LIGER: Learner-Intent-Driven Generative Exercise Recommendation

Anonymous ACL submission

Abstract

Exercise recommendation is central to adaptive learning, but existing methods are usually optimized for system-defined objectives and rarely use learner intent as an explicit control signal. As a result, learners with similar histories and mastery states may receive similar exercises even when their short-term goals differ. We study learner-intent-driven exercise recommendation, where natural-language intents guide the generation of exercises that both match learner goals and improve mastery. We propose LIGER, a generative framework that injects learner intent into both supervised training and reinforcement learning. To address the pedagogical opacity of raw exercise IDs, LIGER first represents exercises with structured identifiers composed of knowledge concept, difficulty, and concrete instance tokens. Built on this EduSID space, LIGER then automatically mines latent intent signals from learners' interaction histories, turning ordinary educational logs into intent-conditioned supervision without requiring explicit intent annotations. The resulting signals are used to train an intent-conditioned generative policy with structured supervised objectives, which is further refined by a knowledge-tracing simulator to optimize mastery improvement on intent-related knowledge. Experiments on three educational datasets show that LIGER consistently improves intent-related expected progress over sequential, reinforcement-learning, and LLM-agent baselines. Ablation and counterfactual-intent analyses further verify the contribution of EduSID representations, latent-intent supervision, and KT-guided optimization. [1]

1 Introduction

Exercise recommendation is a central component of adaptive learning systems. Existing methods use sequential recommendation, knowledge tracing, and reinforcement learning to exploit learner history, response feedback, and mastery estimates (Kang and McAuley, 2018; Sun et al., 2019; Corbett and Anderson, 1994; Piech et al., 2015; Cai et al., 2019; Liu et al., 2019; Li et al., 2023). These methods can model learning state and multi-step practice, but their objectives are still largely specified by the system, such as next-exercise prediction, correctness prediction, or long-term reward maximization, rather than by learner-expressed goals in natural language.

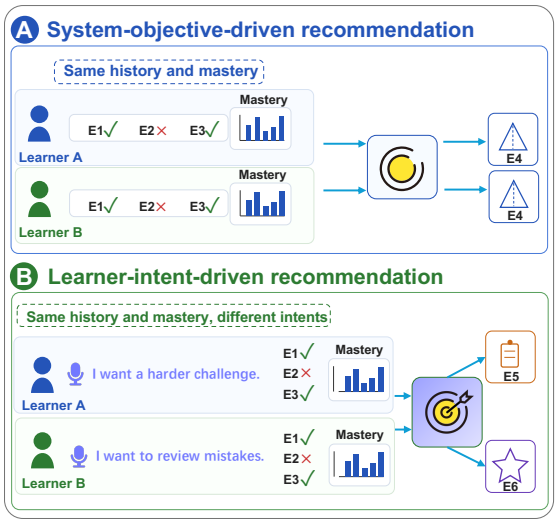

Figure 1: Motivation for learner-intent-driven exercise recommendation. Similar learner states can require different exercises when short-term intents differ.

However, learner state alone is insufficient to determine the next pedagogically appropriate exercise. Two learners may share the same response history and mastery estimate, yet one may want to repair recent mistakes while the other wants a harder challenge. These goals imply different recommendation policies, but are invisible to conventional system-objective-driven recommenders. Recent language-based educational recommenders begin to accept natural-language practice goals. For

[1] Anonymous code repository: https://anonymous.4open.science/r/LIGER_EMNLP

example, GenAL (Lv et al., 2025) supports the retrieval of exercise from the learners goals, but mainly retrieves exercises from a fixed catalog. In contrast, we target a state-aware, multi-step, generative recommender whose behavior is controlled by learner intent. Figure 1 illustrates the core motivation: learners with similar observable states may require different exercise paths when their intents differ.

This setting introduces three technical challenges. First, ordinary exercise logs rarely contain explicit learner intents, making direct intent-supervised training infeasible. Second, raw exercise identifiers are pedagogically opaque: they do not expose the target knowledge path, difficulty level, or alternative exercises that satisfy the same goal. Third, standard next-item imitation is misaligned with learner-intent-driven recommendation, because the logged next exercise is often only one valid option and may not best serve intent-related mastery. While prior generative recommendation work has shown the value of language-modeling formulations and semantic identifiers (Geng et al., 2022; Rajput et al., 2023; Lu et al., 2024; Liang et al., 2026), exercise recommendation requires identifiers and objectives that encode pedagogical decisions rather than only item identity or retrieval semantics.

We propose LIGER, a generative framework for learner-intent-driven exercise recommendation. LIGER first represents each exercise with EduSID, a knowledge–difficulty–instance identifier that exposes the target concept, challenge level, and concrete instance, while enabling constrained decoding over legal exercises. Built on this structured action space, we further introduce Future-Hindsight construction, a latent-intent mining strategy for recovering proxy intent supervision from learner interaction histories. For each training prefix, we map its future window into EduSID sequences, use the knowledge–difficulty prefix of the earliest future EduSID as the target pedagogical prefix, and collect same-prefix future exercises as multi-positive targets. We then combine this EduSID-derived target with prefix-side evidence, including recent errors, KT-estimated mastery, and relative difficulty, to infer the learner's short-term pedagogical need and render it as a proxy natural-language intent.

Our contributions are summarized as follows:

- We formulate learner-intent-driven exercise recommendation as a generative task conditioned on learner history, answer feedback, mastery state, and natural-language intent, enabling recommendation policies to respond to learner-expressed short-term goals.
- We introduce EduSID, a structured knowledge–difficulty–instance action space, and propose Future-Hindsight construction to mine latent intent supervision from ordinary interaction logs by deriving proxy intents, target prefixes, and multi-positive targets in this space.
- We develop LIGER, an intent-conditioned generative framework trained with structured SFT objectives and KT-guided rewards, and validate its effectiveness through main comparisons, ablations, and counterfactual-intent analyses on three educational datasets.

2 Related Work

2.1 Exercise Recommendation

Exercise recommendation selects subsequent exercises from learner histories, response feedback, and estimated learner states. Prior work includes knowledge-aware learning-path methods (Tarus et al., 2018; Nabizadeh et al., 2020), neural sequential recommenders (Hidasi et al., 2016; Kang and McAuley, 2018; Sun et al., 2019), and education-specific methods that use knowledge tracing, concepts, or mastery estimates for selection (Corbett and Anderson, 1994; Piech et al., 2015; Zhang et al., 2017; Pandey and Karypis, 2019; Ghosh et al., 2020; Ai et al., 2019).

A related line formulates recommendation as sequential decision making, using learner states, knowledge structures, or KT-estimated gains to optimize policies (Cai et al., 2019; Chen et al., 2023; Kubotani et al., 2021; Intayoad et al., 2020; Liu et al., 2019; Li et al., 2023; Ozyurt et al., 2025). Recent LLM-based educational systems further use natural language and external knowledge for retrieval, tutoring, simulation, and agent planning (Xu et al., 2024a; Li et al., 2025; Anonymous, 2026; Xiong et al., 2025; Xu et al., 2024b; Lv et al., 2025). In contrast, our work uses natural-language learner intent as an explicit control signal for generating structured exercise identifiers.

2.2 Generative Recommendation

Generative recommendation reformulates recommendation as token generation rather than candidate scoring. P5 (Geng et al., 2022) unifies rec-

ommendation as text-to-text language modeling, while instruction-tuned and controllable LLM recommenders adapt language models to recommendation objectives and controllable outputs (Zhang et al., 2023; Bao et al., 2023; Lu et al., 2024; Wang et al., 2024b). Other work injects text-derived item representations, interaction graphs, generated descriptions, feedback signals, or knowledge constraints into generative models (Zhang et al., 2024; Wang et al., 2024a; Han et al., 2025; Zhang et al., 2025; Mohbat and Zaki, 2025).

These studies motivate our generative formulation, but exercise recommendation requires pedagogically meaningful generation over knowledge, difficulty, and concrete exercises; EduSID therefore structures each exercise into knowledge–difficulty–instance tokens.

3 Task Formulation

Let $Q = \{q_1, q_2, \ldots, q_N\}$ be the exercise set and $\mathcal{K} = \{k_1, k_2, \ldots, k_M\}$ the set of knowledge paths. For learner u, the exercise and answer histories at time step t are:

$$H_t^u = [q_1, q_2, \ldots, q_t], \qquad A_t^u = [a_1, a_2, \ldots, a_t], \tag{1}$$

where $a_i \in \{0, 1\}$ indicates whether the learner answered exercise q_i correctly. A knowledge tracing model estimates the learner's current mastery state:

$$M_t^u = KT(H_t^u, A_t^u). \tag{2}$$

We follow the standard simulator-based protocol used in adaptive learning and learning-path recommendation (Liu et al., 2019; Li et al., 2023; Lv et al., 2025). A DKT-based KT simulator (Piech et al., 2015) updates learner states after each recommended exercise and provides offline proxies for response probability and mastery evolution.

The learner also provides a natural language intent ν_t, such as "I want to consolidate the function graph problems that I got wrong." Let $\mathcal{C}(\nu_t) \subseteq \mathcal{K}$ be the intent-related knowledge set, and let $\rho_t = \langle K, D \rangle$ be its target concept–difficulty prefix.

The recommendation policy π generates the next exercise:

$$q_{t+1} \sim \pi(H_t^u, A_t^u, \nu_t, M_t^u). \tag{3}$$

After L steps, the system produces:

$$P = \{q_{t+1}, q_{t+2}, \ldots, q_{t+L}\}. \tag{4}$$

The objective is to improve intent-related mastery rather than imitate a single logged next exercise. Let $M_\ell(\nu_t)$ be the average mastery over $\mathcal{C}(\nu_t)$ after ℓ rollout steps:

$$M_\ell(\nu_t) = \frac{1}{|\mathcal{C}(\nu_t)|} \sum_{c \in \mathcal{C}(\nu_t)} m_\ell(c). \tag{5}$$

We define Final Intent-Ep, where Ep denotes expected progress under the offline KT simulator:

$$\text{Final Intent-}E_p = \frac{M_L(\nu_t) - M_0(\nu_t)}{\max(1 - M_0(\nu_t), 10^{-6})}, \tag{6}$$

where mastery is normalized by its default upper bound of 1. A larger Final Intent-Ep indicates that the recommended path better improves the learner's mastery of the intended knowledge.

4 Method

LIGER maps learner history, answers, natural-language intent, and KT-estimated mastery to a structured EduSID action. The method has two stages. First, exercises are mapped into a knowledge–difficulty–instance identifier space, and ordinary logs are converted into proxy intents and multi-positive targets through Future-Hindsight construction (Sections 4.1–4.2). Second, these signals train an intent-conditioned policy by SFT and are further refined with KT-guided rewards and constrained decoding (Sections 4.3–4.4). Figure 4 shows the overall pipeline.

4.1 EduSID as a Structured Pedagogical Action Space

Exercise recommendation should decide *what to practice*, *at what difficulty*, and *which concrete exercise to assign*. Raw item IDs do not expose these decisions. As shown in Figure 2, EduSID decomposes each exercise into three tokens: a knowledge-path token, a difficulty token, and a local instance token.

Formally, each exercise $q \in \mathcal{Q}$ is mapped to

$$\phi(q) = \langle K \rangle \langle D \rangle \langle I \rangle, \tag{7}$$

where $\langle K \rangle$ encodes the concept path, $\langle D \rangle$ encodes a difficulty bucket, and $\langle I \rangle$ distinguishes exercises within the same concept–difficulty bucket. When metadata difficulty is unavailable, we estimate the difficulty score from train-split correctness:

$$\hat{d}(q) = 1 - \frac{1}{|\Omega_q^{\text{tr}}|} \sum_{(u,t) \in \Omega_q^{\text{tr}}} a_t, \qquad \Omega_q^{\text{tr}} = \{(u, t) : q_t = q\}. \tag{8}$$

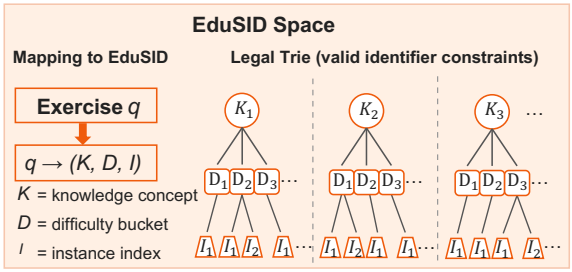

Figure 2: EduSID structure. Each exercise identifier contains a knowledge-concept token, a difficulty token, and a local instance token.

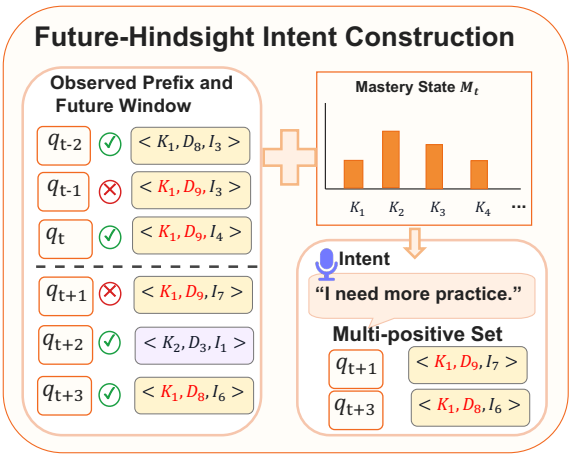

Figure 3: Future-Hindsight data construction. A future item defines the target prefix, same-prefix future items provide positives, and recent responses together with KT-estimated mastery determine the intent type.

The prefix $\langle K, D\rangle$ is the pedagogical target used for intent conditioning and reward computation, while the full $\langle K, D, I\rangle$ sequence denotes an executable exercise. We constrain decoding to the legal set $\mathcal{S}_{\text{legal}} = \{\phi(q) : q \in \mathcal{Q}\}$ with a trie, so every generated identifier maps back to a real exercise.

4.2 Future-Hindsight Intent Construction

Educational logs do not provide explicit learner intents ν_t, so the natural-language intent used by LIGER must be instantiated before intent-conditioned training is possible. As summarized in the abstract, we use four proxy intent types: *error consolidation*, *weakness remediation*, *difficulty challenge*, and *regular practice*. Figure 3 gives the intuition: a same-split future window anchors *what* the learner is about to practice, while prefix-side responses and mastery estimates determine *why* this practice is expressed as one of the four intents.

For split ℓ and prefix length t, we first define the same-split future window:

$$F_t^\ell = \{q_{t+1}, \ldots, q_{t+w}\} \subset \tau_u^\ell. \tag{9}$$

The earliest future exercise is used as the canonical structured target:

$$s_t^\star = \phi(q_{t+1}), \qquad \rho_t = \rho(s_t^\star) = \langle K, D\rangle. \tag{10}$$

Here, ρ_t is the target concept–difficulty prefix induced by hindsight. Because more than one exercise can satisfy the same pedagogical target, all future exercises with this prefix are treated as alternative positives:

$$\mathcal{P}_t = \{\phi(q) : q \in F_t^\ell,\ \rho(\phi(q)) = \rho_t\}. \tag{11}$$

The proxy intent is then rendered from the anchored prefix and the prefix-side learner state:

$$\nu_t = g(F_t^\ell, A_t, \rho_t, M_t). \tag{12}$$

In this renderer, the future window provides the target prefix and candidate positives, but the intent type is selected using learner-state evidence available before recommendation: recent errors support *error consolidation*, low KT-estimated mastery supports *weakness remediation*, a higher-difficulty target for a sufficiently prepared learner supports *difficulty challenge*, and the remaining cases become *regular practice*. The rendered sentence therefore gives the model an explicit natural-language control signal, while ρ_t and $\mathcal{P}_t$ keep the supervision tied to structured EduSID targets.

The resulting Future-Hindsight dataset is

$$\mathcal{D}_{\text{FH}}^\ell = \{(x_t, s_t^\star, \rho_t, \mathcal{P}_t) : x_t = (H_t, A_t, \nu_t, M_t)\}. \tag{13}$$

Thus, x_t contains only the prefix-side learner state and the constructed intent; ρ_t and $\mathcal{P}_t$ serve as supervision targets rather than additional inference-time inputs.

This construction should be distinguished from inference. During training, ν_t is a *synthetic proxy intent* constructed from same-split future windows. During real inference, the system accepts a learner-provided or externally supplied natural-language intent. During offline evaluation, held-out trajectories are used only to construct controlled intent queries and target sets. The model input never contains future exercises or future answers: future windows are used only after train/validation/test splits are fixed and only for supervision or evaluation-query construction. Future-Hindsight is therefore a data construction strategy, not access to future information at prediction time.

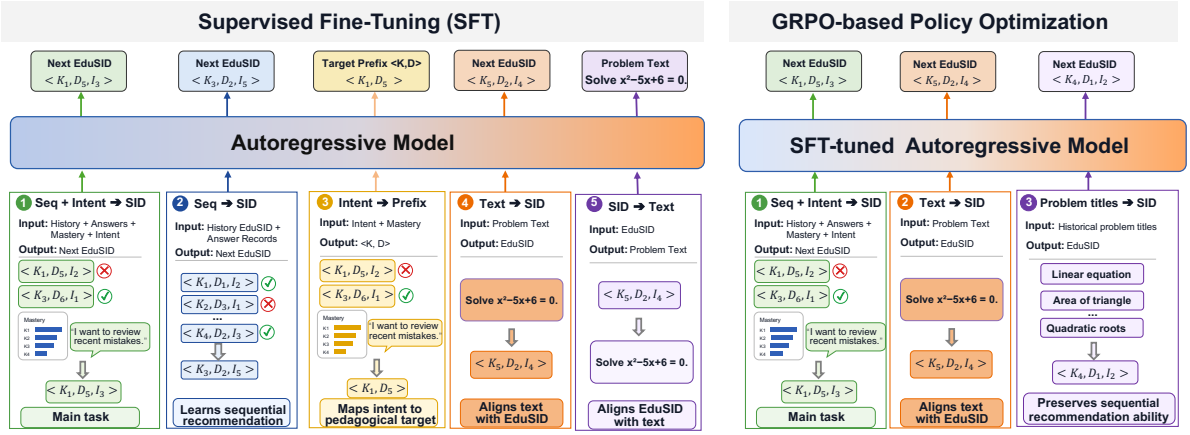

Figure 4: Overview of LIGER. LIGER trains an autoregressive model with SFT tasks and refines it through GRPO-based KT-guided optimization for target consistency, repetition control, and intent-related mastery improvement.

4.3 Learner-Intent-Conditioned Policy Learning

The SFT stage in Figure 4 teaches the model to generate legal EduSID actions from learner state and intent. For the main recommendation task, the prompt is serialized as

$$z_t = \text{Serialize}(H_t, A_t, \nu_t, M_t), \tag{14}$$

where H_t is the historical EduSID sequence, A_t is the answer sequence, ν_t is the learner intent, and M_t is the KT-estimated mastery state. The policy autoregressively generates $s = (s_1, s_2, s_3)$.

As summarized in Appendix A.1, SFT uses five supervised data types. The main *Seq + Intent → SID* records train learner-intent-driven recommendation. *Seq → SID* preserves basic sequential recommendation ability. *Intent → Prefix* aligns natural-language intent with the pedagogical target $\langle K, D\rangle$. The two text–SID tasks align exercise text with the structured output space.

The SFT objective combines the canonical next-EduSID target, prefix consistency, multi-positive matching, and auxiliary alignment losses:

$$\mathcal{L}_{\text{sft}} = \mathcal{L}_{\text{can}} + \lambda_\rho \mathcal{L}_{\text{prefix}} + \lambda_{\text{mp}} \mathcal{L}_{\text{mp}} + \mathcal{L}_{\text{aux}}. \tag{15}$$

$\mathcal{L}_{\text{can}}$ is the token-level loss on the canonical target $s_t^\star$; $\mathcal{L}_{\text{prefix}}$ encourages the generated identifier to share the target prefix ρ_t; $\mathcal{L}_{\text{mp}}$ assigns probability mass to all identifiers in the multi-positive set $\mathcal{P}_t$; and $\mathcal{L}_{\text{aux}}$ covers the remaining SFT alignment tasks listed in Appendix A.1.

4.4 KT-guided Reinforcement Learning and Inference

The RL stage in Figure 4 starts from the SFT policy and refines it with KT-guided offline feedback. Appendix A.1 also lists three reward-bearing data types: *Seq + Intent → SID* optimizes the recommendation policy, *Text → SID* preserves text-to-EduSID alignment, and *Problem Titles → SID* preserves sequential recommendation ability.

For a sampled identifier $s \in \mathcal{S}_{\text{legal}}$, let $q(s) = \phi^{-1}(s)$. The KT simulator predicts an expected correctness probability $\hat{p}_s = P_\psi(a = 1 \mid H_t, A_t, q(s))$, which is used as the expected response for deterministic offline reward computation:

$$M_{t+1}^s = f_\psi\big(H_t \cup \{q(s)\}, A_t \cup \{\hat{p}_s\}\big). \tag{16}$$

The KT gain is the average mastery increase on the intent-related concepts $\mathcal{C}(\nu_t)$:

$$R_{\text{KT}}(x_t, s) = \frac{1}{|\mathcal{C}(\nu_t)|} \sum_{c \in \mathcal{C}(\nu_t)} \left(M_{t+1,c}^s - M_{t,c}\right). \tag{17}$$

The complete reward combines validity, prefix consistency, multi-positive matching, KT gain, and repetition:

$$R(x_t, s) = r_{\text{val}} + \alpha r_\rho + \beta r_{\text{pos}} + \gamma r_{\text{KT}} + \eta r_{\text{rep}}. \tag{18}$$

Here, $r_{\text{val}} = \mathbb{I}[s \in \mathcal{S}_{\text{legal}}]$ encourages legal EduSID generation during RL sampling, $r_\rho = \mathbb{I}[\rho(s) = \rho_t]$ rewards target-prefix consistency, $r_{\text{pos}} = \mathbb{I}[s \in \mathcal{P}_t]$ rewards matching the multi-positive set, $r_{\text{KT}} = R_{\text{KT}}(x_t, s)$ measures KT-estimated mastery gain, and $r_{\text{rep}} = -R_{\text{rep}}(s, H_t)$ penalizes repeated recommendations.

Following group relative policy optimization (GRPO) (Shao et al., 2024), which has also been adopted for generative recommendation alignment (Deng et al., 2025; Kong et al., 2025), for

each prompt x_t we sample a group of G candidate identifiers $\{s_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x_t)$, where G denotes the group size. We then normalize rewards within the group and optimize the policy with advantage-weighted log likelihood:

$$\bar{R} = \frac{1}{G}\sum_{i=1}^{G} R_i, \qquad A_i = \frac{R_i - \bar{R}}{\sqrt{\frac{1}{G}\sum_{j=1}^{G}(R_j - \bar{R})^2 + \epsilon}}. \tag{19}$$

The policy loss is

$$\mathcal{L}_{\text{rl}} = -\frac{1}{G}\sum_{i=1}^{G} A_i \log \pi_\theta(s_i \mid x_t). \tag{20}$$

KT gain is used only for learner-intent recommendation prompts; the text and title alignment prompts use structural rewards.

At deployment time, constrained decoding selects a legal EduSID and maps it back to an exercise:

$$s_t = \arg\max_{s \in \mathcal{S}_{\text{legal}}} \pi_\theta(s \mid x_t), \qquad q_t = \phi^{-1}(s_t), \tag{21}$$

After exercise q_t, the response a_t is appended and the learner state is updated as $H_{t+1} = H_t \cup \{q_t\}$, $A_{t+1} = A_t \cup \{a_t\}$, and $M_{t+1} = f_\psi(H_{t+1}, A_{t+1})$. Repeating this step yields an intent-driven exercise path. The KT gain is used only as an offline proxy.

5 Experiments

We evaluate LIGER under a controlled offline protocol on three full-scale educational datasets. The experiments examine three aspects: overall effectiveness against sequential, RL-based, and LLM-agent baselines; component contributions from SFT supervision, KT-RL rewards, and the EduSID representation; and behavioral reliability under counterfactual intents, different input fields, rollout lengths, and evaluator simulators.

5.1 Experimental Setup

Datasets. We evaluate on three educational datasets from different domains. **BePKT** records programming learning trajectories (Zhu et al., 2022). **Junyi** contains large-scale exercise logs from the Junyi Academy platform (Chang et al., 2015). **XES3G5M** is a mathematics KT benchmark with richer context and more knowledge concepts (Liu et al., 2023). Table 1 summarizes the data statistics.

Dataset	BePKT	Junyi	XES3G5M
Users	733	1,696	1,806
Interactions	75,519	257,693	554,918
Exercises	600	835	7,652
Knowledge Concepts	72	40	865

Table 1: Dataset statistics for the three full datasets.

Offline simulator and metric. At each rollout step, a policy recommends one exercise, and the KT simulator predicts an expected correctness probability to update the learner state before the next recommendation. We use Final Intent-Ep, defined in Eq. (6), as the primary evaluation metric in all experiments.

Baselines. We compare with similar-path, sequential, reinforcement learning, educational RL, and LLM-agent baselines. KNN (Cover and Hart, 1967) is a similarity-based non-parametric baseline that recommends from nearest historical patterns. GRU4Rec (Hidasi et al., 2016) and SASRec (Kang and McAuley, 2018) are sequential recommenders based on recurrent and self-attentive sequence modeling, respectively. DQN (Chen et al., 2018) is a value-based adaptive-learning recommender, while Actor-Critic (Konda and Tsitsiklis, 1999) represents a standard policy-gradient RL baseline. GEHRL (Li et al., 2023) is a graph-enhanced hierarchical RL method for goal-oriented learning-path recommendation. GenAL (Lv et al., 2025) is an LLM-agent baseline for adaptive learning, and ExRec (Ozyurt et al., 2025) is a semantically grounded KT-guided RL method for exercise recommendation.

To evaluate both intent-free and language-conditioned methods, we adopt two complementary protocols. For models that cannot process natural-language intents, we compare mastery improvement on the same target knowledge concept under a shared concept-improvement objective. This allows sequential and RL-based baselines to participate in the main comparison without requiring language inputs. For language-conditioned models, we further evaluate intent-driven generation by measuring how recommendation paths change under different natural-language intents for the same learner state. In both protocols, future exercises are never exposed as model inputs.

Implementation details. We instantiate LIGER with Qwen3-0.6B, Qwen3-1.7B, and Qwen3-4B

Method	Final Intent-Ep ↑		
	BePKT	Junyi	XES3G5M
Baselines			
KNN	$-0.0071_{\pm 0.0011}$	$0.1239_{\pm 0.0023}$	$0.0539_{\pm 0.0026}$
GRU4Rec	$0.0097_{\pm 0.0009}$	$0.1333_{\pm 0.0017}$	$-0.0067_{\pm 0.0032}$
SASRec	$0.0263_{\pm 0.0019}$	$0.1176_{\pm 0.0029}$	$-0.0233_{\pm 0.0036}$
DQN	$0.0249_{\pm 0.0021}$	$0.1223_{\pm 0.0030}$	$0.0040_{\pm 0.0028}$
Actor-Critic	$0.0203_{\pm 0.0015}$	$0.1018_{\pm 0.0034}$	$0.0674_{\pm 0.0042}$
GEHRL	$0.0294_{\pm 0.0018}$	$0.1257_{\pm 0.0026}$	$0.0601_{\pm 0.0013}$
GenAL	$0.0217_{\pm 0.0027}$	$0.1362_{\pm 0.0037}$	$0.0409_{\pm 0.009}$
ExRec	$0.0305_{\pm 0.0017}$	$0.0838_{\pm 0.0035}$	$0.0628_{\pm 0.0023}$
LIGER variants			
$\text{SFT}_{0.6\text{B}}$	$0.0403_{\pm 0.0016}$	$0.1531_{\pm 0.0023}$	$0.0724_{\pm 0.0020}$
$\text{SFT}_{1.7\text{B}}$	$0.0505_{\pm 0.0009}$	$0.1621_{\pm 0.0013}$	$0.0937_{\pm 0.0017}$
$\text{SFT}_{4\text{B}}$	$0.0603_{\pm 0.0014}$	$0.1653_{\pm 0.0039}$	$0.0981_{\pm 0.0028}$
$\text{KT-RL}_{0.6\text{B}}$	$0.0501_{\pm 0.0008}$	$0.1563_{\pm 0.0022}$	$0.0810_{\pm 0.0026}$
$\text{KT-RL}_{1.7\text{B}}$	$0.0565_{\pm 0.0010}$	$0.1738_{\pm 0.0014}$	$0.1092_{\pm 0.0011}$
$\text{KT-RL}_{4\text{B}}$	$\mathbf{0.0645}_{\pm 0.0013}$	$\mathbf{0.1758}_{\pm 0.0041}$	$\mathbf{0.1107}_{\pm 0.0038}$

Table 2: Main comparison on the three full datasets. Values are mean ± standard deviation of Final Intent-Ep over five random seeds.

SFT Training Mixture	BePKT	Junyi	XES3G5M
Full SFT mixture	0.0505	0.1621	0.0937
w/o seq_answer_to_sid	0.0121	0.1277	0.0502
w/o intent_seq_to_sid	0.0193	0.0900	0.0310
w/o intent_to_prefix	0.0283	0.1323	0.0711
w/o text2sid	0.0456	0.0710	0.0827
w/o sid2text	0.0297	0.0903	0.0812

Table 3: SFT data ablation after removing one supervised task.

Reward Setting	BePKT	Junyi	XES3G5M
Full reward	0.0565	0.1738	0.1092
w/o r_{val}	0.0550	0.1701	0.1056
w/o r_{ρ}	0.0537	0.1634	0.1024
w/o r_{pos}	0.0539	0.1652	0.1017
w/o r_{rep}	0.0542	0.1688	0.0996
w/o r_{KT}	0.0517	0.1639	0.0982

Table 4: KT-RL reward ablation after removing one reward component.

backbones (Qwen Team, 2025); later diagnostics use Qwen3-1.7B by default for cost-efficiency. Each prompt contains history, intent, mastery, and response fields, and the tokenizer is extended with all EduSID tokens. Appendices A.1 and A.2 report data mixtures and training details.

5.2 Main Results

Table 2 reports the main comparison under the full-dataset offline protocol using Final Intent-Ep in Eq. (6). The strongest baseline differs by dataset: ExRec on BePKT, GenAL on Junyi, and Actor-Critic on XES3G5M. The best LIGER variant is consistently KT-RL with the Qwen3-4B backbone.

Compared with the strongest non-LIGER baseline on each dataset, LIGER-KT-RL with Qwen3-4B improves Final Intent-Ep by 0.0340, 0.0396, and 0.0433 on BePKT, Junyi, and XES3G5M, respectively. KT-RL consistently improves over the corresponding SFT checkpoint across all backbones and datasets, with gains from 0.0032 to 0.0155. This indicates that KT-guided policy optimization provides additional alignment with intent-related mastery improvement beyond Future-Hindsight imitation. Since the gap from Qwen3-1.7B to Qwen3-4B is relatively small on Junyi and XES3G5M, we use Qwen3-1.7B for the following ablations to balance performance and cost.

5.3 Ablation Studies

All ablation studies use Qwen3-1.7B as the backbone and Final Intent-Ep as the evaluation metric.

SFT supervision ablation. Table 3 shows that the full SFT mixture is best on all datasets. The largest degradation comes from removing `intent_seq_to_sid`, especially on XES3G5M, where the score drops from 0.0937 to 0.0310. Removing `seq_answer_to_sid` also hurts consistently, indicating that answer-conditioned sequence modeling remains a necessary base signal. Text-alignment tasks matter most on Junyi.

KT-RL reward ablation. Table 4 shows that every component contributes. Removing r_{KT} gives the largest average drop, consistent with the mastery-gain objective. Prefix and multi-positive rewards are also important, showing that structural intent consistency and alternative valid targets both shape the rollout. The validity term has the smallest effect because constrained decoding already biases outputs toward legal EduSID tokens.

EduSID representation ablation. Table 5 compares raw item identifiers, unstructured or generic SID variants, and partial EduSID variants. Full EduSID performs best on all three datasets. Raw Item ID and Random SID are much weaker, suggesting that arbitrary identifiers are insufficient for intent-conditioned generation. Semantic SID (Rajput et al., 2023) is better, but removing the knowledge or difficulty component still reduces performance, indicating that EduSID benefits from both pedagogical target and challenge-level structure. Additional input-field and RL-sample ablations are reported in Appendices C.1.2 and C.1.1; learning-rate sensitivity is reported in Appendix A.3.

Variant	BePKT	Junyi	XES3G5M
Raw Item ID	0.0114	0.0417	0.0138
Random SID	0.0346	0.0595	0.0212
Semantic SID	0.0438	0.1169	0.0845
EduSID w/o Knowledge	0.0392	0.1384	0.0776
EduSID w/o Difficulty	0.0476	0.1541	0.0918
Full EduSID	0.0565	0.1738	0.1092

Table 5: EduSID representation ablation.

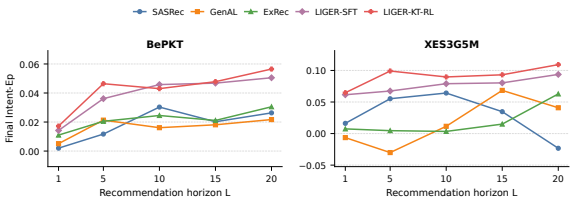

Figure 5: Path-length curves on BePKT and XES3G5M with the Qwen3-1.7B backbone. Values are intermediate Final Intent-Ep across rollout horizons.

5.4 Diagnostic Analysis

Intent sensitivity. We first test whether a method changes its recommendations when only the stated intent changes. For each case, the learner history, answers, and mastery state are fixed, while the original intent is replaced by a counterfactual intent over the same concept but with a different pedagogical need. We generate two length-L paths and report Path Diff., the fraction of positions with different recommended exercises. Larger values indicate stronger responsiveness to intent changes.

Table 6 shows that LIGER is more sensitive to counterfactual intents than GenAL on all three datasets. KT-RL further increases Path Diff. over SFT on each dataset, suggesting that reward-guided refinement strengthens intent responsiveness.

Path length. Figure 5 plots intermediate Final Intent-Ep values for BePKT and XES3G5M from the same selected run at $L \in \{1, 5, 10, 15, 20\}$; the Junyi curve is reported in Appendix C.2.2. The curves indicate that LIGER-KT-RL is not only strong at the final horizon, but also remains competitive or strongest at most intermediate horizons. The few short-horizon exceptions, such as BePKT at $L = 10$ and Junyi at $L = 1$, suggest that KT-RL mainly improves multi-step trajectory quality rather than every early step.

Cross-simulator robustness. To test whether the result depends on a single evaluator, we re-evaluate representative methods with auxiliary KT simulators while keeping the reward and learner-state source fixed to DKT. Table 7 reports the BePKT result under DKT, SAKT, and AKT simulators; the remaining datasets are reported in Appendix C.2.1.

Method	BePKT	Junyi	XES3G5M
GenAL	0.49	0.53	0.47
LIGER-SFT	0.58	0.64	0.61
LIGER-KT-RL	0.67	0.71	0.67

Table 6: Intent sensitivity under counterfactual intent prompts. Values are Path Diff.; higher is better.

Method	DKT Sim.	SAKT Sim.	AKT Sim.
SASRec	0.0263	0.0274	0.0254
ExRec	0.0305	0.0318	0.0308
LIGER-SFT	0.0505	0.0527	0.0486
LIGER-KT-RL	0.0565	0.0580	0.0531

Table 7: Cross-simulator evaluation on BePKT with the Qwen3-1.7B backbone for LIGER variants. Values are Final Intent-Ep at $L = 20$ with the reward and learner-state source fixed to DKT.

The BePKT result preserves the main trend: LIGER-KT-RL remains strongest under all three evaluators. The supplementary Junyi and XES3G5M results show the same ordering, although absolute values vary across simulators. This suggests that the observed gains are not solely tied to the DKT evaluator.

6 Conclusion

We presented LIGER, a learner-intent-driven generative framework for exercise recommendation. LIGER derives intent-conditioned supervision from ordinary logs through Future-Hindsight construction, represents exercises with pedagogically structured EduSID actions, and optimizes an intent- and learner-state-conditioned policy with structured SFT losses and KT-guided rewards. Across three full datasets, LIGER-KT-RL achieves the strongest Final Intent-Ep in the main comparison. The intent-sensitivity and ablation studies further show that natural-language intents, structured EduSID representation, intent-conditioned supervision, text–EduSID alignment, reward components, and learner-state inputs all contribute to the final behavior. These results support learner intent as a useful control signal for generative exercise recommendation and suggest that structured educational identifiers can make generated recommendations more aligned with pedagogical targets.

Limitations

The present study focuses on a controlled offline setting for intent-conditioned exercise generation. Future deployments should collect naturally expressed learner goals and teacher feedback, and should compare templated intent supervision with more open-ended learner requests. The reported gains also measure simulator-estimated mastery under logged-data protocols; online studies with real learners would be needed to establish classroom impact, long-term engagement, and robustness under changing curricula. The action space is also limited to exercise recommendation, whereas practical tutoring systems often need to choose among explanations, hints, videos, worked examples, and multimodal materials. Finally, larger item banks and longer recommendation horizons may require more efficient constrained decoding and additional safeguards for diversity, fairness, and teacher controllability.

Ethical Considerations

This work studies learner-intent-driven exercise recommendation from educational problem-solving data. All experiments are conducted on publicly available datasets that have been anonymized and do not contain personally identifiable information. We do not attempt to infer sensitive attributes or re-identify individual students, and all reported analyses are performed at the aggregate level.

The proposed model is intended for research on educational recommendation and student modeling, rather than for high-stakes educational decision making. As with other data-driven educational models, the learned policies and EduSID representations may reflect biases in the underlying datasets, concept annotations, and logged learning trajectories. Their interpretations and recommendations should therefore be treated with caution, especially when transferred to new learner populations, curricula, or educational contexts.

The learning intents used in this paper are proxy intents constructed from future windows within the same data split; they should not be interpreted as authentic learner utterances. In addition, the mastery-improvement signal used for evaluation and KT-guided reinforcement learning is an offline simulator proxy rather than causal evidence of real learning gains. Any real-world deployment of learner-intent-driven exercise recommendation should involve appropriate human oversight, transparent communication with learners and teachers, and careful consideration of educational, social, and ethical implications.

References

Fangzhe Ai, Yishuai Chen, Yuchun Guo, Yongxiang Zhao, Zhenzhu Wang, Guowei Fu, and Guangyan Wang. 2019. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 240–245.

Anonymous. 2026. Beyond static question banks: Dynamic knowledge expansion via LLM-automated graph construction and adaptive generation. *arXiv preprint arXiv:2602.00020*.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1007–1014.

Dapeng Cai, Xiaofei Zhao, Jun Wang, Xiaohui Tang, and Wei Li. 2019. Learning path recommendation based on knowledge tracing model and reinforcement learning. In *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication*, pages 309–313.

Haw-Shiuan Chang, Hwai-Jung Hsu, Kuan-Ta Chen, and 1 others. 2015. Modeling exercise relationships in e-learning: A unified approach. In *Proceedings of the 8th International Conference on Educational Data Mining*, pages 532–535.

Jyun-Yi Chen, Saeed Saeedvand, and I-Wei Lai. 2023. Adaptive learning path navigation based on knowledge tracing and reinforcement learning. *Preprint*, arXiv:2305.04475.

Yunxiao Chen, Xiaoou Li, Jingchen Liu, and Zhiliang Ying. 2018. Recommendation system for adaptive learning. *Applied Psychological Measurement*, 42(1):24–41.

Albert T. Corbett and John R. Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315.

Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2330–2339.

Donghee Han, Hwanjun Song, and Mun Yong Yi. 2025. Rethinking llm-based recommendations: A personalized query-driven parallel integration. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8395–8419.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*.

Wacharawan Intayoad, Chayapol Kamyod, and Punnarumol Temdee. 2020. Reinforcement learning based on contextual bandits for personalized online learning recommendation systems. *Wireless Personal Communications*, 115(4):2917–2932.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining*, pages 197–206.

Vijay Konda and John Tsitsiklis. 1999. Actor-critic algorithms. *Advances in Neural Information Processing Systems*, 12.

Xiaoyu Kong, Leheng Sheng, Junfei Tan, Yuxin Chen, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. 2025. Minionerec: An open-source framework for scaling generative recommendation. *arXiv preprint arXiv:2510.24431*.

Yoshiki Kubotani, Yoshihiro Fukuhara, and Shigeo Morishima. 2021. RLTutor: Reinforcement learning based adaptive tutoring system by modeling virtual student with fewer interactions. *arXiv preprint arXiv:2108.00268*.

Qingyao Li, Wei Xia, Li'ang Yin, Jian Shen, Renting Rui, Weinan Zhang, Xianyu Chen, Ruiming Tang, and Yong Yu. 2023. Graph enhanced hierarchical reinforcement learning for goal-oriented learning path recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1318–1327.

Zhaoxing Li, Jindi Wang, Wen Gu, Vahid Yazdanpanah, Lei Shi, Alexandra I. Cristea, Sarah Kiden, and Sebastian Stein. 2025. TutorLLM: Customizing learning recommendations with knowledge tracing and retrieval-augmented generation. In *IFIP Conference on Human-Computer Interaction*, pages 137–146. Springer.

Yu Liang, Zhongjin Zhang, Yuxuan Zhu, Kerui Zhang, Zhiluohan Guo, Wenhang Zhou, Zonqi Yang, Kangle Wu, Yabo Ni, Anxiang Zeng, and 1 others. 2026. Rethinking generative recommender tokenizer: Recsys-native encoding and semantic quantization beyond llms. *arXiv preprint arXiv:2602.02338*.

Qi Liu, Shiwei Tong, Chuanren Liu, Hongke Zhao, Enhong Chen, Haiping Ma, and Shijin Wang. 2019. Exploiting cognitive structure for adaptive learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 627–635.

Zitao Liu, Qiongqiong Liu, Teng Guo, Jiahao Chen, Shuyan Huang, Xiangyu Zhao, Jiliang Tang, Weiqi Luo, and Jian Weng. 2023. XES3G5M: A knowledge tracing benchmark dataset with auxiliary information. *Advances in Neural Information Processing Systems*, 36:32958–32970.

Wensheng Lu, Jianxun Zhang, Jun Wang, Hong Chen, Xiang Li, Yang Zhang, and Ji-Rong Wen. 2024. Aligning large language models for controllable recommendations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Rui Lv, Qi Liu, Weibo Gao, Haotian Zhang, Junyu Lu, and Linbo Zhu. 2025. GenAL: Generative agent for adaptive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 577–585.

Fnu Mohbat and Mohammed J. Zaki. 2025. KERL: Knowledge-enhanced personalized recipe recommendation using large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19125–19141.

Amir Hossein Nabizadeh, Daniel Gonçalves, Sandra Gama, Joaquim Jorge, and Hamed N. Rafsanjani. 2020. Adaptive learning path recommender approach using auxiliary learning objects. *Computers & Education*, 147:103777.

Yilmazcan Ozyurt, Tunaberk Almaci, Stefan Feuerriegel, and Mrinmaya Sachan. 2025. Personalized exercise recommendation with semantically-grounded knowledge tracing. *Preprint*, arXiv:2507.11060.

Shalini Pandey and George Karypis. 2019. A self-attentive model for knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 384–389.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J. Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, volume 28.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shashank Rajput, Nikhil Mehta, Aarushi Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukas Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonathan Samost, Maciej Kula, Maheswaran Sathiamoorthy, and Ed H. Chi. 2023. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450.

John K. Tarus, Zhendong Niu, and Ghulam Mustafa. 2018. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, 50:21–48.

Xinfeng Wang, Jin Cui, Fumiyo Fukumoto, and Yoshimi Suzuki. 2024a. Enhancing high-order interaction awareness in llm-based recommender model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11696–11711.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024b. RecMind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4351–4364.

Xiangrui Xiong, Yichuan Lu, Zifei Pan, and Chang Sun. 2025. CLLMRec: LLM-powered cognitive-aware concept recommendation via semantic alignment and prerequisite knowledge distillation. *arXiv preprint arXiv:2511.17041*.

Austin Xu, Will Monroe, and Klinton Bicknell. 2024a. Large language model augmented exercise retrieval for personalized language learning. In *Proceedings of the 14th Learning Analytics and Knowledge Conference*, pages 284–294.

Songlin Xu, Xinyu Zhang, and Lianhui Qin. 2024b. EduAgent: Generative student agents in learning. *arXiv preprint arXiv:2404.07963*.

Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, pages 765–774.

Jizhi Zhang, Chongming Gao, Wentao Shi, Xin Chen, Jingang Wang, Xunliang Cai, and Fuli Feng. 2025. Leveraging unpaired feedback for long-term llm-based recommendation tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24507–24521.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *Preprint*, arXiv:2305.07001.

Yang Zhang, Keqin Bao, Ming Yan, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024. Text-like encoding of collaborative information in large language models for recommendation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9181–9191.

Renyu Zhu, Dongxiang Zhang, Chengcheng Han, Ming Gao, Xuesong Lu, Weining Qian, and Aoying Zhou. 2022. Programming knowledge tracing: A comprehensive dataset and a new model. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 298–307. IEEE.

A Implementation and Reproducibility Details

This appendix summarizes the data mixtures, training configuration, and supplementary diagnostics used to reproduce the reported results.

A.1 Training Data Mixtures

The SFT stage combines five record types: the main sequence-and-intent recommendation task, sequential recommendation without intent, intent-to-prefix supervision, text-to-EduSID alignment, and EduSID-to-text alignment. KT-RL uses three reward-optimized record types: sequence-and-intent recommendation, text-to-EduSID alignment, and history-title-to-EduSID alignment. Figure 6 reports the resulting record-level proportions.

Both stages are dominated by sequence-derived supervision, while text-alignment tasks serve as auxiliary signals. Sequence-derived tasks contribute one record per generated training instance, and text-alignment tasks contribute one record per valid EduSID item. The final mixtures are simple concatenations of these records without additional manual reweighting.

A.2 Training and Evaluation Configuration

The KT simulator is a DKT model trained on the train/validation learner split with Adam, batch size 64, learning rate 10^{-3}, 50 epochs, 128-dimensional embeddings, 128 hidden units, one recurrent layer and dropout 0.1. The checkpoint with the best validation AUC is used for reward computation and offline evaluation. SFT and KT-RL use bfloat16, left padding, validation Final Intent-Ep for checkpoint selection, and are run on 4-GPU H100 servers. Table 8 summarizes the selected training configuration.

Dataset	SFT setting	KT-RL setting
BePKT	lr 10^{-5}, epochs 100, batch 32	lr 5×10^{-7}, epochs 10, $G=2$, $\beta=0.04$
Junyi	lr 2×10^{-6}, epochs 50, batch 64	lr 10^{-6}, epochs 10, $G=4$, $\beta=0.04$
XES3G5M	lr 10^{-5}, epochs 10, batch 32	lr 10^{-6}, epochs 10, $G=4$, $\beta=0.04$

Table 8: Compact configuration summary for the reported full-dataset runs.

The selected configuration reflects dataset-specific tuning under the full-dataset protocol. Junyi uses a smaller SFT learning rate and larger batch size than the other two datasets, while KT-RL consistently uses $\beta = 0.04$ and temperature 1.0. These settings are reported for reproducibility and should not be read as globally optimal hyperparameters outside this protocol.

A.3 Learning-Rate Sensitivity

Figures 7 and 8 report learning-rate sensitivity studies that are useful for reproduction but secondary to the main comparison.

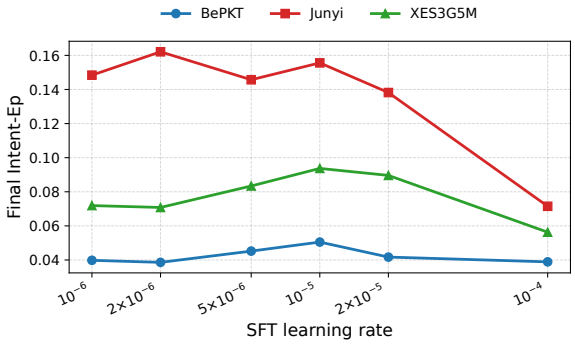

Figure 7: SFT learning-rate sensitivity across the three datasets.

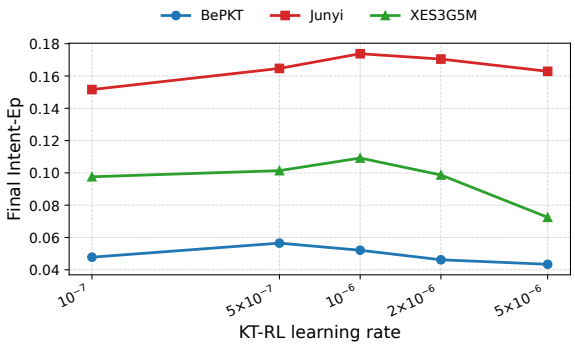

Figure 8: KT-RL learning-rate sensitivity across the three datasets.

The SFT results favor moderate learning rates: BePKT and XES3G5M peak at 10^{-5}, while Junyi prefers 2×10^{-6}. KT-RL is more sensitive, with the best values concentrated around 5×10^{-7} to 10^{-6}; larger rates consistently reduce Final Intent-Ep.

B Future-Hindsight Construction Details

This section makes the Future-Hindsight pipeline explicit without adding additional appendix tables. The construction first fixes the train/validation/test split, then selects a prefix H_t and a same-split future window F_t. The earliest future item defines

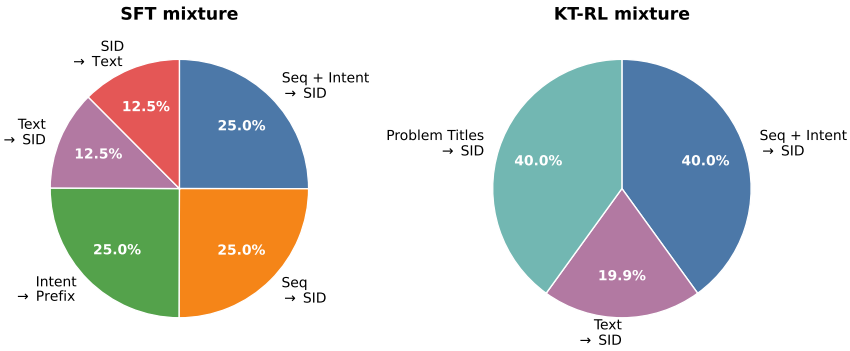

Figure 6: Record-level data-mixture proportions for SFT and KT-RL. Sequence-derived tasks are counted per training record, and text-alignment tasks are counted per valid EduSID item.

the canonical target, same-prefix future items provide multi-positive alternatives, and the final proxy intent type is selected from recent errors, the target prefix, difficulty, and KT-estimated mastery. The key leakage-safe design choice is that future information is used only to construct supervision targets within the same split, never as model input.

The four rendered intent types are controlled supervision signals rather than authentic learner utterances. Representative templates are: *error consolidation*, "I got similar problems wrong recently and want to consolidate this knowledge point"; *weakness remediation*, "I am still weak in this concept and want more practice"; *difficulty challenge*, "I understand the basics and want a harder exercise"; and *regular practice*, "I want to continue practicing this topic".

C Supplementary Results

This section reports additional diagnostics that support, but are not required for, the main comparisons in Section 5. Unless stated otherwise, the results use the same offline Final Intent-Ep metric and the Qwen3-1.7B backbone.

C.1 Additional Ablation Results

C.1.1 KT-RL Data Ablation

KT-RL training mixture			
Variant	BePKT	Junyi	XES3G5M
Full RL	0.0565	0.1738	0.1092
w/o intent	0.0512	0.1634	0.0922
w/o text-to-SID	0.0534	0.1688	0.0995
w/o history-title	0.0541	0.1703	0.1006
Input field ablation			
Variant	BePKT	Junyi	XES3G5M
Full Model	0.0565	0.1738	0.1092
w/o History	0.0153	0.0326	0.0239
w/o Answers	0.0489	0.1542	0.0928
w/o Language Intent	0.0432	0.1310	0.0871
w/o Mastery State	0.0450	0.1526	0.1062

Table 9: Supplementary ablations with the Qwen3-1.7B backbone. Values are Final Intent-Ep.

Table 9 shows that all RL sample types contribute. Removing `intent_rl` causes the largest drop because it directly optimizes the intent-conditioned recommendation prompt. The two auxiliary RL tasks have smaller but consistent effects, suggesting that they help preserve text-to-identifier alignment and sequential executability during policy optimization.

C.1.2 Input Field Ablation

To test whether the policy uses the full learner state rather than behaving as a shallow query-to-exercise matcher, we remove one input field at a time from the KT-RL input. The lower panel of Table 9 shows that the full input performs best on all three datasets. Removing learner history causes the largest drop, especially on Junyi, where the score falls from 0.1738 to 0.0326. Removing language intent also consistently hurts performance, confirming that the model uses the stated goal rather than only sequential patterns. Answer feedback and mastery state

provide additional gains, but their effects are more dataset-dependent.

C.2 Additional Robustness Results

C.2.1 Additional Cross-Simulator Results

Dataset	Method	DKT Sim.	SAKT Sim.	AKT Sim.
Junyi	SASRec	0.1176	0.1231	0.1003
Junyi	ExRec	0.0838	0.0920	0.0812
Junyi	LIGER-SFT	0.1621	0.1728	0.1592
Junyi	LIGER-KT-RL	0.1738	0.1803	0.1695
XES3G5M	SASRec	-0.0233	-0.0186	-0.1096
XES3G5M	ExRec	0.0628	0.0683	0.0593
XES3G5M	LIGER-SFT	0.0937	0.1012	0.0897
XES3G5M	LIGER-KT-RL	0.1092	0.1173	0.0965

Table 10: Additional cross-simulator evaluation on Junyi and XES3G5M with the Qwen3-1.7B backbone for LIGER variants. Values are Final Intent-Ep at $L = 20$ with the reward and learner-state source fixed to DKT.

Table 10 shows that LIGER-KT-RL remains strongest under DKT, SAKT, and AKT evaluation on both datasets.

C.2.2 Junyi Path-Length Curve

Figure 9 is consistent with the main-text path-length analysis: KT-RL remains competitive across intermediate horizons and reaches the strongest final value at $L = 20$.

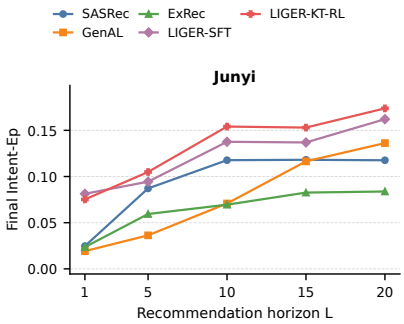

Figure 9: Path-length curve on Junyi with the Qwen3-1.7B backbone. Values are intermediate Final Intent-Ep across rollout horizons.