Combining Denoised Neural Network and Genetic Symbolic Regression for Memory Behavior Modeling via Dynamic Asynchronous Optimization

Jianwen Sun
sunjw@ccnu.edu.cn
Faculty of Artificial Intelligence in Education, Central China Normal University
Wuhan, China

Qirong Chen
qrchen@mails.ccnu.edu.cn
Faculty of Artificial Intelligence in Education, Central China Normal University
Wuhan, China

Zhenya Huang
huangzhy@ustc.edu.cn
School of Computer Science and Technology, University of Science and Technology of China
Hefei, China

Zhihai Hu
huzhihai@mails.ccnu.edu.cn
Faculty of Artificial Intelligence in Education, Central China Normal University
Wuhan, China

Ruxia Liang
rxliang@ccnu.edu.cn
School of Computer Science, Central China Normal University
Wuhan, China

Xiaoxuan Shen*
shenxiaoxuan@ccnu.edu.cn
Faculty of Artificial Intelligence in Education, Central China Normal University
Hubei, China

Abstract

Memory behavior modeling is a key topic in cognitive psychology and education. Traditional approaches use experimental data to build memory equations, but these models often lack precision and are debated in form. Recently, data-driven methods have improved predictive accuracy but struggle with interpretability, limiting cognitive insights. Although knowledge-informed neural networks have succeeded in fields like physics, their use in behavior modeling is still limited. This paper proposes a Self-evolving Psychology-informed Neural Network (SPsyINN), which leverages classical memory equations as knowledge modules to constrain neural network training. To address challenges such as the difficulty in quantifying descriptors and the limited interpretability of classical memory equations, a genetic symbolic regression algorithm is introduced to conduct evolutionary searches for more optimal expressions based on classical memory equations, enabling the mutual progress of the knowledge module and the neural network module. Specifically, the proposed approach combines genetic symbolic regression and neural networks in a parallel training framework, with a dynamic joint optimization loss function ensuring effective knowledge alignment between the two modules. Then, for addressing the training efficiency differences arising from the distinct optimization methods and computational hardware requirements of genetic algorithms and neural networks, an asynchronous interaction mechanism mediated by proxy data is developed to facilitate effective communication between modules and improve optimization efficiency. Finally, a denoising module is integrated into the neural network to enhance robustness against data noise and improve generalization performance. Experimental results on five large-scale real-world memory behavior demonstrate that SPsyINN outperforms state-of-the-art methods in predictive accuracy. Ablation studies confirm the model's co-evolution capability, improving accuracy while discovering more interpretable memory equations, showing its potential for psychological research. Our code is released at: https://github.com/JiaqiDijon/SPsyINN

CCS Concepts

• **Computing methodologies** → **Machine learning**; *Cognitive science*; • **Applied computing** → **Psychology**; • **Information systems** → **Data mining**.

Keywords

Memory Behavior Modeling, Cognitive Psychology, Knowledge-Informed Neural Networks, Genetic Symbolic Regression

ACM Reference Format:
Jianwen Sun, Qirong Chen, Zhenya Huang, Zhihai Hu, Ruxia Liang, and Xiaoxuan Shen. 2025. Combining Denoised Neural Network and Genetic Symbolic Regression for Memory Behavior Modeling via Dynamic Asynchronous Optimization. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3736886

1 Introduction

Memory is a crucial component of human cognition and a major focus of research in psychology and neuroscience. Memory behavior modeling aims to establish a relationship model between historical memory behavior and memory performance (e.g., the recall probability for specific materials) to elucidate key patterns of human memory behavior, predict performance, and simulate the forgetting process. These models help researchers better understand the

*Xiaoxuan Shen is the corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736886

mechanisms of memory and develop effective memory strategies, offering significant academic and practical value [5].

The earliest memory behavior model dates back to 1885 when Ebbinghaus proposed the forgetting curve [8], suggesting that the relationship between memory performance and time interval follows an exponential function. Subsequently, models such as the generalized power law [37], the adaptive control of thought-rational model [1], and the multi-scale contextual model [24] were introduced. These classical models describe the relationships between memory performance and key memory behavior features (e.g., interval time, repetition frequency, and material difficulty) using mathematical formulas. Derived by experts based on experimental data, these theories lack consensus due to the complexity of memory behavior. Current models often face limitations such as insufficient interpretability, inadequate predictive accuracy, and difficulty quantifying descriptors [3].

Recently, data-driven approaches have emerged for memory behavior modeling. Techniques like machine learning and deep learning have been extensively applied to large-scale memory behavior datasets, resulting in various parametric models [17, 18, 28, 33]. These models exhibit significant advantages in predictive accuracy compared to classical theories. However, their complexity makes them difficult to interpret, offering limited theoretical insights. Moreover, data-driven models demand high-quality and large-scale datasets [27, 36], posing additional challenges [14].

Knowledge-informed neural network models incorporate domain knowledge into neural network construction, enhancing stability and interpretability. These models have achieved remarkable success in natural science tasks [35]. For instance, physics-informed neural networks [26] use known equations and boundary conditions as constraints, reducing data dependency and improving both stability and interpretability. However, their application in memory behavior modeling remains limited, primarily due to the insufficient explanatory power of memory knowledge and difficulties in quantifying descriptors. Existing memory equations are neither as precise nor as universally accepted as physical equations for describing or predicting real-world phenomena. Furthermore, abstract descriptors used in classical memory equations, such as memory strength [37] and word difficulty [15], lack precise formulations, making them challenging to convert into computable variables and complicating knowledge representation.

Based on this analysis, we aim to develop a knowledge-informed neural network model for memory behavior modeling by constraining neural network training using existing memory theory equations to achieve knowledge injection and alignment. To address the limited explanatory power and quantification challenges of classical memory equations, genetic symbolic regression algorithm is introduced. It is initialized with classical memory equations as the population and evolves through mutations to search for improved descriptors and memory equations. Compared to other symbolic regression methods, genetic symbolic regression allows the use of initial equations to fully leverage existing theories and can control equation complexity by limiting symbolic tree depth, ensuring model interpretability. Ultimately, we aim to enable mutual learning and co-optimization between memory equation models and neural networks, enhancing both performance and interpretability.

In this paper, we present a **S**elf-evolving **Psy**chology-**I**nformed **N**eural **N**etwork (SPsyINN), comprising a genetic symbolic regression (GSR) module and a neural network module, with knowledge alignment achieved through interaction and constraint mechanisms. Specifically, we propose a Dynamic Asynchronous Optimization (DAO) method to address dynamic differences during training, including model capability differences and optimization efficiency differences. Model capability differences arise as the GSR module, initialized with classical theories, significantly outperforms the randomly initialized neural network in fitting ability at the start, requiring the neural network to learn more from the GSR module while minimizing its influence on the memory equations. As training progresses, this gap dynamically shifts, so we adjust different training objectives using a dynamic knowledge alignment method to ensure stable optimization. In addition, optimization efficiency differences stem from genetic symbolic regression relying on CPU-based genetic algorithms, while neural networks leverage gradient-based GPU optimization, which is significantly faster. To address this, we introduce a proxy dataset to facilitate asynchronous knowledge transfer, ensuring flexible interactions, and design multiple asynchronous interaction strategies to enable decoupled module training while achieving efficient knowledge alignment, allowing synchronized co-optimization across both modules.

The main contributions of this paper can be summarized as follows.

- To the best of our knowledge, this is the first work to integrate psychological theories into neural networks for memory behavior modeling. We propose SPsyINN, consisting of a genetic symbolic regression module and a denoising neural network module, with knowledge alignment achieved through designed interaction and constraint mechanisms.
- To address the differences in capability and optimization efficiency between modules, we introduce the Dynamic Asynchronous Optimization (DAO) framework. For capability differences, we design a dynamic knowledge alignment method to estimate module performance and adjust alignment strategies dynamically. For efficiency differences, we implement a proxy dataset as a knowledge transfer intermediary and design various asynchronous interaction strategies to ensure flexible and efficient joint optimization.
- We introduced a denoising module to enhance the robustness of the neural network model against data noise, improving the model's stability.
- Comprehensive experiments on four real-world memory behavior datasets demonstrate that SPsyINN outperforms state-of-the-art memory behavior modeling methods across all key metrics, and highlighting its potential for theoretical research and practical applications.

2 Background

Traditional Memory Theory Equations: Memory modeling aims to explain and predict human memory (often referred to as forgetting) behavior using mathematical models. Early psychological studies predominantly relied on controlled experimental paradigms, analyzing data from such experiments to establish relationships

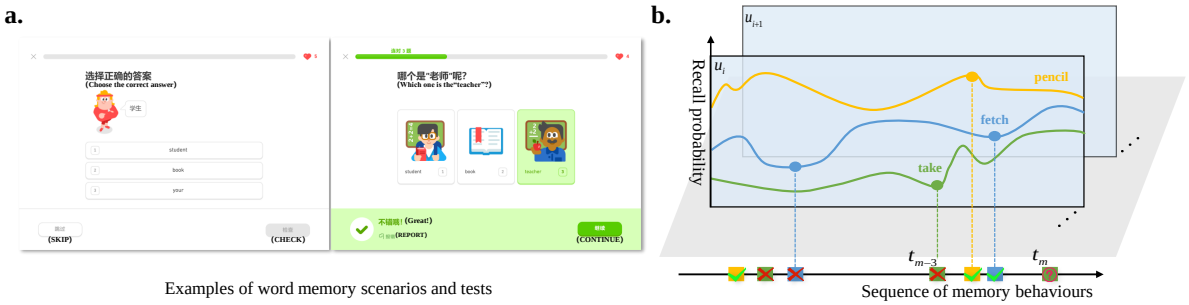

Figure 1: Memory Modeling Scenario: a. Learners engage in vocabulary review using various question types, such as multiple-choice, fill-in-the-blank, and listening exercises, as illustrated by the Word Memorization Software interface. Responses indicate their memorization state: correct answers signify successful retention, while incorrect ones imply incomplete memorization. b. The figure illustrates learners' performance across multiple review tests. Different colored curves represent memory retention trajectories for various words. The horizontal axis tracks testing performance over time, while the vertical axis denotes memory retention rates for specific words.

between memory behavior features and memory performance, typically defined as the recall probability of specific memory materials.

The earliest research on human memory can be traced back to 1885 when Ebbinghaus proposed an approximate forgetting curve equation. He suggested that the interval since the initial memory event is a key factor affecting memory retention, which declines over time at a decreasing rate. This relationship was approximated using an exponential function. Subsequently, researchers explored other reasonable models for memory behavior. In 1974, Wickelgren [37] proposed the generalized power law model ($R = \lambda(1+\beta t)^{-\psi}$), where the recall probability (R) is modeled as a power-law function of initial memory strength (λ), time scale factor (β), forgetting rate (ψ), and time interval (t) since the last memory event. In 1995, Wozniak [39] introduced the dual-component model of long-term memory ($R = e^{-\frac{t}{S}}$), modeling recall probability (R) as an exponential function of memory strength (S) and time interval (t). In 2004, Anderson developed the ACT-R memory model ($R = \beta + \ln(\sum_{k=1}^{N} t_k^{-d_k})$) based on rational adaptation control theory for memory modeling. In 2009, Pashler [24] proposed the MCM model ($R = \sum_{i=1}^{N} \gamma_i exp(-\frac{t}{\tau_i}) x_i(0)$), suggesting that in repeated memory scenarios, memory performance is an aggregate of independent memory curves, similar to Wozniak's exponential model. In 2014, Lindsey introduced the DASH memory modeling method [15]($R = \sigma(a_s - d_c + \sum_{w=1}^{|W|}(\theta_{2w-1} ln(1+c_w) + \theta_{2w} ln(1+n_w))))$, which relates a learner's memory state (R) to their ability (a_s), material difficulty (d_c), attempt counts (c_w), and historical correct recall attempts (n_w). In 2016, the Half-Life Regression (HLR) model introduced the concept of memory half-life to describe the forgetting process of memory materials. Detailed explanations of the variables in these memory equations are provided in Appendix A.

Despite over a century of exploration, researchers have yet to identify a universally accepted memory equation. While theoretical memory equations are concise and interpretable, they have limited explanatory power for memory behavior and insufficient predictive accuracy for memory performance. Furthermore, many theoretical models include abstract psychological descriptors that are difficult to quantify. For instance, in Wozniak's dual-component model ($R = e^{-\frac{t}{S}}$), the descriptor memory strength (S) reflects the depth of impression left by a memory behavior on the learner. However, current research struggles to fully identify the factors influencing memory strength or to provide precise calculation methods, even though it clearly impacts memory performance. In practice, memory strength is often treated as a constant, which is evidently unrealistic. These issues pose significant challenges to building knowledge models based on psychological theories.

Data-driven Parametric Model: The widespread adoption of word memory software has opened new opportunities for memory research. Researchers have utilized data-driven paradigms and machine learning methods to develop parameterized memory behavior models, treating words as knowledge components in knowledge tracing (KT) [2]. This approach integrates memory modeling with KT tasks, driving improvements in both model performance and theoretical insights. Advances in deep learning have further accelerated KT research. Piech et al. introduced the Deep Knowledge Tracing (DKT) model [25], the first to apply Recurrent Neural Networks (RNNs) [16] to KT. DKT captures the temporal dynamics of student interactions with questions to predict responses to new ones, significantly outperforming traditional KT models and highlighting the potential of deep learning in modeling learning behaviors. Subsequent research adopted temporal models like Long Short-Term Memory (LSTM) [18, 32] and Transformer [17], refining model structures [32] and incorporating factors such as difficulty levels [10], review conditions [30], and material relevance [4] to enhance performance. While deep learning-based models excel in data fit and prediction accuracy, their "black-box" nature remains a challenge, limiting interpretability and educational applications. Moreover, building these models requires large-scale, high-quality behavioral data, which is still difficult to obtain.

Physics Informed Neural Networks: In recent years, Physics-Informed Neural Networks (PINNs) have emerged as one of the most successful knowledge-driven neural network models, achieving significant breakthroughs in fields like dynamics modeling [12], fluid mechanics [35], and solving differential equations [21]. Unlike traditional purely data-driven neural networks, PINNs integrate domain-specific physical knowledge with deep learning, offering a novel approach to modeling. The core idea of PINNs is

to embed physical laws (such as conservation laws and boundary conditions) directly into the neural network's loss function, ensuring that predictions and simulations always adhere to physical constraints. This approach not only enhances the physical interpretability of the model but also improves its generalization ability across various scenarios [6], demonstrating substantial potential for scientific computation and engineering applications. For example, the Navier-Stokes equations were applied to analyze the energy extraction efficiency of hydrokinetic turbines, while also improving the high-dimensional design of the turbine blades and ducts [23].

However, in the domain of memory behavior modeling, the exploration of knowledge-informed neural network models remains scarce. Existing memory theory equations find it challenging to describe or predict real-world phenomena with the precision of physical equations. Their mathematical forms are often contentious, making it difficult to offer precise guidance to neural network models. Furthermore, psychological domain knowledge is challenging to express in computational models. Many abstract descriptors introduced in classical memory theory equations are difficult to translate into computable variables, posing significant challenges for knowledge representation.

3 Methodology

3.1 Problem Statement

We aim to develop and validate our approach using a large-scale word memory behavior dataset. Vocabulary Learning scenarios are widely used in memory behavior research [20], and the resulting memory models can guide Word Memorization Software in optimizing repetition strategies. These datasets are derived from real user interaction logs collected through Word Memorization applications. Users engage in word testing tasks provided by the software (as shown in Figure 1**a**), memorize target vocabulary, and retest the words after a certain period to reinforce memory. By analyzing learners' performance across different word tests over time, we can uncover the core patterns underlying their memory evolution and internal mechanisms. Our goal is to build computational models based on learners' historical interaction data from word tests, estimate their memory states for each word, and accurately predict their performance in upcoming tests for specific words (as shown in Figure 1**b**).

Formally, the set of all users in the dataset is denoted as $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$, and the set of all words as $\mathcal{W}$. The dataset encompasses all users' memory test behaviors, represented as $\mathcal{D} = \{\mathcal{D}_{u_1}, \mathcal{D}_{u_2}, \ldots, \mathcal{D}_{u_n}\}$ where each behavior $\mathcal{D}_u$ is defined as:

$$\mathcal{D}_u = \{(w_1, y_1, t_1), (w_2, y_2, t_2), \ldots, (w_m, y_m, t_m)\} \tag{1}$$

where, $\mathcal{D}_u$ represents the memory data of user (u), consisting of a sequence of triples composed of word (w), tag (y), and timestamp (t).

For a specific memory behavior (w, y, t), we use x_u^t to denote the historical memory behavior features of user u at time t, which are derived from all previous behavioral records. These features include six main variables, whose definitions and computational methods are illustrated in Appendix B. Correspondingly, y_u^t represents the performance of user u in word memory testing for word w at time t. Our goal is to construct a memory model f such that $y_u^t = f(x_u^t)$.

3.2 SPsyINN

We propose a Self-evolving Psychology-Informed Neural Network (SPsyINN), combining neural networks and genetic algorithms to design two independent modules: the Denoising Neural Network (DNN) and Genetic Symbolic Regression (GSR). The corresponding memory models are denoted as f_{DNN} and f_{GSR}. Here, f_{DNN} is a parameterized neural network trained via gradient-based optimization, while f_{GSR} is a mathematical function optimized using genetic algorithms, with classical memory theory equations as the initial population.

To align the outputs of f_{DNN} and f_{GSR}, we adopt techniques from knowledge distillation and PINN models, enabling knowledge integration while fitting training data. This chapter introduces the method in three parts: Denoising Neural Network, Genetic Symbolic Regression, and Dynamic Asynchronous Optimization. The first two sections detail the construction of f_{DNN} and f_{GSR}, while the last explains their knowledge alignment and collaborative optimization. The overall framework is illustrated in Figure 2.

3.3 Denoised Neural Network

Neural networks, as universal approximators, have achieved significant success in behavior modeling, with temporal models like LSTM and Transformer widely applied. Our Denoised Neural Network (DNN) module adopts a classical learning behavior prediction architecture, combining a Temporal Neural Network (TNN) with a Multi-Layer Perceptron (MLP) classifier for modeling learners' internal states and classifying their performance. TNN can utilize flexible architectures such as LSTM [11], Transformer [34], Mamba [9], or other specially designed model architectures.

For a learner u with memory behavior data D_u, we concatenate all behavior features as $x_u^{t_{1:m}} = [x_u^{t_1}, x_u^{t_2}, \ldots, x_u^{t_m}]$, with the target memory performance sequence $y_u^{t_{1:m}} = [y_u^{t_1}, y_u^{t_2}, \ldots, y_u^{t_m}]$. The model's output predictions are $\hat{y}_u^{t_{1:m}} = MLP(TNN\left(x_u^{t_{1:m}}, \Theta_{TNN}\right), \Theta_{MLP})$, or, more generally $\hat{y}_u^{t_{1:m}} = f_{DNN}\left(x_u^{t_{1:m}}, \Theta_{DNN}\right)$. The model optimizes its parameters by minimizing the Binary Cross-Entropy Loss (BCELoss) as follows:

$$L_{\hat{D}} = -\frac{1}{|\mathcal{D}|} \sum_{u \in \mathcal{U}} \sum_{i=1}^{m} \left[y_u^{t_i} \log\left(\hat{y}_u^{t_i}\right) + \left(1 - y_u^{t_i}\right) \log\left(1 - \hat{y}_u^{t_i}\right) \right] \tag{2}$$

To address noise in memory behavior data, we design a denoising module by injecting noise into the input features:

$$\tilde{x}_u^{t_{1:m}} = \sqrt{a_m} \cdot x_u^{t_{1:m}} + \gamma \cdot \varepsilon \cdot \sqrt{1 - a_m} \tag{3}$$

where $a_m = \prod_{t=1}^{m}(1 - \beta_t)$ is the cumulative noise schedule, γ is a learnable noise weight, and $\varepsilon \sim N(0, I)$ represents Gaussian noise. This process is consistent with the perturbation kernel used in the Denoising Diffusion Probabilistic Models diffusion process [31]. A detailed proof can be found in Appendix C.

The model's noisy predictions are $\tilde{y}_u^{t_{1:m}} = f_{NN}(\tilde{x}_u^{t_{1:m}}, \Theta_{DNN})$, with the denoising objective to minimize: $L_{\tilde{D}} = \frac{1}{|\mathcal{D}|} \sum_{u \in \mathcal{U}} \sum_{i=1}^{m} (\tilde{y}_u^{t_i} - \hat{y}_u^{t_i})^2$. The DNN module's total training objective combines $L_{\hat{D}}$ and $L_{\tilde{D}}$, while overall optimization details are discussed in the Dynamic Asynchronous Optimization section.

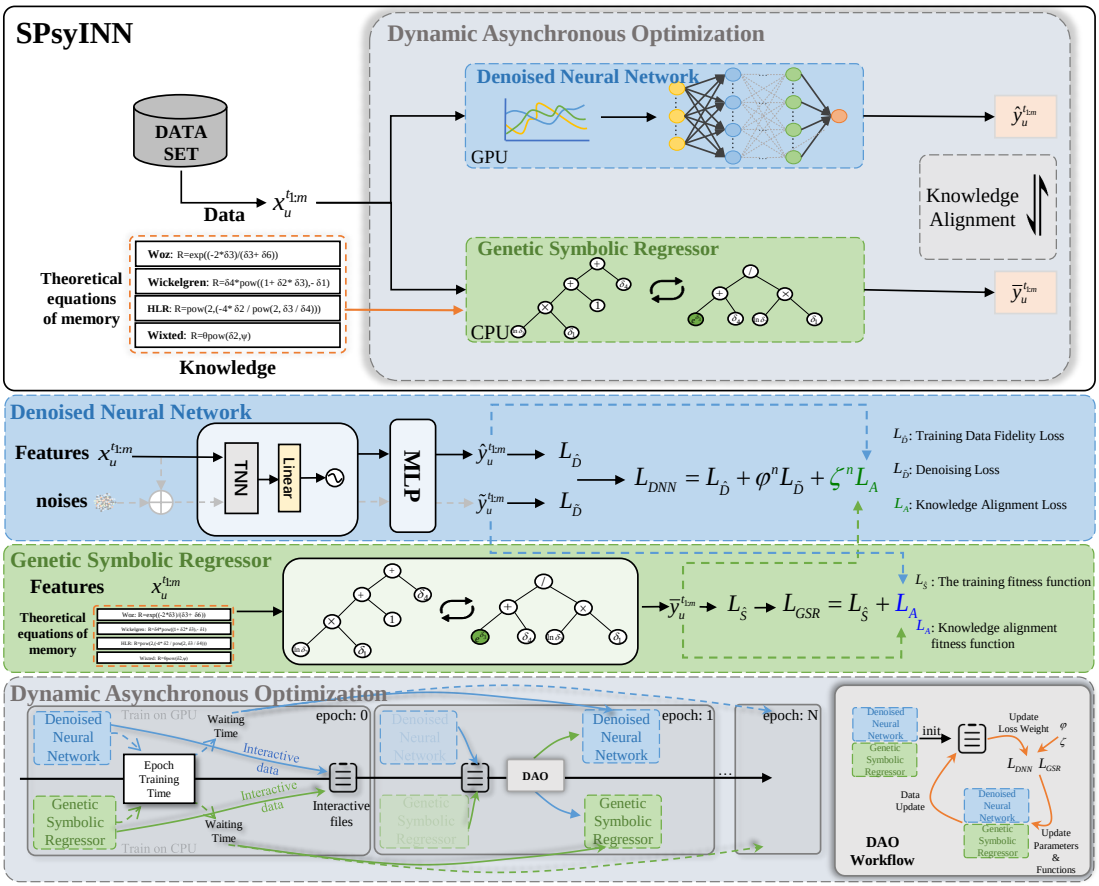

Figure 2: SPsyINN Model Framework Diagram. The blue subfigure describes the training process of the deep learning module; the green subfigure illustrates the training process of symbolic regression; the gray module represents the asynchronous training process.

3.4 Genetic Symbolic Regressor

Genetic symbolic regression (GSR) is a classical symbolic regression algorithm that leverages evolutionary mechanisms of genetic algorithms to search for and optimize mathematical expressions, aiming to generate equations that meet specific requirements. The key steps include initializing a population, evaluating fitness, performing selection, mutation, and crossover operations, and updating the population. To incorporate insights from psychology, we use classical memory theory equations as the initial population for GSR. The predictions from the GSR module are expressed as $\bar{y}_u^{t_{1:m}} = f_{GSR}(x_u^{t_{1:m}}, \Phi, \tau)$, where $\bar{y}_u^{t_{1:m}}$ represents the function values on raw data, $f_{GSR}(\cdot)$ is the optimized function derived from traditional memory equations, $\Phi \in \{+, -, \times, \div, \text{pow}, \exp, \ln\}$ denotes the operator set consistent with classical memory theories, and τ represents the current symbolic tree. The fitness function evaluates the GSR model's predictions and is defined as $L_{\hat{S}} = \frac{1}{|\mathcal{D}|}\sum_{u\in\mathcal{U}}\sum_{i=1}^{m}(\bar{y}_u^{t_i} - y_u^{t_i})^2$, ensuring the searched equations best fit the training data. Our GSR framework is flexible and supports various algorithms (e.g., TPSR [29], DGSR [13]) and libraries (e.g., Eureqa[1], PySR[2], and geppy[3]).

In summary, SPsyINN consists of two modules: a denoised neural network and a genetic symbolic regressor. Each module generates independent memory behavior predictions, namely $\hat{y}_u^{t_{1:m}}$ and $\bar{y}_u^{t_{1:m}}$, respectively. By default, we use $\hat{y}_u^{t_{1:m}}$ (the output of DNN module) as the final output, as the denoised neural network typically achieves better prediction accuracy after training.

3.5 Dynamic Asynchronous Optimization

To align knowledge between the denoised neural network (f_{DNN}) and the genetic symbolic regressor (f_{GSR}) in SPsyINN, we propose the Dynamic Asynchronous Optimization (DAO) method for collaborative training. Knowledge alignment is achieved using the alignment loss L_A, defined as:

$$L_A = \frac{1}{|\mathcal{D}|}\sum_{u\in U}\sum_{i=1}^{m}(\bar{y}_u^{t_i} - \hat{y}_u^{t_i})^2 \tag{4}$$

[1] http://nutonian.wikidot.com/
[2] https://astroautomata.com/PySR/
[3] https://github.com/ShuhuaGao/geppy

Table 1: The overall prediction performance of all baseline models and SPsyINN. The best model performance is in bold and the 2nd best is underlined(Excluding the variants of the proposed SPsyINN). * indicates t-test p-value < 0.05 compared to the 2nd best result. The experimental results for SPsyINN are based on predictions from the DNN module, with the reported values representing the averages of five independent experiments.

Models	En2Es			En2De			Duolingo			MaiMemo		
	PrAUC	G-means	Precision	PrAUC	G-means	Precision	PrAUC	G-means	Precision	PrAUC	G-means	Precision
Wickelgren	.7339	.3378	.7256	.7602	.3331	.7268	.7175	.1657	.6963	.8174	.0924	.6350
ACT-R	.8223	.3621	.6985	.8099	.3427	**.8270***	.8393	.4252	.7355	.8023	.2761	.6339
DASH	.6934	.1259	.6780	.7457	.2623	.7109	.7695	.4422	.7375	.8167	.0671	.6338
HLR	.8449	.3562	.7159	.7763	.3787	.7482	.8409	.4059	.7338	.8135	.2663	.6414
DKT-Forget	.7810	.0411	.7383	.7664	.0821	.7166	.8011	.2515	.7243	.6903	.3272	.6494
FIFKT	.7936	.1356	.7403	.7588	.0819	.7163	.7904	.3264	.7319	.6929	.3486	.6512
SimpleKT	.7994	.1027	.7395	.7795	.0579	.7153	.8067	.3208	.7311	.7245	.2803	.6407
QIKT	.7433	.3621	.7369	.7004	.2169	.7162	.7399	.4003	.7279	.7226	.2803	.6407
MIKT	.8013	.1980	.7436	.7842	.1774	.7191	.8071	.3272	.7320	.7237	.2969	.6415
SPsyINN-C	**.8506***	.4274	.7483	.8323	.4309	.7232	.8451	.4181	.7307	.8187	.2688	.6436
SPsyINN-I	.8477	.4289	.7463	.8324	.4243	.7231	.8414	.4334	.7298	.8182	.3598	.6527
SPsyINN-W	.8497	**.4378***	**.7494***	**.8376***	**.4359***	.7280	**.8547***	**.4516***	**.7421***	**.8191***	**.3956***	**.6584***

where, $\tilde{y}_u^{t_i}$ and $\hat{y}_u^{t_i}$ represent the predictions of the symbolic regressor and the neural network, respectively. During training, a knowledge alignment objective is added on top of the data-fitting objective. This knowledge alignment loss function facilitates mutual learning, allowing weaker modules to benefit more from stronger ones. In the initialization phase, the randomly initialized f_{DNN} primarily learns from f_{GSR}, which is grounded in theoretical equations. However, in later stages, if f_{DNN} outperforms f_{GSR}, the alignment weight should be adjusted accordingly. Therefore, we propose a dynamic training objective adjustment method. The total loss for the neural network is:

$$L_{DNN} = L_{\hat{\mathcal{D}}} + \varphi L_{\tilde{\mathcal{D}}} + \zeta L_A \tag{5}$$

with dynamic weights φ and ζ updated as $\varphi^{n+1} = \frac{L_{\hat{\mathcal{D}}}^n + L_{\hat{\mathcal{S}}}^n}{L_N^n + L_{\hat{\mathcal{S}}}^n}$, $\zeta^{n+1} = \frac{L_{\hat{\mathcal{D}}}^n + L_N^n}{L_N^n + L_{\hat{\mathcal{S}}}^n}$. $L_N = \frac{1}{|\mathcal{D}|}\sum_{u\in\mathcal{U}}\sum_{i=1}^{m}(\tilde{y}_u^{t_i} - y_u^{t_i})^2$ represents the MSE loss of the neural network on noisy data. As $L_{\hat{\mathcal{S}}}$ decreases, indicating improved fitting ability of f_{GSR}, the weight ζ increases, encouraging f_{DNN} to learn more from f_{GSR}. Similarly, as $L_{\tilde{\mathcal{D}}}$, decreases, reflecting improved noise prediction by f_{DNN}, more emphasis is placed on $L_{\tilde{\mathcal{D}}}$. For the symbolic regression module, the total fitness function is fixed as $L_{GSR} = L_{\hat{\mathcal{S}}} + L_A$.

In implementation, f_{DNN} and f_{GSR} are trained as separate processes. For alignment loss computation, a proxy file serves as an intermediary. Predictions from each module for the corresponding batch are stored in this proxy dataset, which is updated after each epoch. Both modules read data from this proxy file to compute alignment loss L_A. The proxy dataset's batch sampling is independent of the training data batch sampling. Its batch size can differ from the training dataset, which, as supported by theoretical proofs (Appendix D), does not affect optimization performance. The overall workflow for DAO is illustrated in Figure 2.

In practical training, f_{DNN} and f_{GSR} often exhibit significant differences in training time per epoch, with DNN typically training much faster than GSR. This makes it crucial to enhance effective data exchange (knowledge alignment) between the modules. Additionally, the efficiency of knowledge alignment must be incorporated into the design, requiring a balance between alignment frequency and effectiveness. Our goal is to enable the two models to optimize synchronously, achieving better knowledge alignment results. Therefore, we have designed three types of asynchronous strategies to adjust the optimization pace between the two modules. **Wait Optimization (SPsyINN-W):** The faster module waits for the slower one to synchronize, ensuring alignment but reducing efficiency. **Continuous Optimization (SPsyINN-C):** Modules optimize independently, maximizing efficiency at the cost of alignment synchronization. **Interval Optimization (SPsyINN-I):** The neural network synchronizes every two epochs, balancing efficiency and alignment. Complete details of the algorithm can be found in Appendix E.

4 Experiment

We conducted extensive experiments on four real-world datasets to verify the effectiveness of the proposed method. nine benchmark models were introduced for comparison. Comparison methods can be found in Appendix F, and detailed descriptions of experimental criteria and evaluation metrics are provided in Appendix G.

4.1 Datasets

We evaluated SPsyINN on four widely used public datasets: Duolingo, MaiMemo, and EdNet.

- **Duolingo**[4]: A large-scale real-world dataset from a popular language learning app, containing 12.8 million logs from 115,222 users learning six languages. We used three subsets:
  - **Duolingo-En2De**: 249,744 records (English UI, learning German)
  - **Duolingo-En2Es**: 563,280 records (English UI, learning Spanish)
  - **Duolingo**: 2,171,328 records from the core dataset.
- **MaiMemo**[5]: Sourced from a leading English learning app in China, originally containing 200 million logs across 17,081 words. We selected 2,809,740 real user interactions for training and evaluation.

[4]https://www.duolingo.com/
[5]https://www.maimemo.com

Table 2: Component Ablation experiments. The model without any selected components is denoted as TNN+Classifier, excluding both denoising and symbolic regression modules. Selecting KA means the neural network and symbolic regression are jointly trained with knowledge alignment using the wait strategy. Selecting DW introduces dynamic weight optimization, where the loss weight of L_{DNN} adjusts dynamically based on performance; otherwise, weights remain fixed. Values represent neural network predictive performance averaged over five independent runs.

Component			En2Es			En2De			Duolingo			MaiMemo		
DN	KA	DW	PrAUC	G-means	Precision	PrAUC	G-means	Precision	PrAUC	G-means	Precision	PrAUC	G-means	Precision
			.8547	.3934	.7470	.8334	.4052	.7196	.8529	.3930	.7321	.8187	.3538	.6523
	✓		.8543	.4108	.7488	.8346	.3983	.7197	.8386	.4134	.7241	.8189	.3735	.6550
	✓	✓	.8552	.4108	**.7495**	.8272	.4325	.7193	.8407	.4133	.7255	.8187	.3788	.6555
✓			.8519	.3998	.7454	.8302	.4003	.7163	.8476	.3944	.7282	.8177	.3475	.6508
✓	✓		**.8584**	.3898	.7489	.8344	.3931	.7190	.8482	.3909	.7283	.8182	.3479	.6512
✓	✓	✓	.8497	**.4378**	.7494	**.8376**	**.4359**	**.7280**	**.8547**	**.4516**	**.7421**	**.8191**	**.3956**	**.6584**

- **EdNet**[6]: Collected over 2 years from 780K Korean users via the AI tutor platform Santa. We used 21,780,165 question-answering records from the EdNet-KT1 subset for our experiments.

4.2 Comparison Experiments

To demonstrate the effectiveness of the proposed SPsyINN, we compared its prediction accuracy with nine baseline methods on four datasets. The results are shown in Table 1. From Table 1, the following observations can be made: **1**.Compared to other baseline models, the proposed SPsyINN method significantly outperforms all benchmarks, demonstrating the effectiveness of our approach and offering a novel framework and perspective for memory behavior modeling. **2**.Neural network models exhibit clear advantages over memory theory equations. They can flexibly integrate various complex attributes, automatically learn relationships, and process these factors in parallel, enabling more accurate capture of intricate interactions and providing more precise predictions. **3**.Experimental results under different waiting strategies (SPsyINN-C, SPsyINN-I, SPsyINN-W) show that the per-round waiting strategy (SPsyINN-W) achieves the best performance. Strategies with higher synchronization rates perform better but at the cost of lower training efficiency. Users must balance performance and efficiency when selecting an appropriate strategy.

4.3 Ablation Study

To evaluate the impact of each module on the SPsyINN model, we conducted ablation experiments targeting one or two modules, including the Denoising module (DN, (3)), Knowledge Alignment (KA, (4)), and the Dynamic Weighting strategy (DW, (5)) in the DAO method.

Based on Table 2, the conclusions are as follows: **Symbolic regression improves performance**: Adding symbolic regression (second row) significantly enhances performance compared to the baseline DNN model (first row), demonstrating that symbolic knowledge can optimize neural network learning. **Synergy of knowledge alignment and dynamic optimization**: Models with both KA and DAO strategies (third and last rows) outperform those with only one (first and fourth rows), highlighting their combined effect in boosting accuracy and robustness. **Combining all components in the ablation study achieves the best results**: The SPsyINN-W model (last row), incorporating all components (DN, KA, DW), achieves the best performance across all datasets.

To further evaluate the adaptability of our model across datasets of varying scales, we compared its average epoch time (E-time), total interactions (n), effective interactions (effective), and performance metrics, as shown in Table 5. Each asynchronous strategy presents a different trade-off between performance and efficiency: SPsyINN-W emphasizes accuracy, SPsyINN-C prioritizes efficiency, while SPsyINN-I achieves a balance between the two. By incorporating a sampling-based update strategy into the genetic optimization process, we significantly alleviated the training delays caused by high synchronization costs on large-scale datasets, demonstrating the practicality and scalability of our approach.

4.4 Application Studies

To examine our method's theoretical implications, we analyzed the new equations discovered by the GSR module (Table 3). The results show that our method outperforms classical theoretical models in both accuracy and structure, highlighting its strength in identifying memory equations. The symbolic regression-derived formulas align with traditional memory theories while capturing more nuanced behavioral patterns. In both the MaiMemo and Duolingo datasets, SPsyINN frequently identifies structural terms like $(\delta_2 - \delta_3)$ and $(-\delta_2 + \delta_5)$, rooted in cognitive psychology. The term $(\delta_2 - \delta_3)$—the time since a word was last seen minus the time since the most recent memory event—reflects interference effects, consistent with interference-based forgetting theories [19, 37]. The term $(-\delta_2 + \delta_5)$ captures the relation between recent exposure and historical performance, in line with the DASH model [15]. On the MaiMemo dataset, SPsyINN often finds exponential forms similar to the HLR model [28], consistent with exponential decay theory [39], emphasizing the roles of time intervals and learner history. Overall, SPsyINN adheres to established psychological principles while uncovering deeper, interpretable patterns like chaining effects and individual learning dynamics, often overlooked by traditional models.

As shown in Table 4, we performed total-order sensitivity analysis on the equations generated by different strategies to evaluate the overall impact of input variables (or parameters) on the model output, including both the direct effects of variables and their interactions with other variables. The analysis results indicate that, in the

[6]https://github.com/riiid/ednet

Table 3: Memory Equation Comparison. The data of the SPsyINN model in the table comes from the output of the GSR module. All constants in the table are rounded to four decimal places. For precise values, please refer to our project.

Dataset	Model	RMSE	PrAUC	Complexity	Function
Duolingo	Wickelgren	.5052	.7175	9	$0.4793(1+0.3252\delta_2)^{0.2406}$
	ACT-R	.4774	.8393	8+$N*4$	$-0.6411+ln(\sum_{k-1}^{N}\delta_{2_k}^{0.9860})$
	DASH	.8108	.7695	8+$W*6$	$\sigma(0.5182-0.8620\delta_6+\sum_{w=1}^{\lvert W\rvert}0.9245\ln(1+\delta_5)+0.2186\ln(1+\delta_4))$
	HLR	.4532	.8409	16	$2^{\frac{\delta_2}{-0.0359\delta_1-0.0784\delta_2+0.0809\delta_3+0.2557\delta_4-0.1946\delta_5+0.2303\delta_6+0.3609}}$
	SPsyINN-C-F	.4421	.8659	9	$-13.7509\cdot\delta_3\cdot(\delta_2-\delta_3)+0.7153$
	SPsyINN-I-F	.4406	.8660	9	$-10.9915\cdot\delta_3\cdot(\delta_2-\delta_3)+0.7764$
	SPsyINN-W-F	**.4402**	**.8661**	8	$-10.5423^{\delta_2}\cdot(\delta_2-\delta_3)+0.7236$
MaiMemo	Wickelgren	.4201	.8174	9	$0.7597(1+0.5595\delta_2)^{-1.1098}$
	ACT-R	.4377	.8023	8+N*4	$-0.2451+ln(\sum_{k-1}^{N}\delta_{2_k}^{0.1715})$
	DASH	.4147	.8167	8+$W*6$	$\sigma(0.7059-0.4189\delta_6+\sum_{w=1}^{\lvert W\rvert}0.2522\ln(1+\delta_5)-0.1313\ln(1+\delta_4))$
	HLR	.4190	.8135	16	$2^{\frac{\delta_2}{-0.0146\delta_1+0.2934\delta_2+0.6771\delta_4+0.0420\delta_5+0.0769\delta_6+0.1214}}$
	SPsyINN-C-F	**.4011**	.8182	9	$0.5109^{((\delta_2\cdot\delta_4)^{(-\delta_2+\delta_5)})}$
	SPsyINN-I-F	.4033	.8186	9	$0.4890^{((\delta_2\cdot\delta_6)^{(-\delta_2+\delta_5)})}$
	SPsyINN-W-F	.4014	**.8204**	9	$0.5288^{((0.2781\cdot\delta_2)^{(-\delta_2+\delta_5)})}$

Table 4: Sensitivity of Equation Coefficients and Variable Sensitivity. "_" indicates that the equation does not include the variable or the sensitivity of the variable is less than 1×10^{-4}.

Dataset	Model	Function	Total-order indices							
			C_1	C_2	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6
Duolingo	SPsyINN-C	$C_1\cdot\delta_3\cdot(\delta_2-\delta_3)+C_2$	.0510	.8227	_	.0960	.1092	_	_	_
	SPsyINN-I	$C_1\cdot\delta_3\cdot(\delta_2-\delta_3)+C_2$	.0510	.8227	_	.0960	.1092	_	_	_
	SPsyINN-W	$C_1^{\delta_2}\cdot(-\delta_2+\delta_3)+C_2$	.0459	.4744	_	.2823	.2564	_	_	_
MaiMemo	SPsyINN-C	$C_1^{((\delta_2\cdot\delta_4)^{(-\delta_2+x_4)})}$	.6608	_	_	.1573	_	.1641	_	_
	SPsyINN-I	$C_1^{((\delta_2\cdot\delta_6)^{(-\delta_2+\delta_5)})}$	.6875	_	_	.2229	_	_	.1849	.0449
	SPsyINN-W	$C_1^{((C_2\cdot\delta_2)^{(-\delta_2+\delta_5)})}$	.6849	.0459	_	.2173	_	_	.1755	_

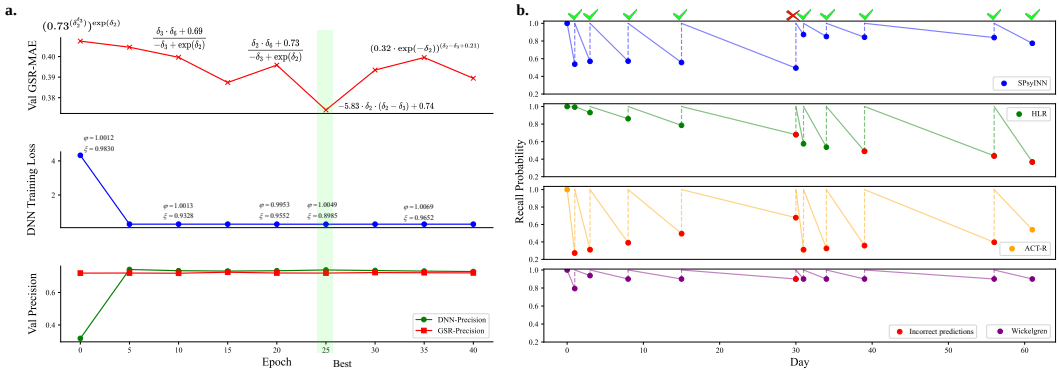

Figure 3: a. Dynamic training example of SPsyINN-W on the Duolingo dataset. b. Example of different memory equations predicting a learner's long-term memory effects (MaiMemo dataset).

Duolingo dataset, variables related to time intervals exhibit higher sensitivity, suggesting that memory states focus more on time interval information; whereas in the MaiMemo dataset, memory states pay attention to both time intervals and learners' historical performance. This difference reflects the distinct focuses of different learning platforms on memory prediction and demonstrates the model's adaptability to diverse data environments.

Table 5: Performance and Efficiency Evaluation of SPsyINN Asynchronous Optimization Strategies on Multiple Datasets.

Model	En2De(250K)			Duolingo(2.1M)			MaiMemo(2.8M)			EdNet(21M)		
	E-time	n/effective	PrAUC	E-time	n/effective	PrAUC	E-time	n/effective	PrAUC	E-time	n/effective	PrAUC
SPsyINN-C	4.5s	6/1	.8323	18s	19/2	.8451	68s	16/5	.8187	148s	13/7	.7744
SPsyINN-I	49s	6/4	.8324	55s	10/7	.8414	72s	10/6	.8182	168s	10/10	.7857
SPsyINN-W	89s	10/4	.8376	95s	18/6	.8547	116s	15/6	.8191	190s	12/7	.7721

Table 6: Performance of all baseline models and SPsyINN on the knowledge tracing dataset EdNet.

Model	EdNet		
	PrAUC	G-means	Precision
Wickelgren	.6908	.1537	.6722
ACT-R	.6865	.2612	.6762
DASH	.7028	.1705	.6749
HLR	.7301	.0065	.6730
DKT-Forget	.6971	.0126	.6758
FIFKT	.7515	.1426	.6786
SimpleKT	.7689	.2184	.6826
QIKT	.7355	.0427	.6760
MIKT	.7104	.0889	.6762
SPsyINN-C	.7744	.1995	.6819
SPsyINN-I	**.7857***	.2892	.6888
SPsyINN-W	.7721	**.3296***	**.6911***

Figure 3**a** illustrates the loss and corresponding evaluation metrics (Precision) changes during the joint training of DNN and GSR. From the perspective of knowledge distillation, learning from the Teacher is beneficial for training the Student model. In our model design, the neural network (DNN) and symbolic regression (GSR) serve as mutual Teachers, engaging in a dynamic, bidirectional distillation process. In the early stages of training, DNN acts as the Student and benefits from the structured "knowledge" provided by GSR, which serves as the Teacher by offering interpretable, theory-grounded guidance. As training progresses, enabled by our Dynamic Asynchronous Alignment (DAO) mechanism, this relationship reverses: DNN, now equipped with stronger predictive performance, takes on the Teacher role and provides feedback to GSR, guiding it toward discovering more accurate and psychologically meaningful equations. This dynamic, role-switching collaborative learning not only improves the model's predictive accuracy but also enhances the interpretability and theoretical alignment of the generated memory equations.

Additionally, considering the nature of behavioral data, relying solely on binary-labeled observations (e.g., correct = 1, incorrect = 0) often leads to "observation collapse," where the continuous nature of learners' recall states is lost. The symbolic equations discovered by GSR act as "soft labels," capturing the underlying probabilistic structure of memory performance. These soft, theory-driven signals enrich the learning process and help bridge the gap between data-driven prediction and cognitively informed interpretation, making the overall modeling process more robust and explainable.

Figure 3**b** compares the memory equations mined by our model with traditional memory equations during long-term memory fits for a specific user on a particular word. As shown, our method accurately finds equations that better fit learners' memory effects, while traditional memory theory equations show poor fitting. This indicates that the method not only adapts to individual memory patterns but also captures key factors in the dynamic changes of long-term memory.

To verify the robustness and scalability of the proposed model across datasets of varying sizes, we first examined its efficiency and performance under different asynchronous optimization strategies on multiple datasets, as shown in Table 5. Subsequently, we further evaluated its performance on the large-scale knowledge tracing dataset EdNet, as presented in Table 6. The experimental results demonstrate that SPsyINN achieves outstanding and stable performance across large-scale datasets including EdNet. Meanwhile, under a unified optimization strategy, the model maintains high computational efficiency, thereby fully reflecting its strong scalability and practical applicability.

5 Conclusion

We propose a novel psychologically interpretable dynamic asynchronous training model, SPsyINN, which effectively models memory behavior through knowledge injection and dynamic asynchronous optimization. Extensive experiments demonstrate that constraining neural networks with knowledge in memory scenarios is effective. Our framework enables efficient collaborative optimization of neural networks and symbolic regression, significantly improving the predictive performance of neural networks and the fitting accuracy of equations, thereby alleviating the issue of insufficient explanatory power of theoretical equations in memory scenarios. Methodologically, the dynamic alignment strategy enhances synergy, while in the asynchronous strategy, we observed a positive correlation between synchronization and model performance, though at the cost of training speed. In practical applications, SPsyINN reveals memory equations consistent with classical theories and identifies the dual influence of time intervals and learners' historical behaviors, offering valuable insights for memory modeling.

Future research will explore broader applications of SPsyINN, such as analyzing cognitive abilities like attention distribution and problem-solving, as well as applications in fields like cognitive science and finance. We aim to further enhance the model's generalizability, enabling it to integrate with other symbolic regression methods and offering a novel approach to scientific discovery.

Acknowledgments

This work was financially supported by the National Science and Technology Major Project (Grant No. 2022ZD0117103), the National Natural Science Foundation of China (Grant No. 62293554), the Youth AI Talents Fund of the Chinese Association of Automation under the Major Program (Grant No. HBRC-JKYZD-2024-310), the Higher Education Science Research Program of the China Association of Higher Education (Grant No. 23XXK0301), the Hubei Provincial Natural Science Foundation of China (Grant No. 2023AFA020) and the Fundamental Research Funds for the Central Universities (Grant No. CCNU25ai005 and CCNU25ai019).

References

[1] John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. An integrated theory of the mind. *Psychological review* 111, 4 (2004), 1036.

[2] Yanhong Bai, Jiabao Zhao, Tingjiang Wei, Qing Cai, and Liang He. 2024. A survey of explainable knowledge tracing. *Applied Intelligence* (2024), 1–32.

[3] Rhonda Douglas Brown and Rhonda Douglas Brown. 2018. Theories for Understanding the Neuroscience of Mathematical Cognitive Development. *Neuroscience of Mathematical Cognitive Development: From Infancy Through Emerging Adulthood* (2018), 1–19.

[4] Jiahao Chen, Zitao Liu, Shuyan Huang, Qiongqiong Liu, and Weiqi Luo. 2023. Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 14196–14204.

[5] Robert E Clark. 2018. A history and overview of the behavioral neuroscience of learning and memory. *Behavioral Neuroscience of Learning and Memory* (2018), 1–11.

[6] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. 2022. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing* 92, 3 (2022), 88.

[7] P Kingma Diederik. 2014. Adam: A method for stochastic optimization. *(No Title)* (2014).

[8] Hermann Ebbinghaus, HA Ruger, and CE Bussenius. 1913. Memory: A contribution to experimental psychology: Teachers college. *Columbia university* (1913).

[9] Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* (2023).

[10] Keejun Han, Mun Y Yi, Gahgene Gweon, and Jae-Gil Lee. 2013. Understanding the difficulty factors for learning materials: a qualitative study. In *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16*. Springer, 615–618.

[11] S Hochreiter. 1997. Long Short-term Memory. *Neural Computation MIT-Press* (1997).

[12] Johannes G Hoffer, Andreas B Ofner, Franz M Rohrhofer, Mario Lovrić, Roman Kern, Stefanie Lindstaedt, and Bernhard C Geiger. 2022. Theory-inspired machine learning—towards a synergy between knowledge and data. *Welding in the World* 66, 7 (2022), 1291–1304.

[13] Samuel Holt, Zhaozhi Qian, and Mihaela van der Schaar. 2023. Deep generative symbolic regression. *arXiv preprint arXiv:2401.00282* (2023).

[14] Xuhong Li, Haoyi Xiong, Xingjian Li, Xuanyu Wu, Xiao Zhang, Ji Liu, Jiang Bian, and Dejing Dou. 2022. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems* 64, 12 (2022), 3197–3234.

[15] Robert V Lindsey, Jeffery D Shroyer, Harold Pashler, and Michael C Mozer. 2014. Improving students' long-term knowledge retention through personalized review. *Psychological science* 25, 3 (2014), 639–647.

[16] Zachary Chase Lipton. 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv Preprint, CoRR, abs/1506.00019* (2015).

[17] Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, and Weiqi Luo. 2023. simpleKT: a simple but tough-to-beat baseline for knowledge tracing. *arXiv preprint arXiv:2302.06881* (2023).

[18] Boxuan Ma, Gayan Prasad Hettiarachchi, Sora Fukui, and Yuji Ando. 2023. Each encounter counts: Modeling language learning and forgetting. In *LAK23: 13th International Learning Analytics and Knowledge Conference*. 79–88.

[19] John A McGeoch. 1932. Forgetting and the law of disuse. *Psychological review* 39, 4 (1932), 352.

[20] Beat Meier, Alodie Rey-Mermet, Nicolas Rothen, and Peter Graf. 2013. Recognition memory across the lifespan: the impact of word frequency and study-test interval on estimates of familiarity and recollection. *Frontiers in Psychology* 4 (2013), 787.

[21] Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. 2023. Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics* 49, 4 (2023), 62.

[22] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. 2019. Augmenting knowledge tracing by considering forgetting behavior. In *The world wide web conference*. 3101–3107.

[23] Jeongbin Park, Bradford G Knight, Yingqian Liao, Marco Mangano, Bernardo Pacini, Kevin J Maki, Joaquim RRA Martins, Jing Sun, and Yulin Pan. 2023. CFD-based design optimization of ducted hydrokinetic turbines. *Scientific Reports* 13, 1 (2023), 17968.

[24] Harold Pashler, Nicholas Cepeda, Robert V Lindsey, Ed Vul, and Michael C Mozer. 2009. Predicting the optimal spacing of study: A multiscale context model of memory. *Advances in neural information processing systems* 22 (2009).

[25] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in neural information processing systems* 28 (2015).

[26] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378 (2019), 686–707.

[27] Cynthia Rudin. 2022. Why black box machine learning should be avoided for high-stakes decisions, in brief. *Nature Reviews Methods Primers* 2, 1 (2022), 81.

[28] Burr Settles and Brendan Meeder. 2016. A trainable spaced repetition model for language learning. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*. 1848–1858.

[29] Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan Reddy. 2023. Transformer-based planning for symbolic regression. *Advances in Neural Information Processing Systems* 36 (2023), 45907–45919.

[30] Shilong Shu, Liting Wang, and Junhua Tian. 2024. Improving Knowledge Tracing via Considering Students' Interaction Patterns. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 397–408.

[31] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).

[32] Jianwen Sun, Fenghua Yu, Qian Wan, Qing Li, Sannyuya Liu, and Xiaoxuan Shen. 2024. Interpretable Knowledge Tracing with Multiscale State Representation. In *Proceedings of the ACM on Web Conference 2024*. 3265–3276.

[33] Yuwei Tu, Weiyu Chen, and Christopher G Brinton. 2020. A deep learning approach to behavior-based learner modeling. *arXiv preprint arXiv:2001.08328* (2020).

[34] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[35] Haixin Wang, Yadi Cao, Zijie Huang, Yuxuan Liu, Peiyan Hu, Xiao Luo, Zezheng Song, Wanjia Zhao, Jilin Liu, Jinan Sun, et al. 2024. Recent Advances on Machine Learning for Computational Fluid Dynamics: A Survey. *arXiv preprint arXiv:2408.12171* (2024).

[36] Jie Wang, Jun Ai, Minyan Lu, Haoran Su, Dan Yu, Yutao Zhang, Junda Zhu, and Jingyu Liu. 2024. A Survey of Neural Network Robustness Assessment in Image Recognition. *arXiv preprint arXiv:2404.08285* (2024).

[37] Wayne A Wickelgren. 1974. Single-trace fragility theory of memory dynamics. *Memory & Cognition* 2, 4 (1974), 775–780.

[38] John T Wixted, Shana K Carpenter, et al. 2007. The Wickelgren power law and the Ebbinghaus savings function. *Psychological Science* 18, 2 (2007), 133.

[39] Piotr Woźniak, Edward Gorzelańczyk, and Janusz Murakowski. 1995. Two components of long-term memory. *Acta neurobiologiae experimentalis* 55, 4 (1995), 301–305.

A Details related to the equations of memory theory

For consistency, memory retention is represented as **Recall (*R*)** in the following equations:

- **Ebbinghaus** [8]: Memory retention ratio b as a function of the time gap t between learning sessions:

$$b = \frac{k}{(\log t)^c + k}$$

- **Wickelgren** [37]: Power-law decay model with initial memory strength λ, time factor β, forgetting rate ψ, and time t:

$$R = \lambda(1 + \beta t)^{-\psi}$$

- **Wozniak** [39]: Exponential decay model based on memory strength S and time t:

$$R = e^{-\frac{t}{S}}$$

- **ACT-R** [1]: Recall rate R depending on material difficulty β, review times t_k, and decay rates d_k:

$$R = \beta + \ln\left(\sum_{k=1}^{N} t_k^{-d_k}\right)$$

- **Wixted** [38]: Power-law function with time t, forgetting rate ψ, and parameter θ:

$$R = \theta t^{\psi}$$

- **MCM** [24]: Multi-exponential forgetting model with coefficients γ_i and time constants τ_i:

$$R = \sum_{i=1}^{N} \gamma_i \exp\left(-\frac{t}{\tau_i}\right) x_i(0)$$

- **DASH** [15]: Recall R modeled via student ability a_s, material difficulty d_c, attempts c_w, and correct counts n_w:

$$R = \sigma\left(a_s - d_c + \sum_{w=1}^{|W|} \theta_{2w-1} \ln(1 + c_w) + \theta_{2w} \ln(1 + n_w)\right)$$

- **HLR** [28]: Memory half-life h determined by features x, with recall decaying exponentially over time t:

$$R = 2^{-t/h}, \quad h = 2^{\theta x}$$

B Feature Description

We extracted six input features from the raw data:

- δ_1: The interval between the learner's first memory of the word and the current timestamp.
- δ_2: The interval between the learner's last memory of the word and the current timestamp.
- δ_3: The interval since the learner's last memory activity, regardless of the consistency of memory material.
- δ_4: The number of times the learner has reviewed the current word in prior memory activities.
- δ_5: The number of times the learner has reviewed the current word in previous memory activities and successfully recalled it during testing.
- δ_6: The length of the word, used as a simple descriptor of word difficulty.

During processing of the MaiMemo data, we did not obtain the δ_3 data and only retained δ_1, δ_2, δ_4, δ_5, and δ_6. In the Duolingo dataset, we use the complete features $\delta_{1:6}$. To account for the presence of certain review strategies in memory software, we standardized the above features using the training data set during model training.

C Principles of Diffusion Processes

In the forward diffusion process of Denosing Diffusion Probabilistic Models (DDPM) [31], the perturbation kernel is defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I\right)$$

where α_t is the noise scheduling parameter at step t. Thus, the Markov property of the forward diffusion in DDPM can be expressed as:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, I)$$

where the noise level at each step is controlled by $\sqrt{1-\alpha_t}$.

This result shows that the data at timestep t in the forward diffusion process of DDPM is a linear combination of x_0 and noise ε, with weights determined by the cumulative noise factor $\bar{\alpha}_t$ and $1-\bar{\alpha}_t$.

We extend DDPM by introducing a learnable noise weight, updating the perturbation kernel in DDPM as follows:

$$q(\tilde{x}_u^{t_{1:m}}|x_u^{t_{1:m}}) = \mathcal{N}\left(\tilde{x}_u^{t_{1:m}}; \sqrt{\alpha_t}x_u^{t_{1:m}}, \gamma^2(1-\alpha_t)I\right)$$

where γ is a learnable noise weight that adjusts the noise component. With this perturbation kernel, the diffusion process can be described as:

$$\tilde{x}_u^{t_{1:m}} = \sqrt{a_m} \cdot x_u^{t_{1:m}} + \gamma \cdot \varepsilon \cdot \sqrt{1-a_m}$$

where, $\alpha_m = \prod_{t=1}^{m}(1-\beta_t)$, consistent with the cumulative noise factor in DDPM. When $\gamma = 1$, the diffusion process is fully equivalent to DDPM.

In our model setup, the learnable noise weight γ provides the capability for dynamic noise level adjustment, enhancing adaptability and expressiveness across various tasks.

D Proof of Optimization by Sampling

Combining with the Monte Carlo approximation, we can express the loss L_{DNN} after adding the previously defined alignment loss in the following process.

$$\begin{aligned}
L_{DNN} &= \tfrac{1}{|\mathcal{D}|}\sum_u \sum_{i=1}^{m} [y_u^{t_i}\log(\hat{y}_u^{t_i}) + (1-y_u^{t_i})\log(1-\hat{y}_u^{t_i})] + \\
&\quad \varphi(\tilde{y}_u^{t_i} - \hat{y}_u^{t_i})^2 + \zeta(\hat{y}_u^{t_i} - \bar{y}_u^{t_i})^2 \\
&= \mathbb{E}_{(x_{u^*}^{t^*}, y_{u^*}^{t^*})\sim P(\mathcal{D})}([y_{u^*}^{t^*}\log(\hat{y}_{u^*}^{t^*}) + (1-y_{u^*}^{t^*})\log(1-\hat{y}_{u^*}^{t^*})] + \\
&\quad \varphi(\tilde{y}_{u^*}^{t^*} - \hat{y}_{u^*}^{t^*})^2 + \zeta(\hat{y}_{u^*}^{t^*} - \bar{y}_{u^*}^{t^*})^2) \\
&\approx \mathbb{E}_{(x_{u^*}^{t^*}, y_{u^*}^{t^*})\sim P(\mathcal{B})}([y_{u^*}^{t^*}\log(\hat{y}_{u^*}^{t^*}) + (1-y_{u^*}^{t^*})\log(1-\hat{y}_{u^*}^{t^*})] + \\
&\quad \varphi(\tilde{y}_{u^*}^{t^*} - \hat{y}_{u^*}^{t^*})^2) + \mathbb{E}_{(x_{u^*}^{t^*}, y_{u^*}^{t^*})\sim P(\mathcal{PD}^{SR})}\zeta(\hat{y}_{u^*}^{t^*} - \bar{y}_{u^*}^{t^*})^2
\end{aligned}$$

Similarly, for L_{GSR}, we can also obtain the following process.

$$\begin{aligned}
L_{GSR} &= \tfrac{1}{|\mathcal{D}|}\sum_u \sum_{i=1}^{m} (y_u^{t_i} - \bar{y}_u^{t_i})^2 + (\hat{y}_u^{t_i} - \bar{y}_u^{t_i})^2 \\
&= \mathbb{E}_{(x_{u^*}^{t^*}, y_{u^*}^{t^*})\sim P(\mathcal{D})}((y_{u^*}^{t^*} - \bar{y}_{u^*}^{t^*})^2 + (\hat{y}_{u^*}^{t^*} - \bar{y}_{u^*}^{t^*})^2) \\
&\approx \mathbb{E}_{(x_{u^*}^{t^*}, y_{u^*}^{t^*})\sim P(\mathcal{B})}((y_{u^*}^{t^*} - \bar{y}_{u^*}^{t^*})^2 \\
&\quad + \mathbb{E}_{(x_{u^*}^{t^*}, y_{u^*}^{t^*})\sim P(\mathcal{PD}^{NN})}(\hat{y}_{u^*}^{t^*} - \bar{y}_{u^*}^{t^*})^2
\end{aligned}$$

where, $(x_{u^*}^{t^*}, y_{u^*}^{t^*})$ refers to data sampled from dataset $\mathcal{D}$. It is sampled according to the probability $P(\mathcal{D})$, where $P(\mathcal{D})$ represents a uniform distribution; $\mathcal{B}$ is generated directly from $\mathcal{D}$, representing the batch size of data when calculating the loss. $\mathcal{PD}$ represents a proxy dataset also generated from $\mathcal{D}$, and it combines the outputs of the neural network and symbolic regression to construct the corresponding proxy data $\mathcal{PD}^{NN} = [x_u^{'t_{1:m}}, y_u^{'t_{1:m}}, \hat{y}_u^{'t_{1:m}}]$ and $\mathcal{PD}^{SR} = [x_u^{'t_{1:m}}, y_u^{'t_{1:m}}, \bar{y}_u^{'t_{1:m}}]$. During model optimization, the DNN uses $\mathcal{PD}^{SR}$, while the GSR uses $\mathcal{PD}^{NN}$.

Algorithm 3: SPsyINN-I

Input: The learner's word learning time series includes time, historical responses, word difficulty descriptions, target data y, number of epochs N, an initial set of traditional memory equations f
Output: Trained model parameters and optimized memory equations

Initialize neural network parameters, initialize genetic symbolic regression with f;
while *condition* **do**
  for $i = 1$ **to** N **do**
    Train neural network parameters and fit optimized memory equations;
    Save the current optimal parameters and equations;
    Sample $x_u^{'t_{1:m}}$ and save its predictions $\hat{y}_u^{'t_{1:m}}$ and $\bar{y}_u^{'t_{1:m}}$ as interaction data;
    GSR reads interaction data and updates the current optimal equation according to 3.5;
    if *i mod 2 == 0* **then**
      DNN reads interaction data from GSR and updates loss weights and parameters according to 3.5
return Parameters and optimized memory equations;

E Algorithm

Algorithm 1: SPsyINN-C

Input: The learner's word learning time series includes time, historical responses, word difficulty descriptions, target data y, number of epochs N, an initial set of traditional memory equations f
Output: Trained model parameters and optimized memory equations

Initialize neural network parameters, initialize genetic symbolic regression with f;
while *condition* **do**
  for $i = 1$ **to** N **do**
    Train neural network parameters and fit optimized memory equations;
    Save the current optimal parameters and equations;
    Sample $x_u^{'t_{1:m}}$ and save its predictions $\hat{y}_u^{'t_{1:m}}$ and $\bar{y}_u^{'t_{1:m}}$ as interaction data;
    Read interaction data from the local file, and update loss weights, parameters, and the current optimal equation according to 3.5;
return Parameters and optimized memory equations;

Algorithm 2: SPsyINN-W

Input: The learner's word learning time series includes time, historical responses, word difficulty descriptions, target data y, number of epochs N, an initial set of traditional memory equations f
Output: Trained model parameters and optimized memory equations

Initialize neural network parameters, initialize genetic symbolic regression with f;
while *condition* **do**
  for $i = 1$ **to** N **do**
    Train neural network parameters and fit optimized memory equations;
    Save the current optimal parameters and equations;
    Save the current optimal parameters and equations, and record the training time;
    Sample $x_u^{'t_{1:m}}$ and save its predictions $\hat{y}_u^{'t_{1:m}}$ and $\bar{y}_u^{'t_{1:m}}$ as interaction data;
    Mutually read interaction data and update loss weights, parameters, and the current optimal equation according to 3.5;
    Based on the training time, the models wait for each other;
return Parameters and optimized memory equations;

F Baselines

To evaluate the effectiveness and robustness of our SPsyINN model, we compare it with several state-of-the-art deep learning and traditional theoretical models. Details of the theoretical models are in Appendix A, and the deep learning models are briefly described below:

- **DKT-F** [22]: An extension of DKT [25] incorporating forgetting via time-related features such as repetition interval and past attempts.
- **FIFAKT** [18]: Uses attention to dynamically integrate forgetting, question format, and semantic similarity for better prediction.
- **SimpleKT** [17]: Models question-specific variations and time-related behaviors via dot-product attention to capture individual learning dynamics.
- **QIKT** [4]: Combines question-sensitive cognitive representations with Item Response Theory (IRT) to model and interpret students' knowledge states.
- **MIKT** [32]: Tracks both domain-level and concept-level knowledge using Rasch modeling and IRT for multi-level interpretability and improved performance.

G Experimental Setup

To train and validate the model, we used 80% of the student sequence data, reserving the remaining 20% for evaluation. All models were trained for 40 epochs using the Adam optimizer [7] and repeated five times. An early stopping strategy was adopted: optimization was halted if the loss on the validation set did not improve within the last five epochs.

All models were implemented in PyTorch and trained on a Linux server cluster equipped with NVIDIA GeForce GTX 2080Ti GPUs. Due to inconsistencies in evaluation metrics between the Duolingo and MaiMemo datasets, we primarily use PrAUC (Precision-Recall Area Under the Curve) and Precision as the main evaluation metrics, while G-Means and RMSE (Root Mean Squared Error) serve as secondary metrics. The calculation methods for these evaluation metrics are as follows.

$$\text{PrAUC} = \int_0^1 \text{Precision}(\text{Recall})\, d(\text{Recall})$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{G-Means} = \sqrt{\text{Precision}_0 \times \text{Precision}_1}$$

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{D}|} \sum_u \sum_{i=1}^{m} (y_u^{t_i} - \hat{y}_u^{t_i})^2}$$