A APPENDIX

The Appendix is organized as follows:

- Appendix A.1 and A.2 provides a detailed derivation of our objective function ELBO in two scenarios: inference involving complete learning histories (Sec. 3.2.1) and inference for in a continual learning setting (Sec. 3.2.2).
- Appendix A.3 provides in-depth descriptions of baseline models, and the details and the selection criterion of the three datasets we use for experiments.
- Appendix A.4 describes the PSI-KT architecture in full detail, including its hyperparameters.
- Appendix A.5 provides additional prediction results. We show the average accuracy score, average f1-score, and their standard error of Fig. 2 in the prediction experiment given entire learning histories. We also show the average accuracy score and their standard error of Fig. 3 in the prediction experiment for continual learning setup.
- Appendix A.6 describes the details of the experiment setup and how we derive the metrics for specificity, consistency, and disentanglement. We also provide comprehensive results on the operational interpretability of baseline models.
- Appendix A.7 elaborates on the graph assessment framework, including details about the alignment metrics and a discussion of causal support, and extends the main text evaluations to all datasets.

A.1 ELBO OF THE HIERARCHICAL SSM

In this section, we derive the single-learner ELBO, Eqs. 8- 10 in the main text. For clarity, we omit the superindex ℓ in these derivations. Note that the parameters ϕ and θ are global, i.e., they are optimized based on the entire interaction data across learners.

In VI, we approximate an intractable posterior distribution $p_\theta(z \mid y)$ with $q_\phi(z \mid y)$ from a tractable distribution family. We learn ϕ and θ together by maximizing the evidence lower bound (ELBO) of the marginal likelihood (Blei et al., 2017; Attias, 1999), given by $\log p_\theta(y) \geq \text{ELBO}(\theta, \phi) = \mathbb{E}_{q_\phi(z \mid y)}\left[-\log q_\phi(z \mid y) + \log p_\theta(y, z)\right]$. The two terms in the ELBO represent the entropy of the variational posterior distribution, $\text{H}\big[q_\phi(z \mid y)\big] = \mathbb{E}_{q_\phi(z|y)}\left[-\log q_\phi(z \mid y)\right]$, and the log-likelihood of the joint distribution of observations and latent states $\mathbb{E}_{q_\phi(z|y)} \log p_\theta(y, z)$.

We now formulate the ELBO for our hierarchical SSM (see Fig. 1) with two layers of latent states. We assume that fixed learning histories $\mathcal{H}_{1:N}$ until time t_N are available and we use capital N to represent the fixed time point. We approximate the posterior $p_\theta(\boldsymbol{z}_{1:N}, s_{1:N} \mid y_{1:N})$ using the mean-field factorization, $q_\phi(\boldsymbol{z}_{1:N}, s_{1:N} \mid y_{1:N}) = q_\phi(\boldsymbol{z}_{1:N})\, q_\phi(s_{1:N})$:

$$\begin{aligned}\text{ELBO}(\theta, \phi) &= \text{H}\big[q_\phi(\boldsymbol{z}_{1:N}, s_{1:N} \mid y_{1:N})\big] + \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:N}, s_{1:N}|y_{1:N})} \log p_\theta(y_{1:N}, \boldsymbol{z}_{1:N}, s_{1:N}) \\ &= \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:N}, s_{1:N} \mid y_{1:N})}\big[-\log q_\phi(\boldsymbol{z}_{1:N}, s_{1:N} \mid y_{1:N}) + \log p_\theta(y_{1:N}, \boldsymbol{z}_{1:N}, s_{1:N})\big] \\ &= \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:N})\, q_\phi(s_{1:N})}\big[-\log q_\phi(\boldsymbol{z}_{1:N}) - \log q_\phi(s_{1:N}) + \log p_\theta(y_{1:N}, \boldsymbol{z}_{1:N}, s_{1:N})\big].\end{aligned} \tag{12}$$

In the generative model of PSI-KT, the observation y_n at time t_n depends on the particular knowledge state z_n^k associated with the interacted KC $k = x_n$. All knowledge states z_n rely on previous states z_{n-1} and cognitive traits s_n, which themselves are influenced by s_{n-1}. Thus, we can factorize the joint distribution $p_\theta(y_{1:N}, \boldsymbol{z}_{1:N}, s_{1:N})$ in Eq. 12 over all latent states and observations:

$$\begin{aligned}p_\theta(y_{1:N}, \boldsymbol{z}_{1:N}, s_{1:N}) &= p_\theta(s_{1:N})\, p_\theta(\boldsymbol{z}_{1:N} \mid s_{1:N}) \prod_{n=1}^{N} p_\theta(y_n \mid z_n^{x_n}) \\ &= p_\theta(s_1)\, p_\theta(\boldsymbol{z}_1) \prod_{n=2}^{N} p_\theta(s_n \mid s_{n-1})\, p_\theta(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}, s_n) \prod_{n=1}^{N} p_\theta(y_n \mid z_n^{x_n}).\end{aligned} \tag{13}$$

Here, $p_\theta(s_1)$ and $p_\theta(\boldsymbol{z}_1)$ are the Gaussian initial priors for the latent states.

By incorporating the factorized joint distribution (Eq. 13), the ELBO for PSI-KT can be derived as follows:

$$\begin{aligned}
\mathrm{ELBO}(\theta,\phi) &= \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:N})\,q_\phi(s_{1:N})}\big[-\log q_\phi(\boldsymbol{z}_{1:N}) - \log q_\phi(s_{1:N}) + \log p_\theta(y_{1:N},\boldsymbol{z}_{1:N},s_{1:N})\big] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:N})\,q_\phi(s_{1:N})}\Big[-\log q_\phi(\boldsymbol{z}_{1:N}) - \log q_\phi(s_{1:N}) + \log p_\theta(s_1) + \log p_\theta(\boldsymbol{z}_1) \\
&\qquad\qquad + \sum_{n=2}^{N}\log p_\theta(s_n\,|\,s_{n-1}) + \sum_{n=2}^{N}\log p_\theta(\boldsymbol{z}_n\,|\,\boldsymbol{z}_{n-1},s_n) \\
&\qquad\qquad + \sum_{n=1}^{N}\log p_\theta(y_n\,|\,z_n^{x_n})\Big] \\
&= \mathbb{E}_{q_\phi(s_{1:N})}\big[-\log q_\phi(s_{1:N}) + \log p_\theta(s_1) + \sum_{n=2}^{N}\log p_\theta(s_n\,|\,s_{n-1})\big] \\
&\quad + \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:N})}\big[-\log q_\phi(\boldsymbol{z}_{1:N}) + \log p_\theta(\boldsymbol{z}_1) + \sum_{n=1}^{N}\log p_\theta(y_n\,|\,z_n^{x_n})\big] \\
&\quad + \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:N})\,q_\phi(s_{1:N})}\big[\sum_{2}^{n}\log p_\theta(\boldsymbol{z}_n\,|\,\boldsymbol{z}_{n-1},s_n)\big].
\end{aligned} \tag{14}$$

A.2 EXTENSION TO CONTINUAL LEARNING

We now extend the ELBO to the continual learning setting, where we observe learning performances $y_{1:n}$ sequentially. Here we use the lower case n to indicate the running time index. We seek the posterior distribution $p_\theta(\boldsymbol{z}_n, s_n\,|\,y_{1:n})$ at each interaction time t_n given all observations so far. Usually, one would approximate the posterior with the variational posterior distribution $q_{\phi_n}(\boldsymbol{z}_n, s_n\,|\,y_{1:n}) = q_{\phi_n}(\boldsymbol{z}_n)q_{\phi_n}(s_n)$ using the mean-field factorization (Eq. 12). In that case, the inference process consists of maximizing the $\mathrm{ELBO}(\theta,\phi_n)$ only over ϕ_n:

$$\mathrm{ELBO}(\theta,\phi_n) = \mathbb{E}_{q_{\phi_n}(\boldsymbol{z}_n)q_{\phi_n}(s_n)}[-\log q_{\phi_n}(\boldsymbol{z}_n) - \log q_{\phi_n}(s_n) + \log p_\theta(y_{1:n},\boldsymbol{z}_n,s_n)]. \tag{15}$$

However, it is challenging to calculate the joint distribution $p_\theta(y_{1:n},\boldsymbol{z}_n,s_n)$ in our setup, since it requires marginalizing the full joint distribution $p_\theta(y_{1:n},\boldsymbol{z}_{1:n},s_{1:n})$ over all $\boldsymbol{z}_{n'}$ and $s_{n'}$ with $n' < n$.

Henceforth, we aim to reconfigure the objective function $\mathrm{ELBO}(\theta,\phi_n)$, which involves the variational posterior distribution $q_{\phi_n}(\boldsymbol{z}_n, s_n\,|\,y_{1:n})$ at the time point t_n, to establish a linkage with the posterior $q_{\phi_{n-1}}(\boldsymbol{z}_{n-1}, s_{n-1}\,|\,y_{1:n-1})$ observed at the preceding time point t_{n-1}. By doing so, we can recursively optimize the variational parameters ϕ_n whenever new observations y_n are received (Nguyen et al., 2017), wherein the initialization draws from the parameters ϕ_{n-1} obtained at time t_{n-1}.

First, we show that the marginal joint distribution $p_\theta(y_{1:n},\boldsymbol{z}_n,s_n)$ is proportional to the prior distribution $p_\theta(\boldsymbol{z}_n,s_n\,|\,y_{1:n-1})$ on $s_n,\boldsymbol{z}_n$ at t_{n-1}:

$$p_\theta(y_{1:n},\boldsymbol{z}_n,s_n) \propto p_\theta(\boldsymbol{z}_n,s_n\,|\,y_{1:n-1})\,p_\theta(y_n\,|\,\boldsymbol{z}_n). \tag{16}$$

Second, we show how the prior distribution $p_\theta(\boldsymbol{z}_n, s_n \,|\, y_{1:n-1})$ can be formulated using the posterior $q_{\phi_{n-1}}(\boldsymbol{z}_{n-1}, s_{n-1} \,|\, y_{1:n-1})$ at the previous time t_{n-1}, which evolves for a single time step:

$$
\begin{aligned}
p_\theta(\boldsymbol{z}_n, s_n \,|\, y_{1:n-1}) &= \int p_\theta(\boldsymbol{z}_n, \boldsymbol{z}_{n-1}, s_n, s_{n-1} \,|\, y_{1:n-1}) \mathrm{d}s_{n-1}\, \mathrm{d}\boldsymbol{z}_{n-1} \\
&= \int p_\theta(\boldsymbol{z}_{n-1}, s_{n-1} \,|\, y_{1:n-1})\, p_\theta(s_n \,|\, s_{n-1})\, p_\theta(\boldsymbol{z}_n \,|\, s_n, \boldsymbol{z}_{n-1})\, \mathrm{d}s_{n-1}\, \mathrm{d}\boldsymbol{z}_{n-1} \\
&= \int q_{\phi_{n-1}}(\boldsymbol{z}_{n-1}, s_{n-1} \,|\, y_{1:n-1})\, p_\theta(s_n \,|\, s_{n-1})\, p_\theta(\boldsymbol{z}_n \,|\, s_n, \boldsymbol{z}_{n-1})\, \mathrm{d}s_{n-1}\, \mathrm{d}\boldsymbol{z}_{n-1} \\
&= \int q_{\phi_{n-1}}(\boldsymbol{z}_{n-1})\, q_{\phi_{n-1}}(s_{n-1})\, p_\theta(s_n \,|\, s_{n-1})\, p_\theta(\boldsymbol{z}_n | s_n, \boldsymbol{z}_{n-1}) \mathrm{d}s_{n-1} \mathrm{d}\boldsymbol{z}_{n-1} \\
&= \underbrace{\mathbb{E}_{q_{\phi_{n-1}}(s_{n-1})}\big[p_\theta(s_n \,|\, s_{n-1})\big]}_{:=\tilde{p}_{\phi_{n-1}}(s_n)} \underbrace{\mathbb{E}_{q_{\phi_{n-1}}(\boldsymbol{z}_{n-1})}\big[p_\theta(\boldsymbol{z}_n \,|\, s_n, \boldsymbol{z}_{n-1})\big]}_{:=\tilde{p}_{\phi_{n-1}}(\boldsymbol{z}_n \,|\, s_n)} .
\end{aligned}
\tag{17}
$$

Substituting $p_\theta(s_n, \boldsymbol{z}_n \,|\, y_{1:n-1}) = \tilde{p}_{\phi_{n-1}}(s_n)\, \tilde{p}_{\phi_{n-1}}(\boldsymbol{z}_n \,|\, s_n)$ using our variational approximation in Eq. 16 in the ELBO, we finally arrive at the objective function for variational continuous learning:

$$
\begin{aligned}
\mathrm{ELBO}_{\mathrm{VCL}}(\theta, \phi_n) = \mathbb{E}_{q_{\phi_n}(s_n)\, q_{\phi_n}(\boldsymbol{z}_n)}\big[&- \log q_{\phi_n}(s_n) - \log q_{\phi_n}(\boldsymbol{z}_n) + \log \tilde{p}_{\phi_{n-1}}(s_n) \\
&+ \log \tilde{p}_{\phi_{n-1}}(\boldsymbol{z}_n | s_n) + \log p_\theta(y_n \,|\, \boldsymbol{z}_n^{x_n})\big] \\
= \mathbb{E}_{q_{\phi_n}(s_n)}\big[&- \log q_{\phi_n}(s_n) + \log \tilde{p}_{\phi_{n-1}}(s_n)\big] \\
+ \mathbb{E}_{q_{\phi_n}(\boldsymbol{z}_n)}\big[&- \log q_{\phi_n}(\boldsymbol{z}_n) + \log p_\theta(y_n \,|\, z_n^{x_n})\big] \\
+ \mathbb{E}_{q_{\phi_n}(\boldsymbol{z}_n)\, q_{\phi_n}(s_n)}\big[&\log \tilde{p}_{\phi_{n-1}}(\boldsymbol{z}_n \,|\, s_n)\big].
\end{aligned}
\tag{18}
$$

This provides a derivation of Eq. 10 as presented in the main text. Here our focus lies in the optimization of the parameters ϕ_n, while holding constant the parameters ϕ_{n-1} acquired from the preceding time step.

A.3 BASELINE MODELS AND DATASETS

A.3.1 BASELINES

KT models aim to predict the performance $\hat{y}_n^\ell$ of the presented KC x_n^ℓ at time t_n^ℓ for each learner ℓ, which amounts to learning the mapping $\hat{y}_n^\ell = f_\theta(\mathcal{H}_{n'<n}^\ell)$ (Sec. 2.1). Because baseline models lack learner-specific parameters, we here describe the prediction process for a single learner, and omit the superindex ℓ for clarity. Extending to multiple learners is straightforward since all parameters θ are global. We use $\tau_n := t_n - t_{n-1}$ to represent the time interval between consecutive interactions of a learner ℓ, and the KC-specific interval $\tau_n^k := t_n^k - t_{n-1}^k$ for consecutive interactions with the same KC k. The number of practice repetitions for each KC k up to time t_n is denoted as c_n^k. The dimension of embeddings D equals 16 in our experiments.

Table 6: Models. *# Emb/KC* is the number of learnable embeddings per KC. *Forgetting* is the functional form of memory decay, with exponential ($\exp$) decay the most common.

Feature	HLR/PPE	DKT	DKTF	HKT	AKT	GKT	QIKT	PSI-KT
# Emb/KC	–	2	2	6	6	3	1	1
Forgetting	exp	–	exp	Hawkes	–	–	–	OU

We compare with eight baseline models (Sec. 4):

a) HLR (Settles & Meeder, 2016) uses the cumulative counts of correct, incorrect, and total interactions of KC k until time t_n, collectively denoted $\boldsymbol{c}_n^k = \big[c_n^{k,1} c_n^{k,0} c_n^k\big]^\intercal \in \mathbb{R}^3$, as well as the last interval τ_n^k. When a learner interacts with KC x_n at time t_n, HLR predicts the probability of a correct performance as:

$$
\hat{y}_n := 2^{-\tau_n^k / h_n^k}, \quad \text{with memory half-life } h_n^k := 2^{\theta^\intercal \boldsymbol{c}_n^k} \text{ and } k = x_n. \tag{19}
$$

The learnable weights $\theta \in \mathbb{R}^3$ modulate the influences of correct, incorrect, and total interaction counts. The training process of HLR does not differentiate features from different

KCs or learners, thus HLR cannot model the relational structure of KCs or any learner-specific characteristics.

b) PPE (Walsh et al., 2018) is similar to HLR in predicting performance as a function of interaction histories. It defines the activation m_n^k of KC k at time t_n, $m_n^k := (c_n^k)^\beta\,(T_n^k)^{-\alpha}$ with separate forgetting rate α and learning rate β. The forgetting term T_n^k is a function of the interaction history, which it summarizes as a weighted average of times τ_n^k elapsed between the exposures to a given KC prior to t_n:

$$T_n^k := \sum_{i=1}^{n-1} w_i^k\,\tau_i^k, \quad \text{with } w_i^k = (\tau_i^k)^{-\eta} \sum_{j=1}^{n-1} (\tau_j^k)^\eta. \tag{20}$$

The forgetting rate α is a function of a *stability term* κ and a cumulative average of interval durations between KC exposures modulated by the slope λ:

$$\alpha_n^k = \kappa + \lambda\left(\frac{1}{n-1}\sum_{j-1}^{n-1}\frac{1}{\ln(\tau_j^k + e)}\right). \tag{21}$$

Finally, PPE treats performance $\hat{y}_n$ as a logistic function of m_n^k with $k = x_n$. The learnable parameters are $\theta = \{\beta, \eta, \kappa, \lambda\}$.

c) DKT (Piech et al., 2015) infers two separate embeddings $u^k = \{u^{k,0}, u^{k,1}\}$ for each KC k, depending on performance. Here, $u^{k,0}, u^{k,1} \in \mathbb{R}^D$ represent incorrect interactions and correct interactions on KC k, respectively, and are shared across all learners, with D being the dimensionality of the embeddings. DKT trains an LSTM (Hochreiter & Schmidhuber, 1997) over $\mathcal{H}_{n'<n}$ to encode the combined information of KC indices and performance. For each learner ℓ and all time points $t_{n'<n}$, DKT takes performance embeddings $u^{x_{n'},0}$ of interacted KC $x_{n'}$ as the input when the performance $y_{n'}$ is incorrect, or $u^{x_{n'},1}$ for correct performance, i.e., DKT takes inputs $\boldsymbol{u}_{n'} = u^{x_{n'},y_{n'}}$ for all $t_{n'}$ with $n' < n$. DKT then predicts the subsequent performance on all KCs $\hat{\boldsymbol{y}}_n = [y_n^1, \ldots, y_n^K]^\intercal \in \mathbb{R}^K$, and chooses only the interacted one, i.e., the x_n-th dimension:

$$\begin{aligned}\boldsymbol{h}_n &= \text{LSTM}(\boldsymbol{u}_{n'<n}; \boldsymbol{W}_h, \boldsymbol{b}_h)\\ \hat{\boldsymbol{y}}_n &= \sigma(\boldsymbol{W}_{\hat{y}h}\boldsymbol{h}_n + b_{\hat{y}})\\ \hat{y}_n &= \hat{\boldsymbol{y}}_n[x_n].\end{aligned} \tag{22}$$

Thus, θ for DKT consists of the neural network parameters $\boldsymbol{W}_h, \boldsymbol{b}_h, \boldsymbol{W}_{\hat{y}h}, \boldsymbol{b}_{\hat{y}}$.

d) DKTF (Nagatani et al., 2019) uses the same LSTM architecture and the same combined KC-performance embeddings, $u^k = \{u^{k,0}, u^{k,1}\}$ that we described above for DKT. DKTF uses additional 3-dimensional features $\boldsymbol{t}_n := [\tau_n, \tau_n^k, c_n^k]$ representing the KC-unspecific and KC-specific intervals defined above and the cumulative interaction counts c_n^k for KC k until time t_n. Then, for inputs of every time point $t_{n'}$, DKTF concatenates the time information $\boldsymbol{t}_{n'}$ with KC performance inputs $\boldsymbol{u}_{n'}$ for the interacted KC $x_{n'}$. DKTF predicts future performances following the same architecture based on concatenated input $[\boldsymbol{t}_{n'<n}; \boldsymbol{u}_{n'<n}]$.

e) HKT (Wang et al., 2021) is the most similar model to our PSI-KT. It uses a Hawkes process to model the structural influence on the state of KC k due to every other KCs state in the past interactions $i \in x_{n'<n}$ until time t_n:

$$\begin{aligned}m_n^k &= \lambda_k + \sum_{i \in x_{n'<n}} a_n^{i,k}\,\kappa(t_n^k - t_{n'}^i)\\ \kappa(t_n^k - t_{n'}^i) &= \exp\Big(-\big(1 + \beta_n^{i,k}\log(t_n^k - t_{n'}^i)\big)\Big).\end{aligned} \tag{23}$$

Here m_n^k includes a base level λ_k and all previous learned KCs' influences $a_n^{i,k}$ weighted by the a temporal exponential decay $\kappa(t_n^k - t_{n'}^i)$. The base level λ_k reflects aspects of KC k but also of the specific assignments that were interacted with at time point t_n, given that distinct assignments can provide practice for a single KC. To model cross-KC influences $a_n^{i,k}$, HKT infers embeddings $\{u_a^{k,0}, u_a^{k,1}, u_a^k\} \in \mathbb{R}^D$ for each KC k. Here $u_a^{k,0}, u_a^{k,1}$ are defined similarly to the DKT embeddings, whereas u_a^k only depends on KC identity k. When interacting with KC k at time t_n, the influence on its state due to having interacted with KC i at time $t_{n'}$ with performance $y_{n'}$ is estimated as $a_n^{i,k} = (u_a^{i,y_{n'}})^\intercal u_a^k$. For the coefficient $\beta_n^{i,k}$, HKT estimates three additional KC-specific embeddings $\{u_\beta^{k,0}, u_\beta^{k,1}, u_\beta^k\}$, and follows similar calculations as for a above. HKT also predicts the performance $\hat{y}_n$ as a logistic function of m_n^k with $k = x_n$.

f) AKT (Ghosh et al., 2020) is a transformer-based model (Vaswani et al., 2017) that learns the structure of KCs implicitly from the self-attention weights. Unlike LSTM models, that only capture temporal information, AKT captures both temporal and structural relations. Specifically, AKT first initializes three embeddings $\{u_a^{k,0}, u_a^{k,1}, u_a^k\}$ for each KC k and a scalar μ^q for each specific assignment q, representing its difficulty level. For each KC k, these embeddings are defined as in HKT to separately reflect KC-specific correct/incorrect interactions and KC identity. However, AKT combines these representations with three additional embeddings $\{u_b^{k,0}, u_b^{k,1}, u_b^k\}$, in order to account for difficulty levels. When a learner interacts at time $t_{n'}$ with an assignment q related to KC k, the KC identity embedding becomes $u^k = u_a^k + \mu^q u_b^k$; after assessing the performance at time $t_{n'}$, the interaction embeddings are similarly updated as $u^{k,y_{n'}} = u_b^{k,y_{n'}} + \mu^q u_b^{k,y_{n'}}$. Consequently, a learner's entire interaction history $\mathcal{H}_{n'<n}$ is represented as a sequence of these combined KC-interaction-difficulty embeddings. AKT processes these sequential embeddings as input, using KC embeddings as queries and keys, and interaction embeddings as values within its attention mechanism. To predict performance $\hat{y}_n$ given the KC and assignment, AKT uses the KC embeddings at time t_n to compare with previous queries and keys in the learning history, and then extract the value. Details about the transformer architecture can be found in Ghosh et al. (2020).

g) GKT (Nakagawa et al., 2019) applies a graph neural network to leverage the graph-structured nature of knowledge. Like AKT, GKT initializes three embeddings $\{u^{k,0}, u^{k,1}, u^k\}$ for each KC k, but instead of only using the embeddings to determine the KC relations, GKT learns an additional undirected KC graph, represented by its adjacency matrix $\boldsymbol{A}$. Here $a_{ij} = 1$ represents KC i and KC j are related, i.e., there is information transmission among KC i and KC j every time the model gets updated. To use the KC relations, GKT first aggregates the hidden states $\boldsymbol{h}_n^k$ and embeddings for the KC reviewed at time t_n, k and its neighboring KCs i:

$$(\boldsymbol{h}_n^k)' = \begin{cases} \left[\boldsymbol{h}_n^k, u^{x_n, y_n}\right] & (i = k) \\ \left[\boldsymbol{h}_n^k, u^i\right] & (i \neq k \text{ with } a_{ik} = 1) \end{cases}$$

After aggregating the information from the neighboring KCs, GKT updates the hidden states based on the aggregated features and the graph structure:

$$\boldsymbol{h}_{n+1}^k = \begin{cases} f_\theta(\boldsymbol{h}_n^{k,\prime}, \boldsymbol{h}_n^k) & (i = k) \\ f_\theta((\boldsymbol{h}_n^k)', (\boldsymbol{h}_n^i)', \boldsymbol{h}_n^k) & (i \neq k \text{ with } a_{ik} = 1). \end{cases}$$

Finally, GKT uses an MLP layer to predict the probability of a correct answer at the next time step, $\hat{y}_{n+1}^k = f_\theta(\boldsymbol{h}_{n+1}^k)$ where $k = x_{n+1}$.

h) QIKT (Chen et al., 2023) focuses on assignments together with KCs, where multiple different assignments can test one KC. Inspired by item-response theory (IRT) (Lord, 2012), QIKT defines three modules, each parameterized by a neural network, to infer interpretable features, namely assignment-specific knowledge acquisition α_n, assignment-specific problem-solving ability ζ_{n+1}, and assignment-agnostic but KC-specific knowledge mastery β_n. Apart from neural network parameters, QIKT learns three sets of assignment-specific embeddings $\{v^{q,0}, v^{q,1}, v^q\}$, which have the same purpose as the KC embeddings defined in AKT, namely for correct interactions, incorrect interactions, and assignment identity. Furthermore, another set of KC-specific embeddings u^k is learned for KC-specific features. QIKT uses LSTM and sum pooling to learn the three features based on each learner's history, $\alpha_n := f_\theta(V_{1:n}, U_{1:n}), \beta_n := f_\theta(U_{1:n})$, where $V_{1:n}$ and $U_{1:n}$ denote respectively the all the assignments and KC embeddings in the learning history. The problem-solving ability $\zeta_{n+1} := f_\theta(V_{1:n+1}, U_{1:n+1})$ is learned by including the assignment and KC information in the coming interaction. To predict performance, all three features are aggregated and input into the sigmoid function, $\hat{y}_{n+1} = \sigma(\alpha_n + \beta_n + \zeta_{n+1})$.

A.3.2 DATASETS

Here we describe the datasets that we have used for evaluation (Assist12, Assist17 and Junyi15), articulate the reasons for their selection, and discuss some of the limitations derived from this choice.

	Assist12		Assist17		Junyi15	
	All	> 50	All	> 50	All	> 50
# Interactions	6,123,270	2,431,788	942,816	942,489	25,925,992	23,907,121
# Learners	46,674	12,443	1,709	1,697	247,606	77,655
# Assignments	179,999	51,866	3,162	3,162	5,174	6,174
# KCs	265	263	102	102	722	721
KC Examples	Rounding		substitution		matrix_basic_distance	
	Unit Rate		fraction-division		circles_and_arcs	
	Perimeter of a Polygon		prime-number		arithmetic_means	

Table 7: Overview of Educational Datasets Assist12, Assist17, and Junyi15, including the number of interactions, learners, assignments, and KCs from overall log data (All) and the log data including learners with more than 50 interactions (> 50).

Description of the selected datasets Assist12 and Assist17 are two subsets of the ASSISTments dataset released by Worcester Polytechnic Institute (Selent et al., 2016). ASSISTments is an online educational tool widely used in U.S. mathematics classes for learners from grades 4 to 12. Predominantly, its users are middle school students (grades 6-8) from Massachusetts or its vicinity. This platform is used for both classroom and homework assignments, and can be used with or without accompanying paper materials. One of the key features of ASSISTments is the immediate feedback provided to students after they answer a question, allowing them to promptly know whether their response was correct.

Junyi Academy is a non-profit Chinese online education platform. Their Junyi15 data release reports the interactions of more than 72,000 students solving mathematics assignments over a year, totaling 16 million attempts. These interaction logs are provided along with two commissioned annotation sets ('expert' and 'crowd-sourced') concerning the structure of 837 KCs in the curriculum. Expert annotations, provided by three teachers, consist of 553 identified prerequisite relations. Crowd-sourced annotations, provided by 51 graduates from senior high school or higher, consist of both prerequisite and similarity evaluations for 1954 KC pairs, with each relation strength rated on a scale from 0 to 9 by at least 3 workers.

We report the numbers of learners, KCs, assignments, and interactions for each dataset in Table 7. In Figure 6 we complement this basic characterization with histograms of the number of per-learner interactions, KCs, and assignments, as well as histograms of elapsed time between learner interactions with arbitrary KCs as well as between interactions with the same KC.

Criteria for dataset selection In order to empirically test our model of learning in structured domains, we sought datasets from domains with a clear prerequisite structure that provide (1) identifiable KC labels, and (2) interaction times with sufficient temporal resolution. In domains where prerequisite relations between KCs are strong, the correct learning order is key for performance, so that performance data be used to uncover structural relations. Additionally, the dependencies in these domains can be identified independently by human annotators, which we use to validate model inferences about the knowledge structure.

1. Identifiable KC labels. Some datasets do not identify the specific KC reviewed at an interaction, but rather a more general assignment or task that could involve multiple unspecified KCs. While this assignment structure can be explicitly modeled (e.g. our baselines AKT, HKT, and QIKT), and we do intend to extend our model in future work to cover this setting, here we intentionally avoided modeling assignment features and concentrated directly on the underlying KCs and their dependencies, which requires KC identities.
2. Timestamped interactions with high temporal resolution. A resolution in the order of seconds or less is essential to adequately track the initial phases of the forgetting process, and to model structural influences that depend on the precise order of KC presentation (see Eq. 5).

Following these criteria, we had to exclude the Statics2011 dataset due to a lack of identified KCs. The Assistments2009 and Assistments2015 datasets lack timestamps entirely, while the 15-minute temporal resolution of the Junyi20 dataset is too coarse for our purposes. This leaves us with Assist12, Assist17 and Junyi15 as appropriate choices to evaluate KT on structured domains. Besides abundant interaction data, Junyi15 provides human-annotated KC relations that, while noisy, offer an invaluable reference to compare the inferred prerequisite graphs to.

Limitations The selection of datasets is limited by design to structured domains, where we can more appropriately put to the test our structure-aware model. We acknowledge that when KCs are largely unrelated (e.g., general knowledge trivia) the inference of prerequisite structure may confer no real advantage. Mathematics, in contrast, provides an ideal testing ground, but more interaction datasets from other domains (e.g., biology, chemistry, linguistics...) and learning stages (primary school, college) are needed for a more representative assessment of the role of structure in learning. In the future, we intend to extend our model to accommodate a broader range of datasets, addressing, in particular, the common case where a single interaction, such as an assignment or a task, is associated with multiple KCs, which entails a more complex interplay of KCs than is displayed in our current dataset selection (Wang et al., 2020).

A.4 PSI-KT MODEL ARCHITECTURE

A.4.1 NETWORK DETAILS

In this section, we introduce the detailed architecture of our PSI-KT model and its hyperparameters. The inference network consists of an embedding network f_ϕ^{Emb}, the cognitive traits encoder f_ϕ^s, and the knowledge states encoder f_ϕ^z. The weights of these interconnected networks collectively constitute the inference parameters ϕ.

Interaction embedding network. The network f_ϕ^{Emb} extracts features from the learning history tuples $\mathcal{H}_{1:N}^\ell = \{x_n, y_n, t_n\}_{1:N}^\ell$, combining information about interaction time, KC identity and performance.

The *KC identity embedding* for KC x_n corresponds to the learned embedding u^{x_n}, which is part of the generative model that parameterizes the graph structure. The *performance embedding* is obtained by expanding the scalar value y_n into a vector $\vec{y}_n$ with the same dimensionality as the time and KC embeddings so that the performance features will be represented on an equal footing. We then concatenate the KC embedding u^{x_n} with the performance embedding $\vec{y}_n$. The *interval embedding* is a positional encoding (Vaswani et al., 2017), $\text{PE}_n = (\sin \alpha(\tau_n); \cos \alpha(\tau_n))$. This embedding approach accommodates intervals spanning different timescales, from minutes to weeks.

Thus, the joint embedding for a learning interaction is given by $v_n = f_{\phi,\text{Emb}}([u^{x_n}; \vec{y}_n]) + \text{PE}_n$, inspired by the transformer architecture (Vaswani et al., 2017).

Latent state encoder. The network f_ϕ^z infers the parameters of the variational posterior distribution $q_\phi(\boldsymbol{z}_{1:n})$. Since learning histories do not have a pre-determined length, we use an LSTM (Hochreiter & Schmidhuber, 1997) as the inference architecture. At each time point, we extract the hidden states in the LSTM, $h_{\boldsymbol{z}_{1:n}} = \text{LSTM}(v_{1:n})$. Meanwhile, in the continual learning setting, information about the history is already encoded and available in the variational parameters for the last time step ϕ_{n-1}, so we use a multi-layer perceptron (MLP), $h_{\boldsymbol{z}_n} = \text{MLP}(v_n)$. Finally, another MLP (similar to the encoder in Kingma & Welling, 2014) takes the hidden states $h_{\boldsymbol{z}_n}$ at every time point as inputs and produces the mean $\mu_{\boldsymbol{z}_n} \in \mathbb{R}^K$ and log-variance $\log \sigma_{\boldsymbol{z}_n} \in \mathbb{R}^K$ for knowledge states $\boldsymbol{z}_n$.

Latent trait encoder. The network f_ϕ^s infers the parameters of the variational posterior distribution $q_\phi(s_{1:n})$. The resulting approximate posterior distribution enables the sampling of learner-specific traits to facilitate personalized predictions. One immediately obvious approach is to use the same architecture of f_ϕ^z. However, the unimodal Gaussian prior over the latent variables cannot account for the diversity of cognitive trait combinations that we expect to find across learners in diverse cohorts. What we need is to allow for multimodality in the distribution of s over all learners.

There is work on factorizing the joint variational posterior as a combination of isotropic posteriors, using a mixture of M experts (MoE; Shi et al., 2019), i.e., $q_\phi(s_{1:n}^\ell \,|\, \mathcal{H}_{1:n}^\ell) = {}^1\!/\!_M \sum_m q_{\phi_m}(s_{1:n}^\ell \,|\, \mathcal{H}_{1:n}^\ell)$, assuming the different modalities are of comparable complexity. However, this may lead to over-parameterization. Instead, inspired by Dilokthanakul et al. (2016), we opt for a mixture of Gaussians as a prior distribution that generalizes the unimodal Gaussian prior and provides multmodality. By assuming that the observed data arises from a mixture of Gaussians, determining the category of a data point becomes equivalent to identifying the mode of the latent distribution from which the data point originates. This approach allows us to partition our latent space into distinct categories. With these discrete variables, it is no longer possible to directly apply the reparameterization trick.

Table 8: PSI-KT architecture and hyperparameters. FC(a, b) represents a fully connected layer with input dimension a and output dimension b; K represents the number of KCs, different across datasets; C represents the number of categories in the mixture of Gaussians for s (we use $C = 10$ in our experiments); the semicolon ; separates connected layers, while the slash / separates the layer architecture for inference on entire histories from the continual learning set-up, where different.

	Inputs & Dim	Hidden Layers	Outputs
f_ϕ^{Emb}	KC Emb & 16 Perf Emb & 16	FC (32, 16) LeakyReLU(0.2) FC (16, 16)	v_n
f_ϕ^{z}	v_n & 16	LSTM (16, 32) / FC (16, 32) FC (32, 16); LeakyReLU(0.2) FC (16, 16); LeakyReLU(0.2) FC (16, K); FC (16, K)	$\mu_{\boldsymbol{z}_n}, \log \sigma_{\boldsymbol{z}_n}$

f_ϕ^{s}	v_n & 16	LSTM (16, 32) / FC (16, 32)
		FC (32, 16); LeakyReLU(0.2)
		FC(16, 64); LeakyReLU(0.2)
		GumbelSoftmax(FC(64, C))
		FC(32 + C, 64); LeakyReLU(0.2)
		FC(64, 16); LeakyReLU(0.2)
		FC(16); FC(64, 4)
			$\mu_{s_n}, \log \sigma_{s_n}$

To solve this inference challenge, we modify the standard VAE architecture by incorporating the Gumbel-Softmax trick (Jang et al., 2016). We employ an LSTM network, taking history embeddings $v_{1:n}$ as inputs and generating one of C category labels through the Gumbel-Softmax technique, denoted as $w = \text{LSTM}(v_{1:n}) \in \text{Cat}(\boldsymbol{\pi})$. Here $\text{Cat}(\boldsymbol{\pi})$ represents the categorical distribution with probabilities $\boldsymbol{\pi} \in \Delta^C$. Simultaneously, we capture hidden states at each time point as $h_{z_{1:n}}$. Subsequently, we utilize an MLP to process both the category label and hidden states as input, producing the mean $\mu_{s_n} \in \mathbb{R}^4$ and log-variance $\log \sigma_{s_n} \in \mathbb{R}^4$ of latent states s_n for each time point.

Table 8 presents an overview of the PSI-KT model architecture and hyperparameters used for all experiments.

A.5 PREDICTION AND GENERALIZATION EXPERIMENTS DETAILS

A.5.1 WITHIN-LEARNER PREDICTION RESULTS AND TRAINING HYPERPARAMETERS

In our prediction experiments, we employ a supervised training approach. For each learner, the first 10 interactions from their learning history are used for training, with the subsequent 10 interactions used as the test set. To report results, we reserve 20% of the learners as a validation set. We employ the Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of 0.005 and apply gradient clipping with a threshold of 10.0. We use a linear decay schedule for the learning rate, halving it every 200 epochs. Additionally, we maintain a consistent batch size of 32 across models.

In Figure 2 in the main text, we present the average accuracy curves for comparison. For a more comprehensive overview of our training protocols, including accuracy, F1-score, and their standard deviation across 5 random seeds, please refer to the detailed results provided in Appendix Tables 9 and 10.

In our baseline models, the original approach was to predict a single time point in the future using all available historical data. However, we believe that relying solely on short-term predictions is insufficient for capturing long-term trends in learners' performance, which is crucial for making accurate recommendations for customized learning materials. Moreover, it's often impractical to assume that we can always access ground-truth data for immediate predictions. Therefore, we predict 10 time points into the future, using the predicted performances as inputs for each step. In other words, instead of using ground-truth data, if the model can predict $\hat{y}_n$ based on all previous training data $y_{n'<n}$, we incorporate the predicted performance along with the historical data $[y_{n'<n}; \hat{y}_n]$ to predict $\hat{y}_{n+1}$.

In the evaluations, we chose to focus on prediction and generalization on a small group of learners, with numbers ranging from 100 to 1,000. This decision is based on the reality that, in educational settings, large datasets are not always available or practical. Additionally, little data is key in practical ITS to minimize the number of learners on an experimental treatment, to mitigate the cold-start

Table 9: Accuracies for within-learner prediction across numbers of learners (mean ± SEM across random seeds).

Dataset	# Learners	HLR	PPE	DKT	DKTF	HKT	AKT	GKT	QIKT	PSI-KT
Assist12	100	$.54_{.03}$	$.65_{.01}$	$.65_{.03}$	$.60_{.01}$	$.55_{.01}$	$\underline{.67}_{.02}$	$.63_{.03}$	$.63_{.03}$	$\mathbf{.68}_{.02}$
	200	$.55_{.02}$	$.63_{.03}$	$.66_{.02}$	$.62_{.01}$	$.58_{.01}$	$\underline{.67}_{.02}$	$.61_{.02}$	$.66_{.02}$	$\mathbf{.70}_{.02}$
	300	$.55_{.01}$	$.66_{.01}$	$.67_{.01}$	$.62_{.00}$	$.58_{.01}$	$\underline{.69}_{.02}$	$.65_{.02}$	$.65_{.02}$	$\mathbf{.71}_{.01}$
	400	$.55_{.01}$	$.65_{.01}$	$\underline{.68}_{.01}$	$.63_{.01}$	$.60_{.02}$	$.67_{.03}$	$.63_{.02}$	$.66_{.01}$	$\mathbf{.71}_{.01}$
	500	$.55_{.01}$	$.64_{.01}$	$\underline{.67}_{.01}$	$.63_{.01}$	$.59_{.03}$	$\underline{.67}_{.02}$	$.63_{.02}$	$.65_{.02}$	$\mathbf{.70}_{.01}$
	1,000	$.54_{.00}$	$.65_{.00}$	$.68_{.01}$	$.63_{.01}$	$.60_{.01}$	$\mathbf{.70}_{.02}$	$.64_{.01}$	$.64_{.01}$	$\mathbf{.70}_{.01}$
Assist17	100	$.45_{.01}$	$.53_{.02}$	$.57_{.02}$	$.53_{.03}$	$.52_{.03}$	$.56_{.02}$	$.56_{.04}$	$\underline{.58}_{.02}$	$\mathbf{.63}_{.02}$
	200	$.45_{.01}$	$.53_{.02}$	$.57_{.02}$	$.54_{.02}$	$.54_{.01}$	$.55_{.01}$	$.56_{.02}$	$\underline{.60}_{.02}$	$\mathbf{.63}_{.01}$
	300	$.46_{.01}$	$.53_{.01}$	$.57_{.02}$	$.55_{.02}$	$.55_{.02}$	$.56_{.04}$	$.58_{.02}$	$\underline{.61}_{.01}$	$\mathbf{.63}_{.01}$
	400	$.45_{.01}$	$.53_{.01}$	$.56_{.01}$	$.57_{.02}$	$.56_{.02}$	$.56_{.02}$	$.58_{.02}$	$\underline{.61}_{.01}$	$\mathbf{.64}_{.00}$
	500	$.46_{.01}$	$.53_{.00}$	$.60_{.01}$	$.58_{.01}$	$.54_{.01}$	$.56_{.02}$	$.58_{.01}$	$\underline{.61}_{.02}$	$\mathbf{.63}_{.01}$
	1,000	$.44_{.01}$	.55.01	$.60_{.01}$	$.57_{.01}$	$.57_{.01}$	$.61_{.01}$	$.60_{.01}$	$\underline{.63}_{.01}$	$\mathbf{.64}_{.00}$
Junyi15	100	$.55_{.02}$	$.66_{.03}$	$.79_{.03}$	$.78_{.01}$	$.63_{.02}$	$\underline{.81}_{.02}$	$.78_{.02}$	$\underline{.81}_{.02}$	$\mathbf{.83}_{.02}$
	200	$.57_{.01}$	$.65_{.03}$	$.79_{.01}$	$.78_{.02}$	$.68_{.03}$	$\underline{.80}_{.01}$	$\underline{.80}_{.01}$	$\underline{.80}_{.01}$	$\mathbf{.84}_{.01}$
	300	$.56_{.02}$	$.65_{.03}$	$\underline{.81}_{.01}$	$.79_{.01}$	$.70_{.01}$	$\underline{.81}_{.01}$	$.78_{.02}$	$\underline{.81}_{.01}$	$\mathbf{.85}_{.01}$
	400	$.61_{.02}$	$.65_{.02}$	$.81_{.01}$	$.80_{.02}$	$.69_{.02}$	$\underline{.82}_{.02}$	$.75_{.02}$	$.80_{.01}$	$\mathbf{.85}_{.01}$
	500	$.61_{.01}$	$.67_{.02}$	$\underline{.82}_{.01}$	$.80_{.02}$	$.70_{.01}$	$\underline{.82}_{.02}$	$.78_{.02}$	$.81_{.01}$	$\mathbf{.85}_{.01}$
	1,000	$.59_{.01}$	$.66_{.02}$	$.81_{.01}$	$.81_{.00}$	$.69_{.01}$	$.82_{.01}$	$.79_{.02}$	$\underline{.83}_{.01}$	$\mathbf{.85}_{.01}$

Table 10: F1 scores for within learner prediction across learner numbers (mean ± SEM across random seeds.)

Dataset	# Learners	HLR	PPE	DKT	DKTF	HKT	AKT	GKT	QIKT	PSI-KT
Assist12	100	$.59_{.02}$	$.77_{.01}$	$.77_{.03}$	$.72_{.01}$	$.64_{.01}$	$\underline{.79}_{.02}$	$.76_{.01}$	$.73_{.03}$	$\mathbf{.80}_{.01}$
	200	$.60_{.02}$	$.74_{.03}$	$\underline{.78}_{.02}$	$.73_{.01}$	$.68_{.01}$	$.76_{.02}$	$.74_{.02}$	$.77_{.02}$	$\mathbf{.82}_{.01}$
	300	$.59_{.02}$	$.77_{.01}$	$\underline{.79}_{.01}$	$.74_{.00}$	$.69_{.01}$	$.73_{.03}$	$.76_{.02}$	$.77_{.01}$	$\mathbf{.83}_{.01}$
	400	$.60_{.02}$	$.77_{.01}$	$\underline{.79}_{.01}$	$.74_{.01}$	$.70_{.03}$	$.73_{.03}$	$.75_{.01}$	$.76_{.01}$	$\mathbf{.83}_{.01}$
	500	$.60_{.01}$	$.76_{.01}$	$\underline{.79}_{.01}$	$.74_{.01}$	$.64_{.10}$	$.74_{.02}$	$.75_{.02}$	$.76_{.01}$	$\mathbf{.82}_{.01}$
	1,000	$.60_{.01}$	$.76_{.00}$	$\underline{.79}_{.00}$	$.74_{.01}$	$.71_{.01}$	$.73_{.02}$	$.76_{.01}$	$.76_{.01}$	$\mathbf{.82}_{.00}$
Assist17	100	$\underline{.45}_{.01}$	$.44_{.01}$	$.42_{.02}$	$.40_{.03}$	$.42_{.03}$	$.40_{.02}$	$.40_{.02}$	$.41_{.02}$	$\mathbf{.48}_{.03}$
	200	$\underline{.45}_{.01}$	$.44_{.01}$	$.40_{.03}$	$.42_{.01}$	$.43_{.01}$	$.44_{.01}$	$.41_{.02}$	$.43_{.02}$	$\mathbf{.47}_{.04}$
	300	$\underline{.45}_{.01}$	$\underline{.45}_{.02}$	$.40_{.02}$	$.41_{.01}$	$.42_{.03}$	$\underline{.45}_{.03}$	$.42_{.01}$	$.44_{.03}$	$\mathbf{.46}_{.03}$
	400	$.44_{.01}$	$.44_{.02}$	$.41_{.01}$	$.42_{.02}$	$.43_{.02}$	$\underline{.45}_{.03}$	$.42_{.02}$	$\underline{.45}_{.01}$	$\mathbf{.47}_{.03}$
	500	$\underline{.46}_{.01}$	$.45_{.01}$	$.40_{.01}$	$.42_{.00}$	$.40_{.10}$	$.45_{.02}$	$.43_{.02}$	$.45_{.01}$	$\mathbf{.47}_{.02}$
	1,000	$.44_{.01}$	$.44_{.02}$	$.40_{.01}$	$.43_{.00}$	$.43_{.03}$	$\underline{.46}_{.02}$	$.43_{.02}$	$\mathbf{.47}_{.01}$	$\mathbf{.47}_{.04}$
Junyi15	100	$.53_{.02}$	$.70_{.03}$	$.88_{.02}$	$.87_{.01}$	$.75_{.03}$	$\underline{.89}_{.01}$	$.87_{.01}$	$\underline{.89}_{.01}$	$\mathbf{.92}_{.01}$
	200	$.54_{.02}$	$.71_{.02}$	$.88_{.01}$	$.87_{.01}$	$.80_{.02}$	$.88_{.01}$	$.88_{.01}$	$\underline{.89}_{.01}$	$\mathbf{.91}_{.01}$
	300	$.53_{.02}$	$.71_{.02}$	$.89_{.01}$	$.88_{.01}$	$.80_{.01}$	$\underline{.90}_{.01}$	$.87_{.02}$	$.89_{.01}$	$\mathbf{.92}_{.01}$
	400	$.52_{.02}$	$.72_{.03}$	$.89_{.01}$	$.88_{.01}$	$.80_{.01}$	$\underline{.90}_{.01}$	$.87_{.02}$	$.89_{.01}$	$\mathbf{.92}_{.02}$
	500	$.53_{.01}$	$.70_{.02}$	$\underline{.89}_{.01}$	$.88_{.01}$	$.74_{.08}$	$.88_{.01}$	$.86_{.01}$	$\underline{.89}_{.01}$	$\mathbf{.92}_{.01}$
	1,000	$.52_{.01}$	$.71_{.02}$	$\underline{.90}_{.01}$	$.89_{.00}$	$.80_{.02}$	$\underline{.90}_{.01}$	$.85_{.01}$	$\underline{.90}_{.01}$	$\mathbf{.93}_{.00}$

problem, and extend the usefulness of the model to classroom-size groups. To provide ITS with a basis for adaptive guidance and long-term learner assessment, we always predict the 10 next interactions.

In order to ensure a fair evaluation of deep learning models and to avoid biasing our results, we expanded our dataset to include over 1,000 learners. This expansion was done post-filtering, where we excluded learners with fewer than 50 interactions. Additionally, 20% of these learners were designated as a validation set. The average accuracy, along with the number of learners and the number of parameters used in each model, is detailed in Table 11.

It's crucial to recognize that deep learning models, despite benefiting from extensive datasets, face specific challenges. Firstly, PSI-KT has remarkable predictive performance when trained on small cohorts whereas baselines require training data from at least 60k learners to reach similar performance. Secondly, the deployment of these deep learning models in real-time applications is challenging due to their substantial number of parameters.

A.5.2 BETWEEN-LEARNER FINE-TUNING HYPERPARAMETERS

For between-learner generalization, we employ pre-trained models from within learner prediction, where the details can be found in Appendix A.5.1. These models are trained using data from 100 learners, and we retain the one that achieved the highest prediction accuracy on the validation set.

Table 11: Accuracy score in within-learner prediction with all learners in each dataset (mean ± SEM across random seeds).

Dataset	# Learners	HLR	PPE	DKT	DKTF	HKT	AKT	GKT	QIKT	PSI-KT
Assist17	1,358	$.46_{.00}$	$.55_{.00}$	$.58_{.01}$	$.55_{.01}$	$.57_{.01}$	$.60_{.01}$	$.60_{.01}$	$\underline{.61}_{.01}$	$\mathbf{.64}_{.00}$
Assist12	9,954	$.44_{.02}$	$.47_{.00}$	$\underline{.69}_{.00}$	$.66_{.00}$	$.66_{.00}$	$68_{.01}$	$\mathbf{.70}_{.00}$	$.68_{.00}$	$\mathbf{.70}_{.01}$
Junyi15	62,124	$.65_{.01}$	$.71_{.01}$	$.85_{.00}$	$.85_{.00}$	$.84_{.01}$	$\mathbf{.86}_{.01}$	$\mathbf{.86}_{.02}$	$\mathbf{.86}_{.00}$	$.85_{.01}$

Then, predictions are made by randomly selecting 100 learners from the group that were not included in the training or validation sets.

In the experiment without fine-tuning, we directly apply the pre-trained models to unseen out-of-sample learners and present the results in Table 2. This entails using the pre-trained models to predict the next 10 interactions for out-of-sample learners based on their first 10 interactions as input.

In the fine-tuning experiment, we perform fine-tuning for each model using a batch size of 32. Additionally, we also set aside 20% of the learners as a validation set during this process to save the model that achieves the highest accuracy after fine-tuning. For baseline models, HLR, PPE, and HKT, which are comprised entirely of learner-independent and KC-dependent parameters, we conduct fine-tuning for all of these parameters. In this scenario, we use the pre-trained models as the initial weight values for the fine-tuning process. Conversely, for models DKT, DKTF, and AKT, we perform fine-tuning specifically on their KC embedding parameters and the last fully connected layer within the neural network, while keeping the remaining layers frozen during the fine-tuning process.

A.5.3 CONTINUAL-LEARNING RESULTS

Table 12: Continual learning accuracy. We report accuracy in predicting 10 subsequent outcomes. # Data indicate the number of interactions from each learner for training.

Dataset	# Data	10	20	30	40	50	60	70	80	90
Assist12	HLR	$.54_{.03}$	$.57_{.08}$	$.58_{.08}$	$.59_{.09}$	$.57_{.10}$	$.56_{.07}$	$.54_{.07}$	$.55_{.06}$	$.57_{.08}$
	PPE	$.65_{.01}$	$.55_{.07}$	$.53_{.07}$	$.52_{.08}$	$.54_{.06}$	$.57_{.06}$	$.59_{.06}$	$.61_{.06}$	$\underline{.69}_{.04}$
	DKT	$.65_{.03}$	$.66_{.07}$	$.64_{.06}$	$\underline{.68}_{.05}$	$\underline{.69}_{.04}$	$\underline{.66}_{.05}$	$\underline{.66}_{.05}$	$\underline{.68}_{.03}$	$.65_{.01}$
	DKTF	$.60_{.01}$	$\underline{.67}_{.04}$	$\underline{.65}_{.04}$	$.64_{.04}$	$.66_{.03}$	$.62_{.06}$	$.61_{.04}$	$.63_{.02}$	$.63_{.02}$
	HKT	$.55_{.01}$	$.56_{.05}$	$.62_{.04}$	$.62_{.05}$	$.63_{.02}$	$.60_{.02}$	$.61_{.03}$	$.61_{.02}$	$.62_{.02}$
	AKT	$.67_{.02}$	$.66_{.04}$	$.62_{.04}$	$.61_{.04}$	$.61_{.05}$	$.65_{.02}$	$.62_{.02}$	$.61_{.02}$	$.63_{.02}$
	GKT	$.65_{.02}$	$.62_{.02}$	$.62_{.01}$	$.64_{.05}$	$.65_{.04}$	$.65_{.03}$	$\underline{.66}_{.06}$	$.65_{.05}$	$.65_{.05}$
	QIKT	$\mathbf{.70}_{.02}$	$.63_{.01}$	$.64_{.02}$	$.63_{.01}$	$.62_{.03}$	$.62_{.01}$	$.62_{.02}$	$.62_{.02}$	$.63_{.01}$
	PSI-KT	$\underline{.68}_{.02}$	$\mathbf{.70}_{.03}$	$\mathbf{.68}_{.03}$	$\mathbf{.72}_{.03}$	$\mathbf{.75}_{.02}$	$\mathbf{.73}_{.03}$	$\mathbf{.74}_{.02}$	$\mathbf{.74}_{.02}$	$\mathbf{.74}_{.02}$
Assist17	HLR	$.45_{.01}$	$.46_{.07}$	$.45_{.07}$	$.53_{.06}$	$.55_{.08}$	$.57_{.06}$	$.55_{.06}$	$.55_{.04}$	$.54_{.03}$
	PPE	$.53_{.02}$	$.52_{.06}$	$.52_{.06}$	$.52_{.07}$	$.52_{.06}$	$.52_{.05}$	$.51_{.05}$	$.54_{.04}$	$.56_{.04}$
	DKT	$.57_{.02}$	$.52_{.05}$	$.52_{.05}$	$.52_{.06}$	$.59_{.04}$	$.57_{.05}$	$.60_{.04}$	$\mathbf{.63}_{.02}$	$.59_{.03}$
	DKTF	$.53_{.03}$	$.58_{.05}$	$.54_{.05}$	$\underline{.58}_{.05}$	$.58_{.04}$	$.55_{.05}$	$.56_{.05}$	$.56_{.04}$	$\underline{.61}_{.02}$
	HKT	$.52_{.03}$	$.57_{.04}$	$\underline{.60}_{.03}$	$\mathbf{.60}_{.03}$	$\mathbf{.62}_{.02}$	$\underline{.61}_{.03}$	$\underline{.61}_{.02}$	$.60_{.02}$	$\underline{.61}_{.02}$
	AKT	$.56_{.02}$	$.53_{.05}$	$.52_{.06}$	$.54_{.04}$	$.53_{.04}$	$.53_{.02}$	$.50_{.03}$	$.51_{.03}$	$.57_{.02}$
	GKT	$\underline{.63}_{.02}$	$\underline{.59}_{.05}$	$.54_{.04}$	$\mathbf{.60}_{.04}$	$.56_{.03}$	$.54_{.02}$	$.57_{.02}$	$.58_{.02}$	$.58_{.03}$
	QIKT	$\mathbf{.65}_{.02}$	$.58_{.03}$	$.59_{.03}$	$.56_{.05}$	$.58_{.03}$	$.56_{.02}$	$.58_{.02}$	$.58_{.01}$	$.56_{.02}$
	PSI-KT	$\underline{.63}_{.02}$	$\mathbf{.62}_{.04}$	$\mathbf{.65}_{.04}$	$\mathbf{.60}_{.05}$	$\underline{.60}_{.05}$	$\mathbf{.62}_{.05}$	$\mathbf{.62}_{.04}$	$\underline{.62}_{.04}$	$\mathbf{.64}_{.03}$
Junyi15	HLR	$.55_{.02}$	$.43_{.06}$	$.42_{.06}$	$.44_{.05}$	$.60_{.04}$	$.63_{.04}$	$.63_{.03}$	$.63_{.04}$	$.64_{.03}$
	PPE	$.66_{.03}$	$.67_{.06}$	$.64_{.06}$	$.64_{.05}$	$.62_{.04}$	$.63_{.05}$	$.60_{.05}$	$.60_{.03}$	$.61_{.03}$
	DKT	$.79_{.03}$	$\underline{.80}_{.04}$	$.78_{.04}$	$.76_{.05}$	$.77_{.04}$	$.75_{.04}$	$\mathbf{.84}_{.04}$	$.73_{.02}$	$.74_{.01}$
	DKTF	$.78_{.01}$	$.74_{.05}$	$.77_{.05}$	$.74_{.06}$	$.71_{.05}$	$.71_{.04}$	$.74_{.03}$	$.71_{.03}$	$.72_{.02}$
	HKT	$.63_{.02}$	$.63_{.08}$	$.69_{.07}$	$.67_{.07}$	$.70_{.04}$	$.73_{.04}$	$.73_{.03}$	$\underline{.79}_{.02}$	$\underline{.84}_{.02}$
	AKT	$.81_{.02}$	$.79_{.04}$	$.78_{.05}$	$\underline{.79}_{.04}$	$.75_{.04}$	$.75_{.03}$	$.76_{.03}$	$.74_{.02}$	$.74_{.03}$
	GKT	$.82_{.01}$	$\underline{.80}_{.02}$	$.78_{.03}$	$.78_{.03}$	$\mathbf{.79}_{.04}$	$\underline{.79}_{.03}$	$.79_{.03}$	$\underline{.79}_{.02}$	$.80_{.02}$
	QIKT	$\mathbf{.84}_{.00}$	$\underline{.80}_{.02}$	$\underline{.80}_{.05}$	$\underline{.78}_{.03}$	$.78_{.04}$	$\mathbf{.81}_{.03}$	$.80_{.02}$	$.78_{.01}$	$\mathbf{.85}_{.02}$
	PSI-KT	$\underline{.83}_{.02}$	$\mathbf{.81}_{.04}$	$\mathbf{.81}_{.04}$	$\mathbf{.80}_{.04}$	$.77_{.06}$	$\mathbf{.81}_{.04}$	$\underline{.82}_{.04}$	$\mathbf{.83}_{.03}$	$\underline{.84}_{.03}$

In this experiment, we randomly select 100 learners using five different random seeds, and collect their first 100 interactions. Initially, the models are trained using only the first 10 interactions, following the same setup as in the within-learner prediction experiment. Following the initial training, we continuously integrate new interaction data into the training process, introducing one interaction at a time for each learner. The model iteratively predicts the subsequent 10 performances. This simulates a common real-world scenario, where learners continually interact with existing or even new KCs.

In the objective function ELBO shown in Eq. 10, all historical information up to time t_n has already been fully encoded into the variational parameters ϕ_n. Additionally, to allow for the

Table 13: Specificity, consistency, and disentanglement.

Metric	Dataset	DKT	DKTF	AKT	QIKT	PSI-KT
Specificity $\mathrm{MI}(s;\ell)\uparrow$	Assist12	**8.83**	6.62	6.45	2.47	$\underline{8.40}$
	Assist17	8.08	7.50	**10.05**	2.95	$\underline{9.98}$
	Junyi15	12.75	$\underline{13.50}$	13.34	4.09	**14.37**
Consistency^{-1} $\mathbb{E}_{\ell_{\text{sub}}}\mathrm{MI}(s^{\ell};\ell_{\text{sub}})\downarrow$	Assist12	14.13	12.24	20.15	$\underline{8.35}$	**7.48**
	Assist17	14.95	13.11	24.47	$\underline{6.35}$	**6.35**
	Junyi15	13.10	17.81	22.15	$\underline{7.66}$	**5.00**
Disentanglement $D_{\text{KL}}(s\|\ell)\uparrow$	Assist12	-1.64	0.38	-8.17	$\underline{2.31}$	**7.42**
	Assist17	-3.01	-0.44	-9.81	$\underline{0.56}$	**8.39**
	Junyi15	-0.62	$\underline{4.96}$	-6.65	1.57	**11.49**

possibility of learners encountering a new KC during their learning journey, we allow for optimization over the KC parameters in the generative model. As a result, when new interaction data $\mathcal{I} = (x_{n+1}, y_{n+1}, t_{n+1})^{1:L}$ becomes available at time t_{n+1}, we use the new data to update both the inference model parameters ϕ_{n+1} and the generative model parameters U and M which are related to KCs.

For the baseline models, which are designed to predict performances based on fixed learning histories, there is no need to update the model parameters for each new data point from each learner individually. Instead, when new interaction data, denoted as $\mathcal{I} = (x_{n+1}, y_{n+1}, t_{n+1})^{1:L}$, becomes accessible at time t_{n+1}, we update all the model parameters using all the interaction data collected up to that point, referred to as $\mathcal{H}_{1:n+1}$. This update is performed through 10 gradient descent processes. It is important to note that we do not include an additional validation set to determine when to stop training each model separately. Instead, we aim for a fair comparison among all models, ensuring that they are trained on equal footing with the same limited data and resources available to them.

A.6 LEARNER-SPECIFIC REPRESENTATIONS ANALYSIS

In this experiment, we examine temporal latent features (learner representations) derived from baseline models. When considering baseline models, it is noteworthy that only DKT, DKTF, AKT, and QIKT incorporate learner-specific temporal embedding vectors. While HKT utilizes temporal embeddings, all these embeddings originate from global parameters associated with KCs, rendering them non-learner-specific. Consequently, our comparative analysis focuses exclusively on PSI-KT compared with DKT, DKTF, AKT, and QIKT.

We initially present comprehensive results in Table 13, complementing Table 3 from Section 4.3, wherein only the results from the best-performing baseline models are displayed. Subsequent sections will detail the experimental setups and metrics employed.

A.6.1 EXPERIMENTAL SETUP FOR SPECIFICITY

In personalized learning, we assume each learner has a unique cognitive profile shaped by past experiences and educational contexts. Our first step is to connect learner representations s_n^{ℓ} with these inherent learner-specific cognitive traits, i.e., the *specificity* of learners given corresponding representations.

To quantify specificity, we employ mutual information, denoted as $\mathrm{MI}(s;\ell) := \mathrm{H}(s) - \mathrm{H}(s|\ell)$ among all learners, as a measure of the information shared between learner identities and learner representations. The detailed computation of the metric $\mathrm{MI}(s;\ell)$ is outlined as follows:

$$\begin{aligned}\mathrm{MI}(s;\ell) &= \mathrm{H}(s) - \mathrm{H}(s|\ell)\\ &= -\int p(s)\log p(s) - \frac{1}{L}\sum_{\ell}\int p(s|\ell)\log p(s|\ell)\\ &= \frac{1}{2}\big(D(1+\log 2\pi) + \log|\Sigma_s|\big) - \frac{1}{2L}\sum_{\ell}\big(D(1+\log 2\pi) + \log|\Sigma_{s^{\ell}}|\big)\\ &= \frac{1}{2}\big(\log|\Sigma_s| - \frac{1}{L}\sum_{\ell}\log|\Sigma_{s^{\ell}}|\big). \end{aligned} \tag{24}$$

Here Σ_s and $\Sigma_{s^{\ell}}$ are the covariance matrices obtained from fitting learner representations from all L learners or, respectively, a single learner with a Gaussian distribution, and D is the dimensionality of learner representations. In experiments, we begin by randomly selecting 1,000 learners from each dataset and then extracting their first 50 interactions for training. To determine when to stop training effectively, we set aside a validation set of 20% of learners, which amounts to 200 learners in our case. This setup mirrors our approach in the prediction experiment. The metric $\text{MI}(s;\ell)$ is calculated for the learners in the training set. Since our goal here is to evaluate the model's capacity to distill representations s^{ℓ} that uniquely identify learners, there is no need for a test set. Note that the baseline models have higher-dimensional learner representations (16 dimensions in our experiments), potentially allowing them to capture more information.

A.6.2 EXPERIMENTAL SETUP FOR CONSISTENCY

We proceed with a supplementary *consistency* analysis to determine, among the shared information quantified in specificity, whether the learner representations capture intricate learner attributes or merely reflect transient dynamic fluctuations. In the experiment, we split the interaction data of each learner into five separate groups, i.e., subsets. Each subset contains 30 interactions. These specific sizes were chosen to ensure we have both enough learners for robust training and enough interactions in subsets to estimate covariance matrices for our metrics. We thus exclude learners who have engaged in fewer than 150 interactions.

To form subsets, we find out the average presentation time of each KC and assign the KCs to separate subsets, so that the overall average interaction time is as similar as possible across subsets. With this, we aim to wash out, to the extent possible with the limited amount of data, systematic biases in the partition induced by the dependence of learner representations on time.

The mutual information metric $\mathbb{E}_{\ell_{\text{sub}}}\text{MI}(s^{\ell};\ell_{\text{sub}}) := \mathbb{E}_{\ell_{\text{sub}}}\left[\text{H}(s|\ell) - \text{H}(s|\ell_{\text{sub}})\right]$, employed in the consistency experiments, undergoes the a similar derivation process to Eq. 24.

$$\begin{aligned}
\mathbb{E}_{\ell_{\text{sub}}}\text{MI}(s^{\ell};\ell_{\text{sub}}) &= \frac{1}{L}\sum_{\ell}\big(\text{H}(s|\ell) - \frac{1}{5}\sum_{\ell_{\text{sub}}}\text{H}(s|\ell_{\text{sub}})\big) \\
&= \frac{1}{L}\sum_{\ell}\Big(-\mathbb{E}\big[\log\mathcal{N}\left(\mu_{s^{\ell}},\Sigma^2_{s^{\ell}}\right)\big] + \frac{1}{5}\sum_{\ell_{\text{sub}}}\mathbb{E}\big[\log\mathcal{N}\left(\mu_{s^{\ell_{\text{sub}}}},\Sigma^2_{s^{\ell_{\text{sub}}}}\right)\big]\Big) \\
&= \frac{1}{L}\sum_{\ell}\big(\log|\Sigma_{s^{\ell}}| - \frac{1}{5}\sum_{\ell_{\text{sub}}}\log|\Sigma_{s^{\ell_{\text{sub}}}}|\big). \qquad (25)
\end{aligned}$$

We fit each sub-learner separately and quantify the divergence metric $\mathbb{E}_{\ell_{\text{sub}}}\text{MI}(s^{\ell};\ell_{\text{sub}})$ between learners and their sub-learners. A lower value of $\mathbb{E}_{\ell_{\text{sub}}}\text{MI}(s^{\ell};\ell_{\text{sub}})$ suggests a higher degree of consistency, reflecting the difficulty in distinguishing between sub-learners and their corresponding overarching learners given learner representations. Overall, Table 3 shows that the learner representations of PSI-KT provide comparable learner specificity and superior consistency. The lower consistency displayed by baseline models suggests that most of the representational capacity available in their higher-dimensional representations might be spent on capturing learner-unspecific characteristics of the training sample.

A.6.3 EXPERIMENTAL SETUP FOR DISENTANGLEMENT

With the insights gained from specificity, our analysis progresses to evaluating to what extent learner-specific representations, are disentangled. Disentanglement in machine learning has been characterized as the process of isolating and identifying distinct, independent, and informative generative factors of variation in the data (Bengio et al., 2013).

In our disentanglement experiments, we use the same setup for specificity, and compute the discrepancy $D_{\text{KL}}(s\|\ell)$ based on 50 interactions of 1,000 learners. Our approach bears similarity to (Kim & Mnih, 2018), but we relax the unrealistic assumption of independent representations. In real-world scenarios, independence in cognitive attributes is not a priority. To assess how much information about learner identity is present in the covariance across the representation dimension, we use the divergence between full trait-vector entropy and diagonal learner-conditional trait-vector entropy.

Table 14: (regression coefficient, p-value) tuples for performance difference and initial performance across models and latents' dimensions. If there is no significant dimension in one model and dataset ($p > 0.05$), we show the dimension with the highest regression coefficient. Bold values indicate the dimension and the baseline model with the highest statistically significant linear relationship in one dataset, with which we show the regression results in Figures 8 and 9.

Behavioural signature	Dataset	DKT	DKTF	AKT	PSI-KT
Performance difference	Assist12	(-.009, .643)	(.008, .736)	(-.009, .665)	(**.300**, <.001)
	Assist17	(-.008, .758)	(-.029, .304)	(-.001, .957)	(**.556**, <.001)
	Junyi15	(-.003, .853)	(.021, .771)	(.025, .064)	(**.721**, <.001)
Initial performance	Assist12	(.021, .078)	(.039, .008)	(.017, <.001)	(**.544**, <.001)
	Assist17	(.048, .004)	(.038, .030)	(.010, .030)	(**3.705**, <.001)
	Junyi15	(-.025, .034)	(.044, .021)	(.017, <.001)	(**.921**, <.001)

The discrepancy $D_{\text{KL}}(s\|\ell)$ is estimated by the full entropy of representations $\mathrm{H}(s)$ and the diagonal elements of the covariance matrix in the conditional entropy $\mathrm{H}(s|\ell)$

$$D_{\text{KL}}(s\|\ell) := \mathrm{H}(s)_{\text{full}} - \mathrm{H}(s|\ell)_{\text{diag}} = \frac{1}{2}\left(\log|\Sigma_s| - \frac{1}{L}\sum_{\ell}\sum_{i=1}^{D}\log\left(\Sigma_{s|\ell}\right)_{ii}\right). \tag{26}$$

Small non-diagonal elements of the covariance matrix in $\mathrm{H}(s)$ suggest low cross-correlations. This can be interpreted as a form of disentanglement. As illustrated in the third row of Table 3, the representations from PSI-KT consistently exhibit a higher degree of disentanglement across all datasets.

A.6.4 MIXED-EFFECT LINEAR REGRESSIONS IN OPERATIONAL INTERPRETABILITY

Mixed-effects regression extends linear regression to handle data with hierarchical or clustered structures, such as repeated measurements from the same subjects (learners in our case). Taking one of our experiments as an example, we conduct regressions based on $y_n^\ell \sim \tilde{\mu}_n^{\ell,k} + (1 \mid \text{learner})$. Here, y_n^ℓ represents the dependent variable and $\tilde{\mu}_n^{\ell,k}$ is a predictor variable at time t_n. Also, $(1 \mid \text{learner})$ represents the random intercept associated with each learner. This random intercept accounts for variability between learners that cannot be explained by the fixed effect $\tilde{\mu}_n^{\ell,k}$. In other words, it accounts for the fact that different learners might have different biases in their responses, allowing us to capture a more robust estimate of the group-level effect.

For regression calculations, we use the models trained in the prediction experiments, as described in Section A.5. For consistent comparisons with specificity experiments, we opt for models trained on a group of 1,000 learners. This experiment goes beyond a simple sanity check (as in Sec. A.6.1), so we use the testing data. This choice aligns with our objective of using operational interpretability to gain insights and inform future controlled experiments with unseen data. We use pre-trained models, specifically DKT, DKTF, AKT, and our PSI-KT model, selected based on their accuracy scores on the validation data.

To fairly compare with baseline models, we investigate whether any dimensions within the learner representations capture behaviors similar to our interpretable cognitive traits. Thus, we perform regression for each dimension within the learner representations of the baseline models. While Figure 4 in the main paper presents the regression results concerning the dimension featuring the most pronounced correlation among baseline models, we provide a complete list of dimensions that exhibit significant relationships with the behavioral data in Table 14.

Performance decay and forgetting rate To analyze the exponential decay of learner performances over time, we first show the relationship between performance decay Δy_n^ℓ and the raw time difference τ_n^ℓ, which is divided into 10 bins. We select bin centers to ensure an equal number of data points in each bin. This binning approach helps minimize the impact of outliers and ensures a balanced representation of data within each bin. Also, we show the relationship between decay Δy_n^ℓ and the time difference scaled by the corresponding forgetting rate α_n^ℓ at each time point, or each dimension of learner representations in the baseline models. We assume that if the forgetting rate α_n^ℓ is meaningful for each time interval and effectively controls the decay, then the standard error of behavior data Δy_n^ℓ within each bin should be smaller than the error of binning raw time differences.

This indicates that the decay is better described as a function of $\alpha_n^\ell \tau_n^\ell$ than as a function of τ_n^ℓ alone We also compute the standard error for each dimension of learner representations in the baseline models, and we show the dimension v_n^* with the lowest standard error in Figure 8. Then, we perform mixed-effect regression over the exponential term $\exp(-\alpha_n^\ell\, \tau_n^\ell)$ (or $\exp(-v_n^*\, \tau_n^\ell)$ in baselines) to assess how well learner representations predict performance decay (as an exponential function). The results show that at least one dimension in the learner representations groups certain behavioral data and reduces the standard errors. However, none of these dimensions exhibit a statistically significant relationship with the behavioral data.

Initial performance and long-term mean We conducted a mixed-effect regression analysis between the initial performance and the long-term mean, with the results presented in Figure 9. These results indicate that, with the exception of DKT on the Assist12 dataset, at least one dimension in the baseline learner representations predicts initial performance. It is important to note that none of the dimensions exhibit a stronger effect compared to the identified trait $\tilde{\mu}_n^{\ell,k}$ in our PSI-KT. Additionally, we note that embedding dimensions in baselines are trained in a permutation-invariant manner, suggesting that it is not possible for these models to route any particular generative factor of variation in the data (e.g. a behavioral signature) to a specific dimension.

Prerequisite transfer ability and learning variances In our experiments, we sought to correlate two additional cognitive traits – transfer ability γ and learning volatility σ - with behavioral data. This task proved more complex than assessing the forgetting rate and long-term mean because assessing transfer ability requires reliable annotations of prerequisite relations and learning volatility can be connected to many unconstrained factors during the learning process.

Regarding transfer ability, our hypothesis posits that given the identified prerequisite KC i for KC j, a higher transfer ability γ_n^ℓ suggests an increased likelihood of correctly transitioning from one KC i to KC j. We calculate this transition probability $p(j^+ \mid i^+)_n^\ell$ by observing the frequency of correct responses to KC i followed by correct responses to KC j up to a certain time t_n. This implies that learners with greater transfer abilities are more likely to answer questions related to KC i correctly after mastering KC i. However, this approach depends on accurately identifying prerequisite relationships between KCs. Therefore, we utilized the Junyi15 dataset, which includes expert-annotated and crowd-sourced prerequisite graphs, for our regression analysis. For learning volatility σ_n^ℓ, we connect the average squared mean $(\bar{\sigma}^\ell)^2$ for each learner with the variance in their performance $\mathrm{Var}(y_{1:n}^\ell)$.

In Figure 10, we present the results of our mixed-effect regression analyses. Each regression demonstrates a significant relationship. However, due to the sparsity of the expert-annotated graph, we do not have enough data to fit the regression model effectively. Thus we choose to use the crowd-sourcing graphs and consider the edge existence if the edge weight is above 0.5.

A.6.5 VISUALIZATION OF KNOWLEDGE STATES

In this section, we display the curve of inferred knowledge states within the Junyi15 dataset. We chose sequences where the involved skills are linked by established prerequisite relations. Two such prerequisites were identified: 'alternate interior angles' as a prerequisite for 'corresponding angles', and 'number properties terminology' for 'properties of numbers'. These prerequisites were determined based on crowd-sourcing annotations, where the average score for the annotated relation exceeded half. We note that PSI-KT can estimate knowledge states at all times and not just interaction times, which allows us to use natural time in the abscissae and display knowledge states with curves instead of using the discrete color maps common in the KT literature.

A.7 GRAPH INFERENCE ANALYSIS

A.7.1 DETAILS OF THE METRICS FOR GROUND-TRUTH GRAPH COMPARISON

Here we report comprehensive evaluations of the alignment of the inferred graphs with the human-annotated graphs in the Junyi15 dataset under different metrics.

As discussed in Section 4.3.2, the Junyi15 dataset provides two types of graph annotations - crowd-sourced similarity and prerequisite ratings (with 1,954 rated edges), as well as more sparse expert-annotated prerequisite relations (837 edges). We use the following metrics to compare graph representations learned by each model against these annotations:

1. **Mean Reciprocal Rank** (MRR). We compare the inferred graph with the expert-annotated using the MRR, defined as $|K|^{-1}\sum_{i=1}^{|K|}(\mathrm{rank}(i))^{-1}$, where K is the total number of KCs. We compute the rank of each expert-identified prerequisite relation $i \rightarrow k$ in the relevant sorted list of inferred probabilities $\{a^{jk}\}_{j=1}^{K}$ and take the harmonic average.
2. **Jaccard Similarity** (JS) is a classic measure of similarity between two sets, defined as the size of the intersection of set A and set B (i.e., the number of common elements) over the size of the union of set A and set B (i.e., the number of unique elements): $\mathrm{JS}(A, B) = {}^{|A \cap B|}/_{|A \cup B|}$. Here we define the edge sets by thresholding weights at half the scale (0.5 for probability-scaled weights, 5 for the average of the 1-9 crowd-sourced rating).
3. **Negative Log-likelihood** (nLL) of edge weights given crowd-sourced annotations. The crowd-sourced annotations provide multiple 1-9 ratings per node pair. One set of annotations rates the strength of the directed prerequisite relations, whereas the other just rates the undirected similarity of the pair of nodes. We normalize the ratings from 0 to 1 and fit them with a Gaussian distribution. Then we compute the log-likelihood of the inferred edge probability under the Gaussian. The variance of the Gaussian accounts for inter-rater disagreements when comparing a model's inferred edge probability with the mean edge rating.
4. **Linear Regression Coefficient** between edge weights and the causal support (details are in Sec. A.7.3) from node i to node k on correctness of k if having correct interactions on i. We compute the causal support for transitions of every KC pair. However we remove the causal support of pairs of KCs that have only one transition in the dataset to avoid adding noise to our estimate.

A.7.2 QUANTITATIVE COMPARISON RESULTS ON THE JUNYI15 DATASET

Note that the graphs of baselines are based on KC embeddings (as in Sec. A.3), and thus there is no edge directionality. For the baselines that have at least two embeddings for each KC, we can use to compute the directed edges, since one embedding for KC will end up in a symmetric structure adjacency matrix (DKT, DKTF, HKT, AKT). Thus, to conduct a fair comparison with the baseline models, we leniently compute edge weights based on every combination of KC embeddings. For example, in DKT, there are two embeddings $u^{k,0}, u^{k,1} \in \mathbb{R}^D$ representing incorrect interactions and correct interactions on KC k, respectively, and embeddings are shared across all learners. We compute the edge weights a^{ik} based on two different combinations here, both $a^{ik} := u^{i,0\intercal}u^{k,1}$ and $a^{ik} := u^{i,1\intercal}u^{k,0}$, and report the graph with the best results. When extracting undirected graphs, we concatenate all KC embeddings to compute $a^{ik} := (u^{i,0} + u^{k,1})^\intercal(u^{i,0} + u^{k,1})$, in order to reflect all available information from all KC embeddings. In the case of baselines with a single embedding per KC, such as QIKT, or those using a parameterized undirected graph, like GKT, we allow their inferred graphs to be less accurate. This means that for these models, the presence of an edge between two KCs is deemed correct if there is a directed edge from either direction in annotated graphs, without the necessity for these edges to accurately indicate the directionality. We then compute the weights by min-max normalization. This normalization is necessary for computing the log-likelihood, where we also use a threshold of 0.5 to determine whether there is an edge when the comparison requires binary edges.

In Table 5, we show the comparison of ground-truth prerequisite graphs and inferred graphs from our PSI-KT, and the best baseline models on the Junyi15 dataset under the different metrics. These results demonstrate that our inferred prerequisite graph outperforms others when compared with crowd-sourced and expert-annotated graphs under different metrics.

Here we show all of the comparison results, including a comparison of the similarity (undirected) graphs and the prerequisite (directed) graphs on four metrics in Table 15. We do not report MRR ranking scores for similarity graphs because the ground-truth similarity graph does not contain an expert-annotated version.

A.7.3 CAUSAL SUPPORT

Causal induction is the problem of inferring underlying causal structures from data. Here, we use a Bayesian framework (Griffiths & Tenenbaum, 2009; 2005) to infer a singular cause-and-effect relationship between all pairs of KCs, asking how performance on one node influences performance on another, and whether the strength of the causal relationship corresponds to our inferred prerequisite

Table 15: Comparison between ground-truth graphs and inferred graphs in Junyi15 dataset. **pre** indicates evaluation against a prerequisite graph and **sim** evaluation against a similarity graph.

		DKT	DKTF	HKT	AKT	GKT	QIKT	PSI-KT
MRR ↑	expert pre	.0069	.0067	.0074	.0075	.0082	.0073	**.0086**
JS ↑	expert pre	1.46e-3	1.37e-3	1.47e-3	1.44e-3	1.46e-3	1.19e-3	**1.86e-3**
	crowd pre	4.60e-3	4.28e-3	4.66e-3	4.48e-3	3.44e-3	5.21e-4	**9.48e-3**
	crowd sim	5.90e-4	0.00	0.00	5.18e-4	3.43e-3	0.00	**4.66e-3**
nLL ↓	crowd pre	5.735	5.580	6.092	5.677	**3.033**	4.228	4.106
	crowd sim	6.598	4.039	4.042	9.100	9.028	10.622	**2.352**

graph. In this context, we model the relationship between a candidate cause C and a candidate effect E (i.e., a pair of KCs), assuming an ever-present background cause B (i.e., the learner's general ability and the influence of other nodes). The objectives are to determine the probability of a causal relationship between C and E, known as *causal support* (Eq. 27).

In our prerequisite graph scenario, we assume that if KC i is a prerequisite of KC k, the correctness on KC i contributes to the correctness on KC k. This implies the presence of a prerequisite relationship between KC i and KC k, signified by a causal link between their correctness levels. Consequently, for every pair of nodes, a candidate cause C corresponds to performance $y_n^i = 1$ at time t_n, and an effect E corresponds to $y_{n+1}^k = 1$ at time t_{n+1}, with inputs from all remaining nodes relegated to the background cause B.

When examining elemental causal induction, we adhere to the following two-step procedure: i) We establish the nature of the relationship through causal graphical models, and ii) we quantify the strength of the relationship, provided it exists, as a problem of inferring structural parameters. In the subsequent text, C and E variables are denoted using uppercase letters, while their specific instances are represented using lowercase letters. Specifically, c^+ and e^+ indicate the presence of the cause and effect (i.e., correct performance), whereas c^- and e^- signify their absence (i.e., incorrect performance).

Causal graphical models. Causal graphical models are a formalism for learning and reasoning about causal relationships (Glymour et al., 2019). Nodes in the graph represent variables, and directed edges represent causal connections between those variables. To identify whether a causal relationship exists between a pair of variables, we consider two directed graphs denoted Graph 0 $G_{C\nrightarrow E} : B \rightarrow E$ and Graph 1 $G_{C\rightarrow E} : B \rightarrow E \leftarrow C$, as shown in Figure 5b. Thus, $G_{C\nrightarrow E}$ represents the null hypothesis that there is no relationship between C and E (i.e., the effect E can be accounted for by background cause B), while $G_{C\rightarrow E}$ represents the alternative hypothesis that the causal relationship exists.

In our case, the cause C and the effect E are equivalent to KC i and KC k, respectively, for every pair of KCs. The process of inferring the underlying structure between KC i and KC k, whether the learners' behavioral learning history $\mathcal{H}$ are generated by $G_{i\nrightarrow k}$ or $G_{i\rightarrow k}$, can be cast in a Bayesian framework (Griffiths & Tenenbaum, 2009; 2005). *Causal support* quantifies the degree of evidence present in the data $\mathcal{H}$ that favors Graph 1 $G_{i\rightarrow k}$ over Graph 0 $G_{i\nrightarrow k}$:

$$\text{support} = \log \frac{P(\mathcal{H} \mid G_{C\rightarrow E})}{P(\mathcal{H} \mid G_{C\nrightarrow E})} = \log \frac{P(\mathcal{H} \mid G_{i\rightarrow k})}{P(\mathcal{H} \mid G_{i\nrightarrow k})}. \tag{27}$$

Intuitively, the joint presence of the cause and effect, i.e., correctness on KC i followed by correctness on KC k, offers support for a causal link from node i to node k. Conversely, the absence of the cause, i.e., incorrectness on KC i but is followed by correctness on KC k, presents evidence against the notion that KC i is a prerequisite for KC k.

Causal support. Causal graphical models depict dependencies using conditional probabilities. Defining these probabilities entails parameterizing each edge, and this parameterization determines the functional expressions that govern causal relationships.

For Graph 0 $G_{i\nrightarrow k}$ and Graph 1 $G_{i\rightarrow k}$, we define $P_0(y_{n+1}^k = 1 \mid B) = \omega_0$ and $P_1(y_{n+1}^k = 1 \mid y_n^i = 1) = \omega_1$ respectively. In other words, the probability of correctness on KC k given just background causes is ω_0, and the probability of correctness on KC k given previous correctness on KC i is ω_1; and when both prerequisite KC i and background causes are present, they have independent opportunities to produce the effect.

For **Graph 0** $G_{C \nrightarrow E}$, the sole parameter ω_0 denotes the likelihood of the effect being present given the background cause

$$P_0(\mathrm{e}^+ \mid b^+; \omega_0) = \omega_0. \tag{28}$$

The corresponding likelihood for the data $\mathcal{H}$ given Graph 0 $G_{i \nrightarrow k}$ is accomplished by integrating over all possible parameters ω_0 with a uniform prior over ω_0:

$$\begin{aligned} P(\mathcal{H} \mid G_{i \nrightarrow k}) &= \int_0^1 P_0(\mathcal{H} \mid \omega_0, G_{i \nrightarrow k}) P(\omega_0 \mid G_{i \nrightarrow k}) \mathrm{d}\omega_0 \\ &= \int_0^1 \omega_0^{N(\mathrm{e}^+)} (1-\omega_0)^{N(\mathrm{e}^-)} \mathrm{d}\omega_0 \\ &= \mathrm{Beta}(N(\mathrm{e}^+)+1, N(\mathrm{e}^-)+1) \\ &= \mathrm{Beta}(N(y_{n+1}^k = 1)+1, N(y_{n+1}^k = 0)+1). \end{aligned} \tag{29}$$

Here $\mathrm{Beta}()$ is the beta function, and $N(\mathrm{e}^+)$ and $N(\mathrm{e}^-)$ are the marginal frequencies of the effects.

For **Graph 1** $G_{i \rightarrow k}$, the likelihood of the effect is given by:

$$P_1(\mathrm{e}^+ \mid b, c; \omega_0, \omega_1) = 1 - (1-\omega_0)^b (1-\omega_1)^c, \tag{30}$$

where ω_0 again defines the influence of the background cause, and the additional parameter ω_1 defines the influence of the cause. Here b and c are binary, which means if cause C exists, then $c = 1$. We compute the likelihood of the data $P(\mathcal{H} \mid G_{i \rightarrow k})$ by integrating over parameters ω_0 and ω_1. Each parameter value is defined by a prior probability, which when combined with the likelihood of the data, yields a joint posterior distribution over data and parameters for the structure. To determine the observed data likelihood for Graph 1 $G_{i \rightarrow k}$, we have

$$\begin{aligned} P(\mathcal{H} \mid G_{i \rightarrow k}) &= \int_0^1 \int_0^1 P_1(\mathcal{H} \mid \omega_0, \omega_1, G_{i \rightarrow k}) P(\omega_0, \omega_1 \mid G_{i \rightarrow k}) \mathrm{d}\omega_0 \, \mathrm{d}\omega_1 \\ &= \int_0^1 \int_0^1 \prod_{e,c} P_1(e \mid c, \mathrm{b}^+; \omega_0, \omega_1)^{N(e,c)} P(\omega_0, \omega_1 \mid G_{i \rightarrow k}) \mathrm{d}\omega_0 \, \mathrm{d}\omega_1. \end{aligned} \tag{31}$$

Here $N(e,c)$ represents the number of occurrences. To compute $\prod_{e,c} P_1(e \mid c, \mathrm{b}^+; \omega_0, \omega_1)^{N(e,c)}$, we iterate over all possible sets of (e,c). Based on Eq. 30, we get:

$$\begin{aligned} \prod_{e,c} P_1(e \mid c, \mathrm{b}^+; \omega_0, \omega_1)^{N(e,c)} =& P_1(\mathrm{e}^+ \mid \mathrm{c}^+, b^+; \omega_0, \omega_1)^{N(\mathrm{e}^+,\mathrm{c}^+)} P_1(\mathrm{e}^+ \mid \mathrm{c}^-, \mathrm{b}^+; \omega_0, \omega_1)^{N(\mathrm{e}^+,\mathrm{c}^-)} \\ =& (\omega_0 + \omega_1 - \omega_0\omega_1)^{N(\mathrm{e}^+,\mathrm{c}^+)} \omega_0^{N(\mathrm{e}^+,\mathrm{c}^-)}. \end{aligned} \tag{32}$$

While Eq. 31 is not analytically tractable, it can be effectively approximated using Monte Carlo simulations. With uniform priors on ω_0 and ω_1, a reliable estimation of $P(\mathcal{H} \mid G_{i \rightarrow k})$ can be obtained by generating m samples of ω_0 and ω_1 from a uniform distribution spanning the interval $[0,1]$, followed by computation of:

$$\begin{aligned} P(\mathcal{H} \mid G_{i \rightarrow k}) =& \frac{1}{m} \sum_{i=1}^m P_1(\mathcal{H} \mid \omega_{0i}, \omega_{1i}, G_{i \rightarrow k}) \\ =& \frac{1}{m} \sum_{i=1}^m \prod_{e,c} P_1(e \mid c, b^+; \omega_{0i}, \omega_{1i})^{N(e,c)} \\ =& \frac{1}{m} \sum_{i=1}^m (\omega_{0i} + \omega_{1i} - \omega_{0i}\omega_{1i})^{N(\mathrm{e}^+,\mathrm{c}^+)} \omega_{0i}^{N(\mathrm{e}^+,\mathrm{c}^-)} \\ =& \frac{1}{m} \sum_{i=1}^m (\omega_{0i} + \omega_{1i} - \omega_{0i}\omega_{1i})^{N(y_{n+1}^k=1, y_n^i=1)} \, \omega_{0i}^{N(y_{n+1}^k=1, y_n^i=0)}. \end{aligned} \tag{33}$$

	Assist12	Assist17	junyi15
PSI-KT	$.68_{.017}$	$.63_{.015}$	$.83_{.015}$
w/o Graph	$-.04_{.005}$	$-.04_{.002}$	$-.07_{.002}$
w/o Individual traits	$-.03_{.002}$	$-.06_{.006}$	$-.04_{.001}$
w/o Dynamic traits	$-.06_{.001}$	$-.09_{.003}$	$-.03_{.002}$

Table 16: The accuracy in three kinds of ablation study (mean ± SEM across random seeds). We show the accuracy gap compared with the complete PSI-KT model.

A.8 ABLATION STUDY

To thoroughly examine the various elements of PSI-KT, including cognitive traits and the prerequisite graph, we executed three distinct ablation studies:

- Without the graph inference (w/o graph): We omit the graph inference process and the influence of prerequisite KCs on the long-term mean. Essentially, this approach treats each KC independently.
- Without individual cognitive traits (w/o individual): We alter the variational inference network in this scenario to produce a uniform distribution across all learners. This change effectively removes the consideration of individual differences in learners' cognitive traits.
- Without dynamic cognitive traits (w/o dynamics): We remove the dynamic transition distribution over the traits in the generative model. This assumes that each learner has static traits over time.

In Table 16 and Figure 13, we present the results of our three ablation studies. We observed that the significance of each of the three components varied across the datasets. These findings underscore the diversity inherent in educational datasets and simultaneously reinforce the effectiveness of our unified framework.

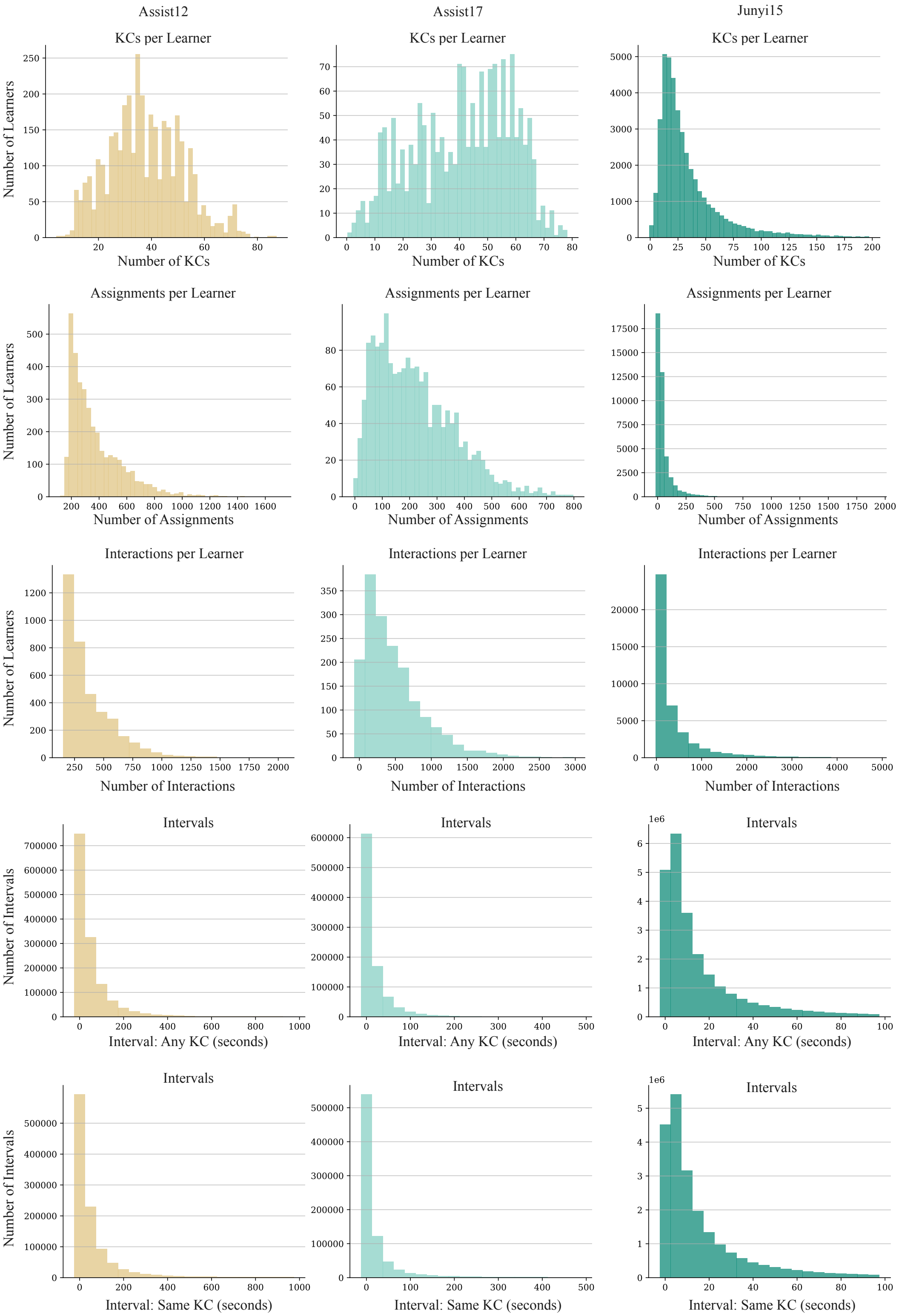

Figure 6: Histograms of key features in three datasets, including the number of interactions, KCs, and assignments per learner, and the intervals between two interactions with any KC and with the same KC.

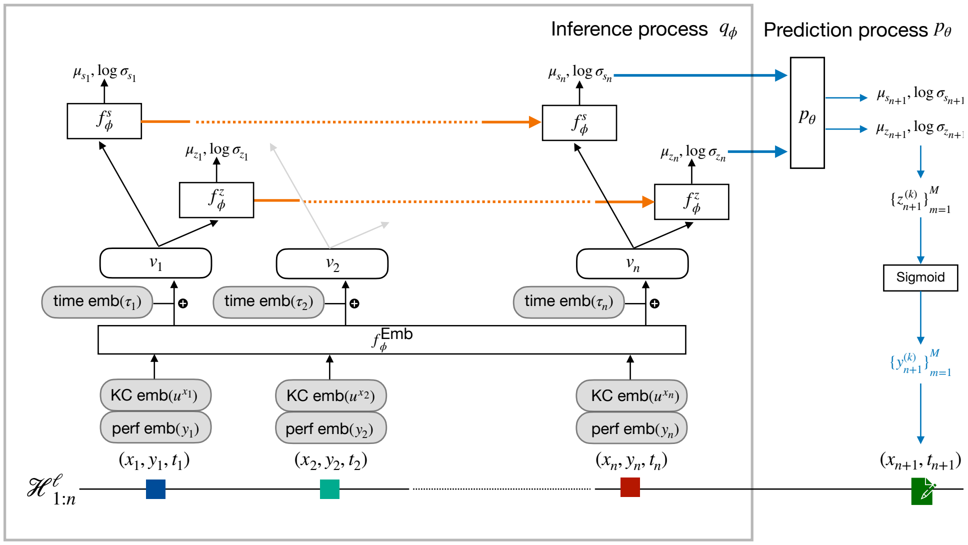

Figure 7: Inference model of PSI-KT with an example of a single learner's history as the input. Note that all parameters ϕ, θ are shared across learners. Grey backgrounds mark inputs. Right rectangles are neural networks' layers and rounded rectangles designate features. Layer f^{Emb}_ϕ maps the input features (time τ, KC u, and performance y, described in the text below) into embedding vectors. Layers f^s_ϕ and f^z_ϕ output the parameters of the variational posterior distribution. These are all part of the inference networks and parameterized by ϕ (surrounded by the grey box). The orange arrow is only applicable for inference on entire learning histories. Blue arrows represent the prediction stage, where during prediction M samples are drawn from the predicted distribution based on $\mu_{z_{n+1}}$ and $\log\sigma_{z_{n+1}}$.

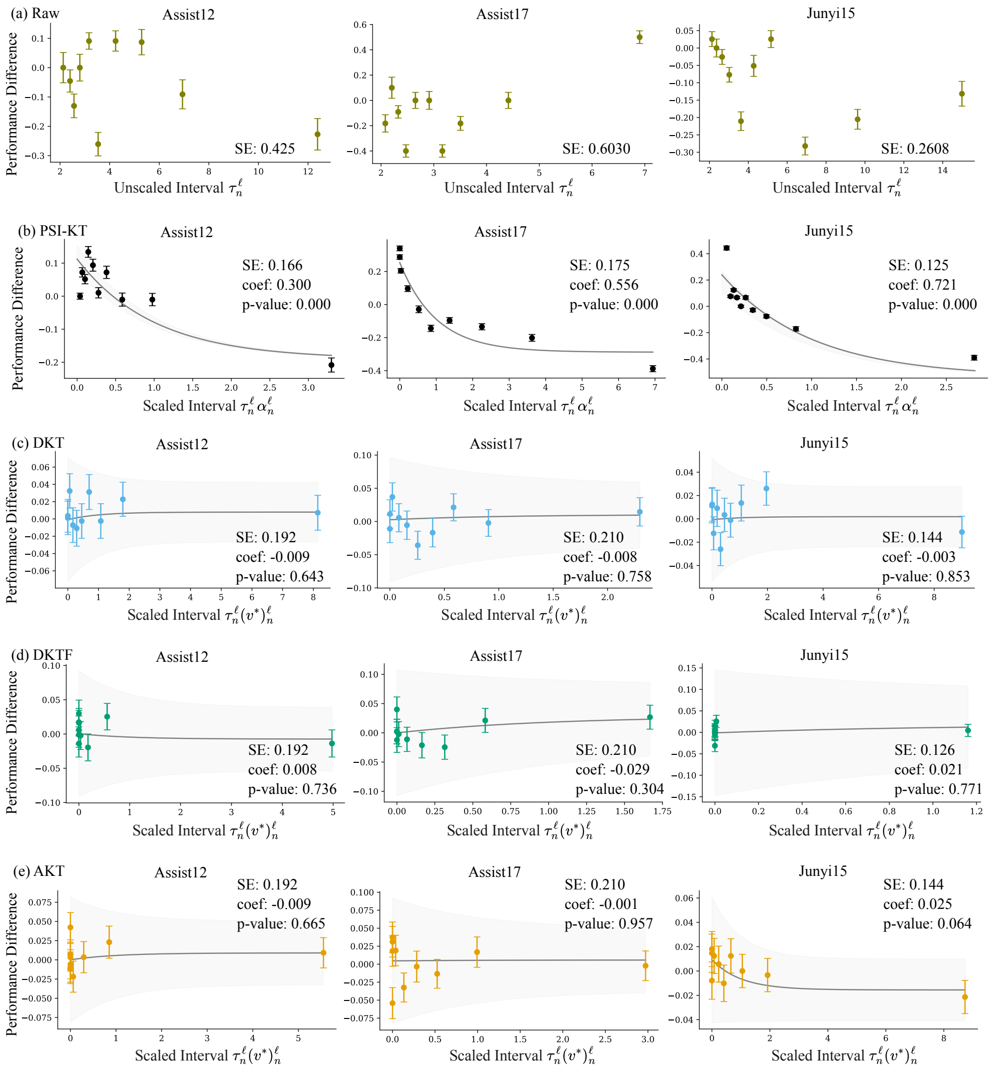

Figure 8: The mixed-effect regressions of performance decay Δy_n^ℓ vs. scaled interval $\tau_n^\ell \alpha_n^\ell$ (scaled interval with the best dimension in baselines $\tau_n^\ell (v^*)_n^\ell$). The first row (a) shows the unscaled interval in the raw data. The aggregate standard error over 10 bins (SE), the regression coefficient (coef), and its p-value are reported in each panel.

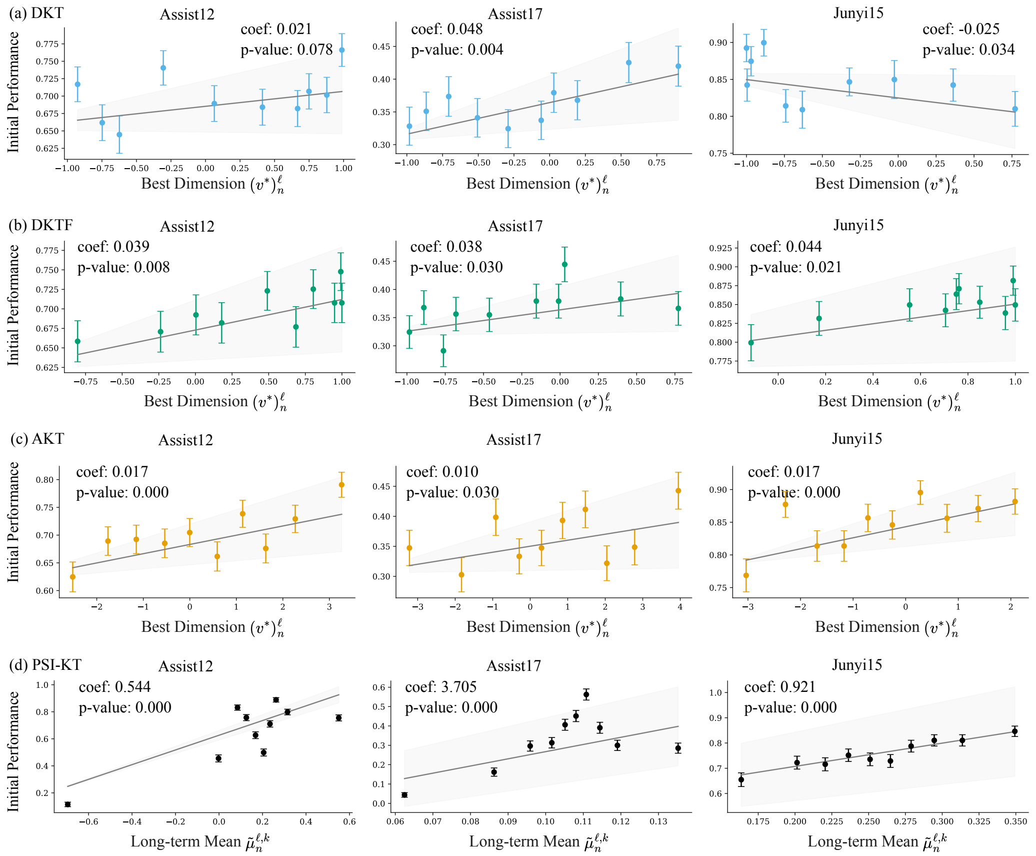

Figure 9: The mixed-effect regressions of initial performance y_n^ℓ vs. long-term mean $\tilde{\mu}_n^{\ell,k}$ (the best dimension $(v^*)_n^\ell$ in baselines). The aggregate standard error over 10 bins (SE), the regression coefficient (coef) and its p-value are reported in each panel.

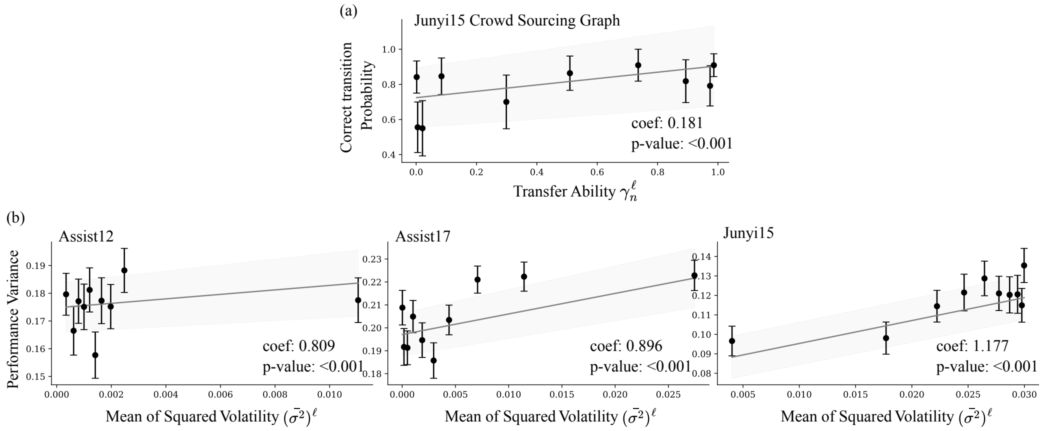

Figure 10: The mixed-effect regressions of transfer ability γ with behavioral correct transition probability (a), and learning volatility σ with the variance in learning performances (b). We report the regression coefficient (coef) and its p-value in each panel, and each point illustrates the mean ($\pm$ SEM) of the corresponding decile.

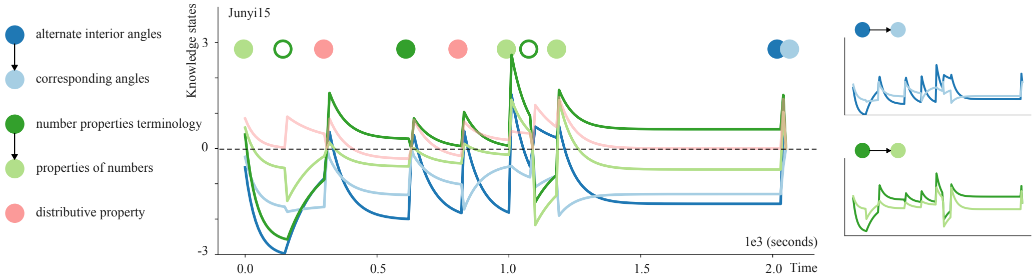

Figure 11: An example of inferred sequential knowledge states in the Junyi15 dataset.

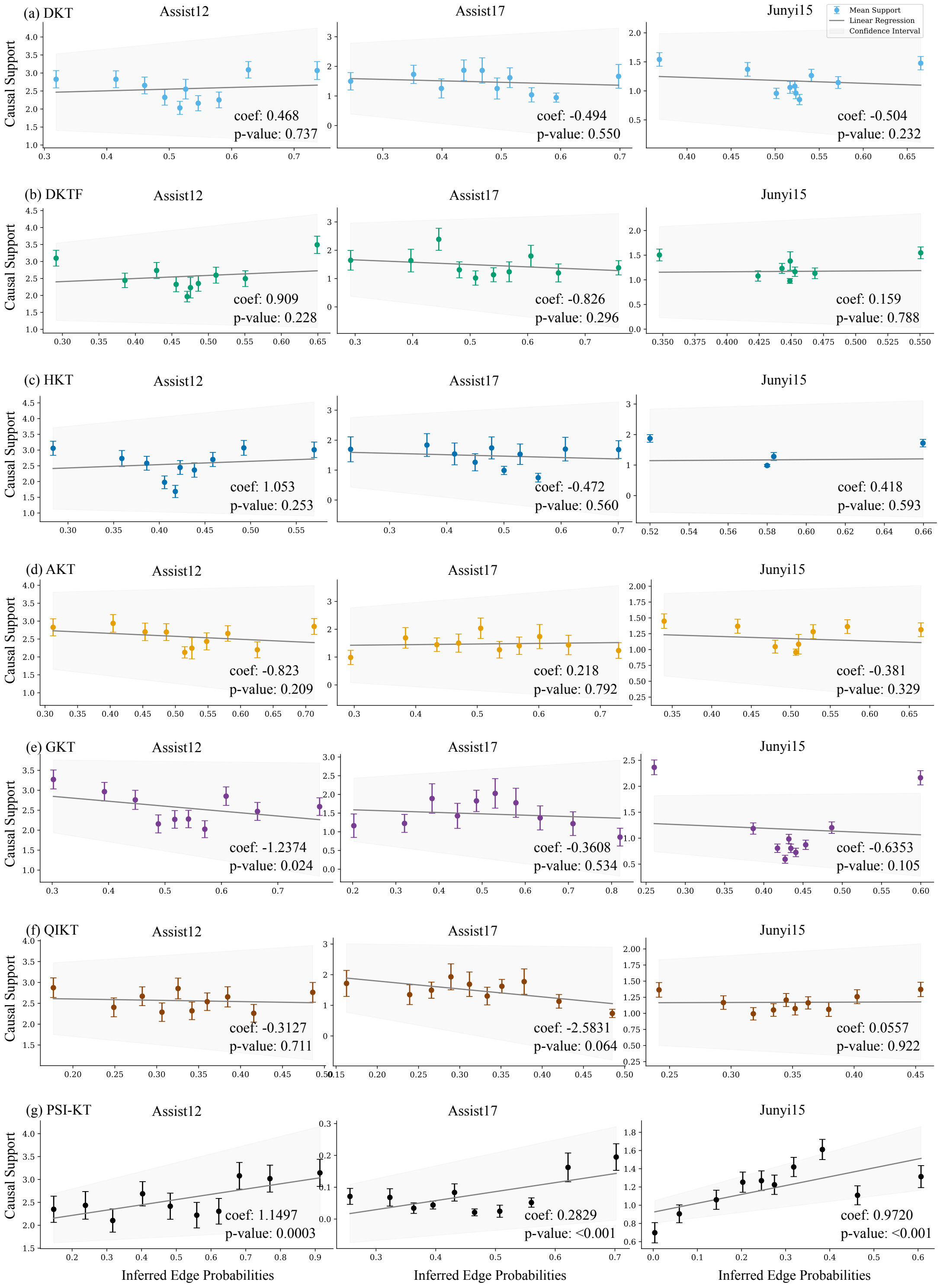

Figure 12: Linear regressions relating causal support to the inferred edges for baseline models *(a)* DKT, *(b)* DKTF, *(c)* HKT, *(d)* AKT, *(e)* GKT, *(f)* QIKT, and *(g)* PSI-KT. The x-axis represents the normalized edge weights inferred by the respective baselines. The coefficient (coef) and its p-value are reported in the lower right of each panel.

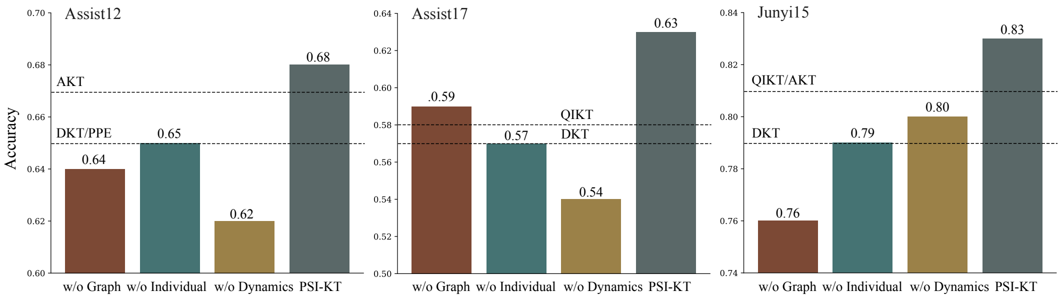

Figure 13: Mean accuracy of PSI-KT vs. ablations of a) the prerequisite structure (w/o graph), b) individualized learner traits (w/o individual) and c) time-dependent learner traits (w/o dynamics). Dashed lines indicate the accuracy of the two best-performing baselines.