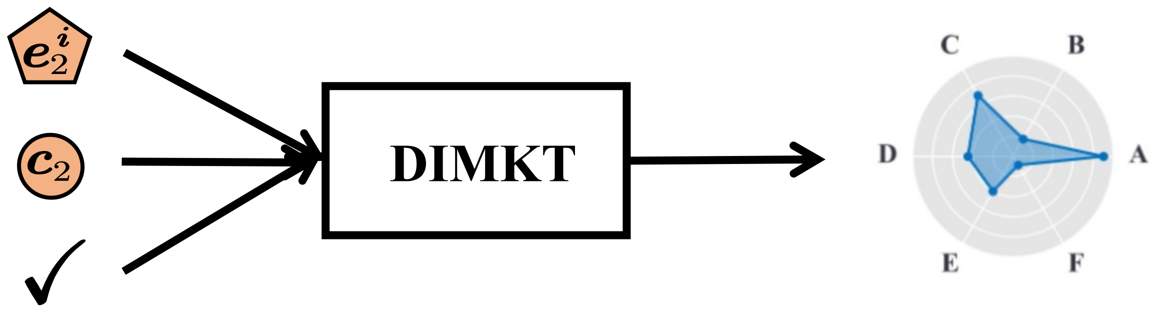
Our Method

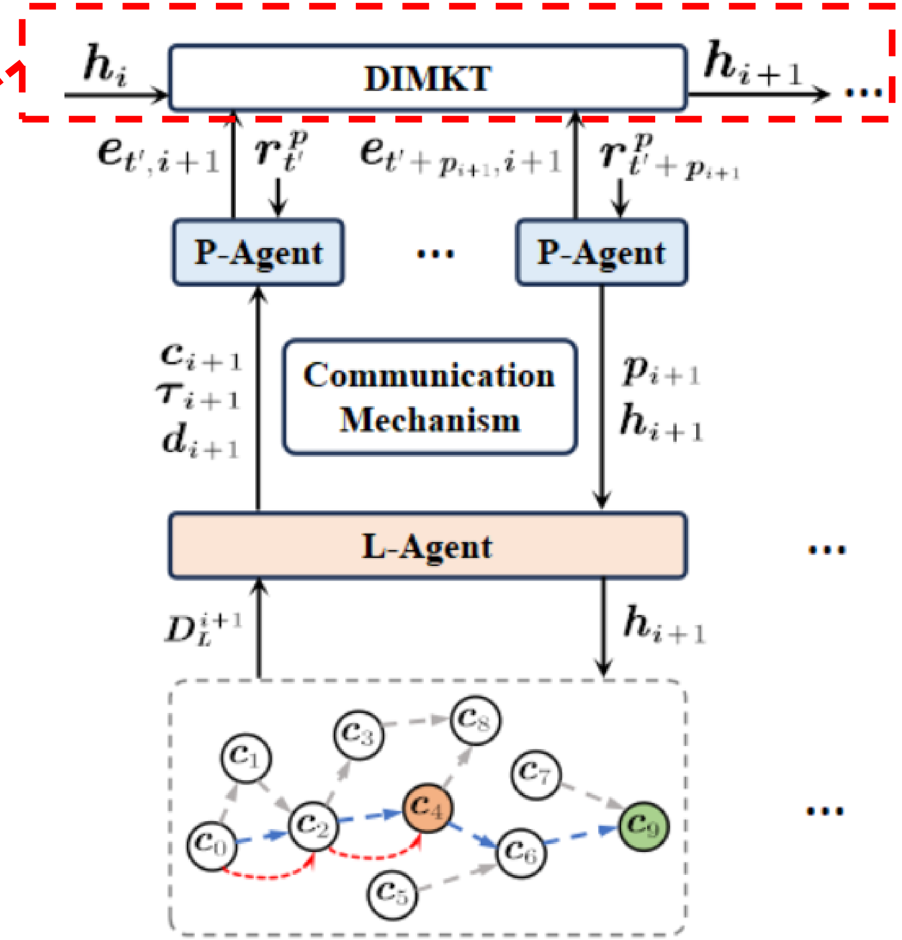
➢ Difficulty-driven Hierarchical Reinforement Learning

Knowledge State Estimation

DIfficulty Matching Knowledge Tracing (DIMKT) [2]

Comprehending learners' knowledge state is a prerequisite for recommending suitable learning resources.





[2] Shen, Shuanghong, et al. "Assessing student's dynamic knowledge state by exploring the question difficulty effect." Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. 2022.

Our Method

- Difficulty-driven Hierarchical Reinforement Learning

L-Agent

- **State** $s_i^l = h_{i-1} \oplus \boxed{\mathcal{G}}$ **learning goals**
- **Action Space**
- **Policy**

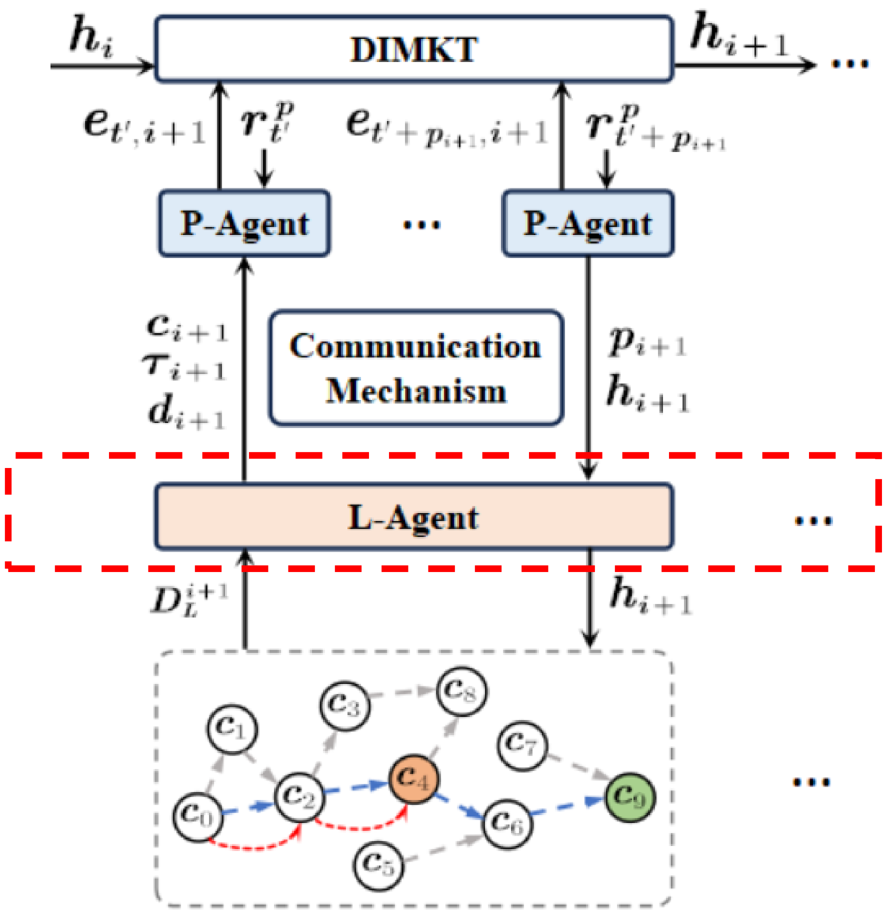
$$c_i = Softmax(FC(s_i^l)),$$

$$\mathcal{V}^L(s_i^l; \phi_L) = FC(s_i^l),$$

promotion

- **Reward**

$$r_i^l = \begin{cases} \boxed{E_p}, & \text{if } i \text{ is the last learning stage} \\ 0, & \text{otherwise,} \end{cases}$$



Our Method

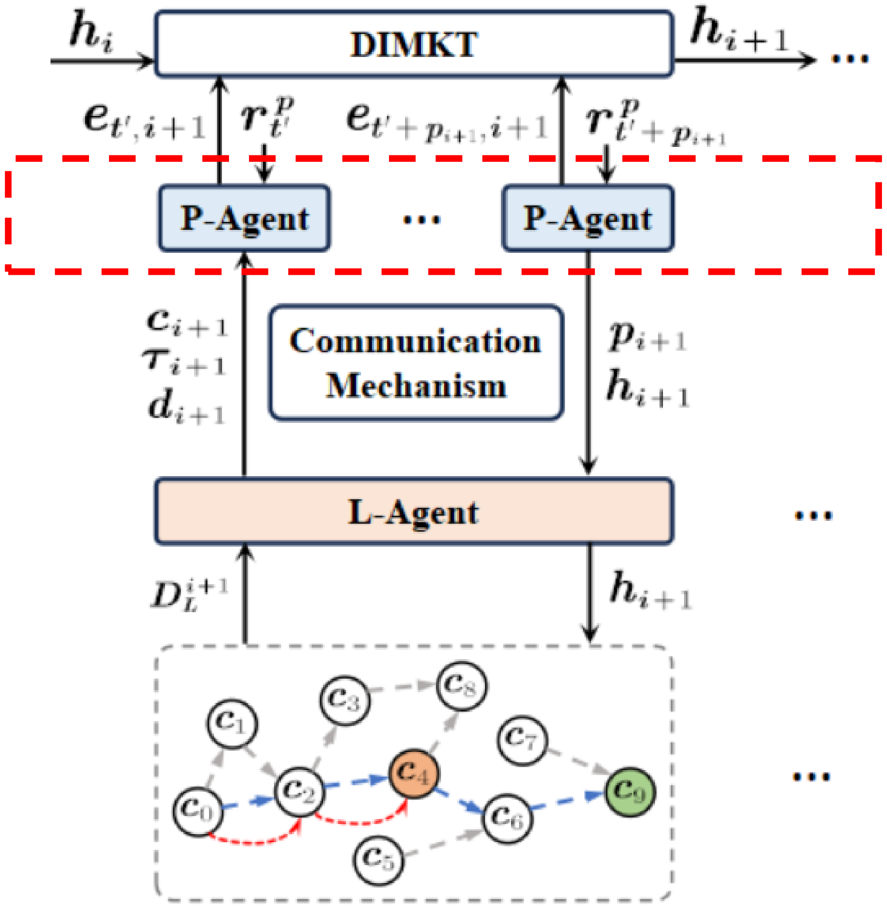
- Difficulty-driven Hierarchical Reinforement Learning

P-Agent

- **State** $s_{t,i}^{p} = h_t \oplus \boxed{c_i}.$ **learning item**
- **Action Space** $D_P^i = \Psi(c_i) \in \mathcal{PI}$
- **Policy** **difficulty control**
- **Reward** $\boxed{r_t^{p1} = \mathcal{L}(DP_t, DP_{t-1}) = -(DP_t - DP_{t-1})^2}$

$$r_t^{p2} = \begin{cases} h_i - h_{i-1}, & \text{if } h_i > There \\ 0, & otherwise, \end{cases}$$

$$r_t^p = \alpha_1 \times r_t^{p1} + \alpha_2 \times r_t^{p2}, \alpha_1, \alpha_2 \in [0, 1]$$



Our Method

- Difficulty-driven Hierarchical Reinforement Learning

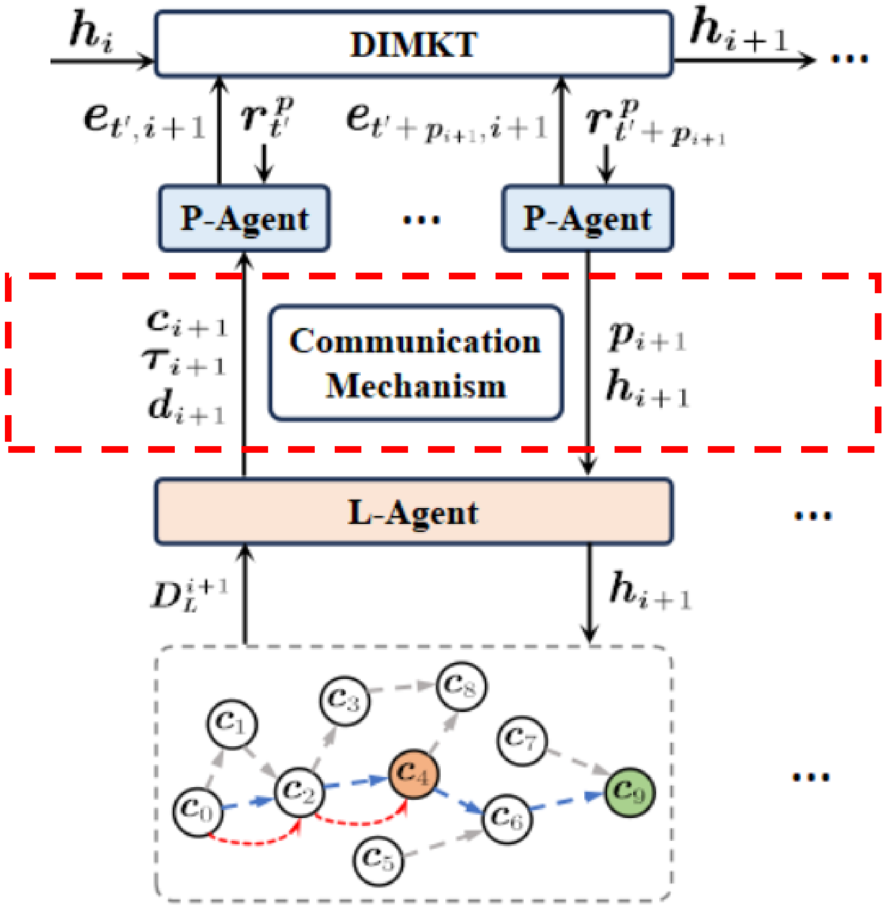
Communication Mechanism

- **Initial difficulty control**

$$d_i = \dot{h}_i + ln(\frac{Prob_i}{1 - Prob_i}),$$

- **Practice tolerance control**

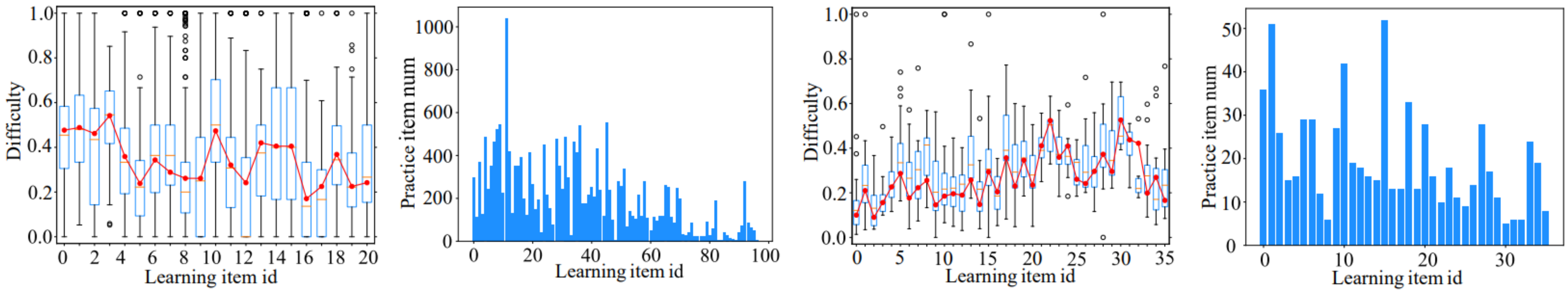
$$\tau_i = f(p_{i-1}, h_{i-1}, \tau_{i-1}, DL_i),$$



CONTENTS

Experiments

- Datasets



(a) Item-diff distribution in ASSIST09 (b) Practice items in ASSIST09 (c) Item-diff distribution in Junyi (d) Practice items in Junyi

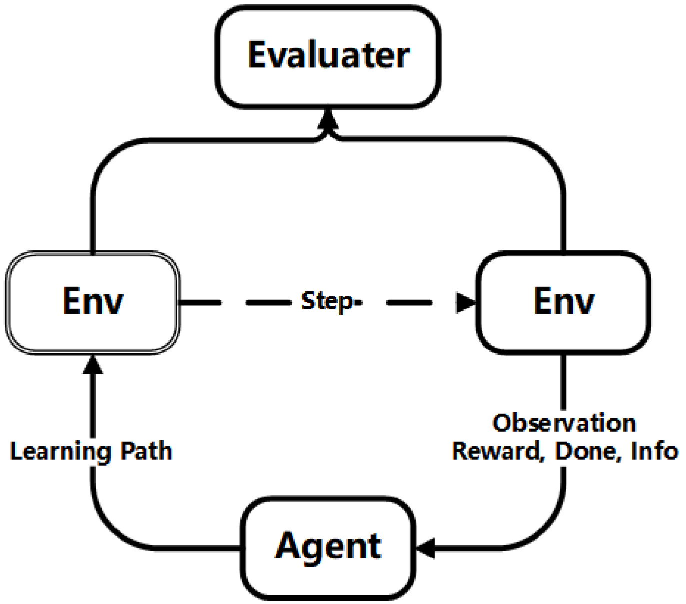
- **Junyi:** use the **prerequisite graph** of items provided
- **Assist09:** construct a **transition graph** as an estimation

Dataset	Junyi	ASSIST09
learning items	36	97
practice items	711	16,836
learners	245,511	4,092
records	25,367,573	397,235
number of edges in HG	267	683

- Simulators
  - For evaluation, a key issue is that existing realistic data **only contains static information**. This data cannot directly analyze if practice items not in a sequence can be answered correctly.

KSS: Knowledge Structure based Simulator — **Rule-based**

KES: Knowledge Evolution based Simulator — **Data-based**
i.e. KES-Junyi & KES-ASSIST



More infomation about simulators can be found in [1]

[1] Liu, Qi, et al. "Exploiting cognitive structure for adaptive learning." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.

Experiments

- Baseline Approaches
  - **Rec-based: KNN** (Cover et al., 1967)**, GRU4Rec** (Hidasi et al., 2015)
  - **RL-based: DQN** (Chen et al., 2018)**, Actor-Critic** (Konda et al., 1999)**,**
    CB (Intayoad et al., 2020)**, RLTutor** (Kubotani et al., 2021)
  - **RL&Cog-based: CSEAL** (Liu et al., 2019)**, GEHRL** (Li et al., 2023)

- Following previous works, we evaluate these methods based on the **promotion** given by simulators.

$$E_p = \frac{E_e - E_s}{E_{sup} - E_s}$$

Experiments

➢ Overall Performance Comparison

Table 1: Performance comparison for learning path recommendation methods. Existing state-of-the-art results are underlined and the best results are bold. Our DLPR is compared with the SOTA GEHRL and * indicates a p-value < 0.05 in the t-test.

		KNN	GRU4Rec	DQN	Actor-Critic	CB	RLTutor	CSEAL	GEHRL	**DLPR**
KSS	step=5	0.1005	0.1124	0.1559	0.1437	0.0852	0.1999	0.2095	<u>0.2321</u>	**0.5583***
	step=10	0.3133	0.2767	0.2836	0.4072	0.2643	0.4008	0.4233	<u>0.5644</u>	**0.7294***
	step=20	0.2972	0.1998	0.3236	0.4931	0.2614	0.5303	0.5716	<u>0.7426</u>	**0.8305***
KES-Junyi	step=5	-0.0902	-0.0047	0.0299	0.1004	0.0666	-0.0007	0.0975	<u>0.1198</u>	**0.2049***
	step=10	-0.1455	-0.0721	-0.1058	0.1671	0.1451	-0.0379	0.2021	<u>0.2278</u>	**0.3835***
	step=20	0.1343	0.0993	0.1536	0.1916	0.2098	-0.1034	0.2505	<u>0.4206</u>	**0.6124***
KES-ASSIST09	step=5	-0.0549	-0.0536	-0.0495	-0.0004	-0.0563	-0.0611	0.0482	<u>0.0751</u>	**0.0807***
	step=10	-0.0731	-0.1003	-0.0934	-0.0327	-0.1294	-0.1096	0.0637	<u>0.0918</u>	**0.1544***
	step=20	-0.0932	-0.1344	-0.0267	0.0676	0.0038	0.0784	0.1009	<u>0.1971</u>	**0.3283***

- Our proposed DLPR **outperforms all baselines** in all three simulators
- Reinforcement Learning methods exhibit superior performance due to **real-time interactive feedback**
- Fewer steps and higher complexity reduce performance, underscoring the need for **better recommendations**

Experiments

➢ Learning Path Efficiency and Smoothness

Table 2: Learning path efficiency and smoothness for learning path recommendation methods. It should be noted that "-" in the table indicates that the method cannot achieve absolute promotion and meet the learning goals.

efficiency

smoothness

		KNN	GRU4Rec	DQN	Actor-Critic	CB	RLTutor	CSEAL	GEHRL	**DLPR**
KSS	Learning-Steps	54	72	48	41	56	34	29	14	**8**
	Cog-Gap	0.3216	0.3889	0.3709	0.3921	0.3514	0.3336	0.3639	0.2436	**0.1017**
	Diff-MAD	0.3886	0.3267	0.3574	0.3749	0.3309	0.3696	0.3231	0.2704	**0.1200**
KES-Junyi	Learning-Steps	-	-	-	-	219	-	128	96	**33**
	Cog-Gap	-	-	-	-	0.3413	-	0.3898	0.3641	**0.1313**
	Diff-MAD	-	-	-	-	0.3687	-	0.2791	0.3144	**0.1614**
KES-ASSIST09	Learning-Steps	-	-	-	-	-	536	378	185	**69**
	Cog-Gap	-	-	-	-	-	0.4197	0.3823	0.3562	**0.1391**
	Diff-MAD	-	-	-	-	-	0.2947	0.2893	0.2998	**0.1577**

➢ The **Learning-Steps** metric captures the average number of steps required to achieve learning goals.

$$\text{Cog-Gap} = \frac{\sum_{j=1}^{n} |h_j - DP_j|}{n}, \qquad \text{Diff-MAD} = \frac{\sum_{j=1}^{n-1} |DP_{j+1} - DP_j|}{n-1},$$

Experiments

➢ Ablation Study

- DLPR w/o ACS which refers to DLPR without adaptive learning action space in section 5.2.2.
- DLPR w/o Init-diff which eliminates the guidance of initial difficulty control in section 5.4.1.
- DLPR w/o Torlerance which excludes the practice tolerance control in section 5.4.2.
- DLPR w/o Diff-reward which removes the reward r_t^{p1} that controls difficulty variations in section 5.3.4.

Table 3: Results of ablation experiments.

	Learning Steps	Cog-Gap	Diff-MAD
w/o ACS	154	0.1707	0.2099
w/o Init-diff	86	0.2133	0.1796
w/o Torlerance	94	0.2654	0.1832
w/o Diff-reward	61	0.3516	0.3235
DLPR	**33**	**0.1313**	**0.1614**

➢ The **complete model achieved the best** overall performance

➢ **ACS** had the most significant impact on the number of learning steps

➢ **Diff-reward** had the greatest influence on cognitive differences and difficulty smoothness

➢ **Init-diff** and **Tolerance** played varying roles in all three factors

Experiments

➢ Impact of Difficulty Level Segmentation

$$DP_j = \frac{\sum_{j=1}^{|S_j|} a_j == 0}{|S_j|} \boxed{\times \lambda_P}$$

predefined difficulty level

Similar to the common practice of categorizing difficulty levels into three tiers: easy, medium, and hard, here we aim for a more granular analysis of difficulty by classifying it into 10, 30, 50, 100, and 500 levels.

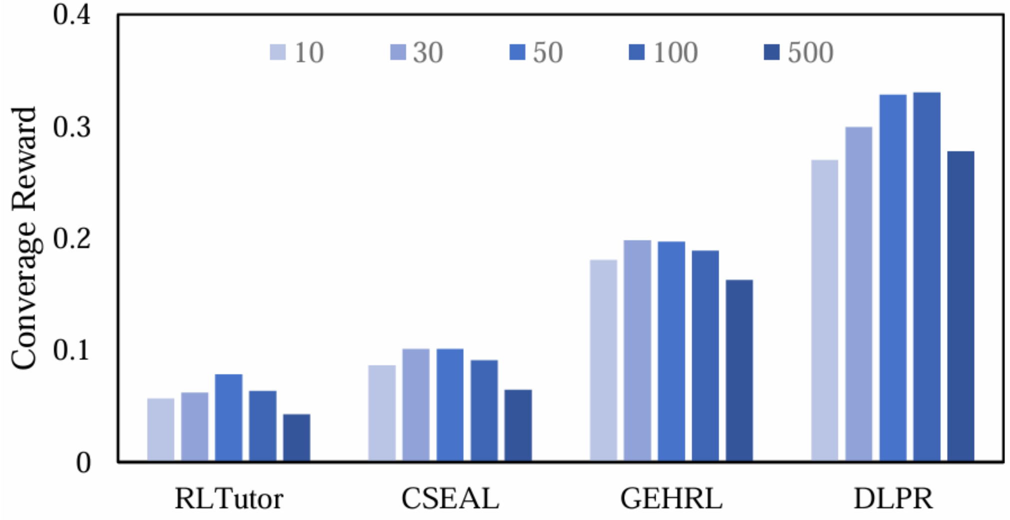


Figure 5: The performance of some methods under different predefined difficulty levels.

Properly defining difficulty level segmentation is important !

Experiments

➢ Case Study

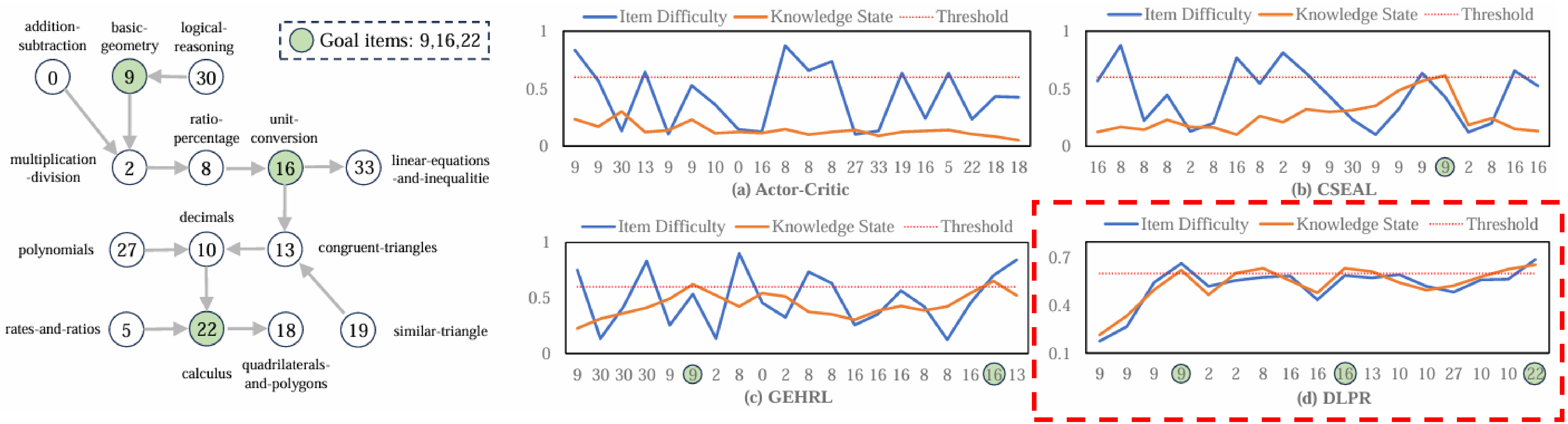


Figure 4: Visualization of different learning paths recommended by four selected methods for the same learning goal items.

➢ Our method, considering both **knowledge structure and item difficulty**, efficiently recommends paths with smoothness and achieves all goals.

CONTENTS

Conclusion

- we addressed **two special issues "rough" and "in efficient"** of learning paths from a real walking perspective and proposed an effective and smooth learning path recommendation method considering item difficulty
- we **constructed a hierarchical graph** of learning and practice items to capture their difficulty and higher-order correlations.
- we **designed a Difficulty driven Hierarchical Reinforcement Learning framework** to generate learning paths smoothly and efficiently through two agents' collaboration.
- Extensive **experimental results validate the superiority of our framework** in providing highly satisfactory learning path recommendations by thoroughly considering item difficulty.

- Never theless, as we primarily conducted the experiments in the simulated environments, further research will **develop the system and test the model in practical settings in the real-world environments** to investigate broader impacts.

Thank you

Q&A

sosweetzhang

sosweetzhang@mail.ustc.edu.cn

https://sosweetzhang.github.io/

bigdata.ustc.edu.cn