Journal of Computer Science and Technology

Indexed in: SCIE, EI, Scopus, INSPEC, DBLP,CSCD, etc.
Sponsored by: ICT, CAS & CCF

JCST Papers

Only for Academic and Non-Commercial Use

Thanks for Reading!

JCST WeChat Subscription Account

JCST URL: https://jcst.ict.ac.cn

SPRINGER URL: https://www.springer.com/journal/11390

E-mail: jcst@ict.ac.cn

Online Submission: https://mc03.manuscriptcentral.com/jcst

Twitter: JCST_Journal

LinkedIn: Journal of Computer Science and Technology

Li F, Zhang TC, Yin YF *et al.* A survey of static and temporal explainable methods and their applications in knowledge tracing. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY, 40(4): 1022-1045, July 2025. DOI: 10.1007/s11390-025-4239-0, CSTR: 32374.14.s11390-025-4239-0

A Survey of Static and Temporal Explainable Methods and Their Applications in Knowledge Tracing

Fan Li[1] (李　凡), Tian-Cheng Zhang[1,*] (张天成), *Senior Member, CCF*, Yi-Fang Yin[2] (尹一方)
Di Fan[1] (樊　迪), *Senior Member, CCF*, Ming-He Yu[3] (于明鹤), *Senior Member, CCF*
and Ge Yu[1] (于　戈), *Fellow, CCF, Senior Member, ACM, IEEE*

[1] *School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China*

[2] *Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 138634, Singapore*

[3] *Software College, Northeastern University, Shenyang 110819, China*

E-mail: 2110691@stu.neu.edu.cn; tczhang@mail.neu.edu.cn; yin_yifang@i2r.a-star.edu.sg; 2290185@stu.neu.edu.cn
yuminghe@mail.neu.edu.cn; yuge@mail.neu.edu.cn

Received March 1, 2024; accepted April 2, 2025.

Abstract Deep learning has found widespread application across diverse domains owing to its exceptional performance. Nevertheless, the lack of transparency in deep learning models' decision-making processes undermines their usability, especially in critical contexts. While researchers have made noteworthy advancements in explaining these models, they have frequently overlooked the differences between static and temporal models during explanation generation. In temporal models, features change over time, posing new challenges in the generation of explanations. Though extensive research has been dedicated to surmounting these hurdles, a survey summarizing these contributions is currently absent. To bridge this gap, this paper endeavors to summarize existing methods and their contributions in terms of both static and temporal models, highlighting their disparities. Additionally, we propose an innovative classification approach based on the comprehensibility of explanations, demonstrating that different explanation methods vary in their understandability for users. Finally, to assess the limitations of the explanation capabilities of existing methods, we specifically choose knowledge tracing to analyze the evolution of explanation methods in this context of temporal modeling and interpretations.

Keywords deep learning, explainable artificial intelligence, temporal model, knowledge tracing

1 Introduction

Artificial intelligence, particularly deep learning, has demonstrated remarkable performance across various domains. These impressive achievements often come at the cost of increased model complexity and the integration of nonlinearity. This limitation significantly hampers human understanding of the decision-making processes of these models, commonly referred to as the black-box nature of deep learning models. In critical domains, understanding a model's decision-making process and mitigating potential errors are essential. For example, in autonomous driving, model errors could result in fatal accidents.

To solve the above issue, researchers have proposed explanation methods to assist humans in understanding the decisions made by deep learning models. These methods have a wide range of applications, such as identifying features that significantly impact the model's output[1] and providing explanations and recommendations for individuals subjected to unfavorable treatment by deep learning decisions, a process known as recourse[2]. Therefore, a growing amount of work has been proposed for generating explanations and mainstream machine learning libraries have incorporated their own explainable artificial intelligence libraries, such as PyTorch Captum[3]. In some studies[4], "explanation" and "interpretation" are con-

Survey

This work was supported by the National Natural Science Foundation of China under Grant Nos. 62272093, 62137001, and 62372097.

*Corresponding Author

sidered equivalent, but in others[5], "explanation" refers to explanations provided by other methods for the model, while "interpretation" refers to explanations provided by the model itself. In this paper, we will adopt the terminology used by the original authors when it does not lead to ambiguity. In other cases, we will provide sufficient information for readers to differentiate between distinct terminologies.

Recent surveys have summarized these studies from various perspectives, including the explanation stage[6], the scope of the explanation[7], and the integration of external knowledge[4], among others.

However, these surveys have overlooked the difference between explanation methods for static models and those for temporal models when summarizing their findings. Temporal deep learning models have exhibited exceptional performance across a variety of domains, including critical applications such as electronic health records[8] and clinical early warning systems[9], but merely providing predictions may not establish adequate trust with users. Some work[10–13] has attempted to address these challenges. However, there is no survey summarizing these efforts.

This survey aims to address the existing gaps and provide an updated overview of the progress of explanation generation for both static and temporal models, with the main contributions summarized below.

• This paper systematically summarizes the development of explanation methods, encompassing the entire lifecycle of explanation methods, including design, evaluation, and application. This perspective aids in understanding key principles and best practices for designing effective explanations, promoting real-world adoption.

• We provide an overview of the evolution of explainability in temporal models, extending beyond static models. Furthermore, we analyze and discuss the distinctions between these two domains, emphasizing the necessary advancements in interpretability techniques to better accommodate sequential data.

• We introduce a novel hierarchical taxonomy to classify and summarize prior arts based on the comprehensibility of their explanations. This taxonomy offers a structured and intuitive way to categorize existing methods, aiding researchers in identifying suitable approaches.

• We evaluate the effectiveness of universal interpretable methods on knowledge tracing models and discuss the challenges and constraints of their application in this domain. By highlighting their limitations, we aim to pave the way for future improvements tailored to knowledge tracing.

The structure of this survey unfolds as follows. We introduce our hierarchical taxonomy for explanation methods in Section 2. We present the evolution of static explanation methods in Section 3, and explore the development and advancements of temporal explanation methods in knowledge tracing in Section 4. We elaborate on the evolution of evaluation methods designed for explanation methods in Section 5. We apply general explanation methods to knowledge tracing, and evaluate their performance and identify their limitations in Section 6. We explore potential research directions in Section 7. We conclude in Section 8.

2 Proposed Hierarchical Taxonomy

The details of our hierarchical taxonomy are illustrated in Fig.1. First, we categorize existing methods into static and temporal based on their application domains, and then classify and summarize them by their explanation comprehensibility in both static and temporal domain.

Static methods refer to explanation methods designed for static models that process static data, such as images and tables. In contrast, temporal methods are explanation methods tailored for temporal models

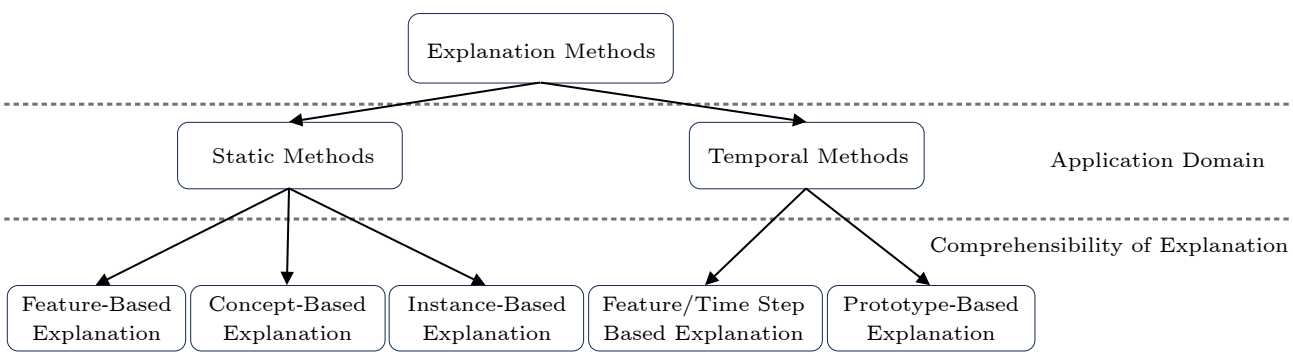


Fig.1. Overview of our taxonomy.

that handle sequential data, including time series and text. Notably, if an explanation method applies to any model but does not account for the complexities of temporal data, it is classified as a static method.

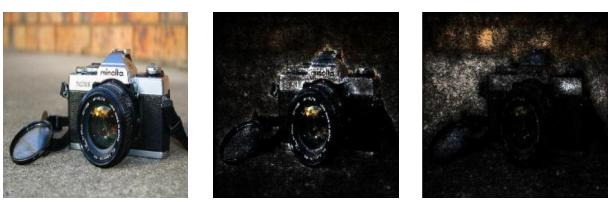
Static methods can be further categorized into three types: feature-based, concept-based, and instance-based explanations, with comprehensibility progressively increasing. Similarly, temporal methods are divided into feature/time step based and prototype-based explanations.

3 Static Methods

Static methods generate explanations based on the assumption that data is time-invariant or lacks temporal dependencies among features.

3.1 Feature-Based Explanation

Feature-based explanation provides insights by highlighting the importance of each raw feature in a model's prediction. We present an example of image classification models[14] in Fig.2, where the explanation highlights the parts of the input image that are significant for the model's output. Based on how they assign importance scores to each feature, we classify feature-based explanation methods into five categories: gradient-based, attribution propagation based, perturbation-based, surrogate model based, and attention-based methods.



(a) (b) (c)

Fig.2. Example of saliency map generated by Integrated Gradients and Gradients. (a) Original image. (b) Integrated Gradients. (c) Gradients.

3.1.1 *Gradient-Based Methods*

Gradient-based methods utilize the gradient of the model output with respect to each input feature to measure the importance score of each feature[15]. However, solely relying on gradients may underestimate the importance of features whose contribution to the output has saturated, which is referred to as the insensitivity problem. To solve this problem, Sundararajan *et al.*[14] proposed Integrated Gradients, which integrates the gradient along a path from reference input $\hat{x}$ to current input x. This integration path yields feature attributions that precisely sum to the difference between the model's output at input x and at the reference input $\hat{x}$, thereby addressing the insensitivity problem. As shown in Fig.2, the Integrated Gradients method generates a saliency map with clearer salient features. When the trajectory is the straightline path between $\hat{x}$ and x, the important score $S(x_i)$ of the i-th feature of x to model f is defined as follows:

$$S(x_i) = (x_i - \hat{x}_i) \int_{\alpha=0}^{1} \frac{\partial f(\hat{x} + \alpha(x - \hat{x}))}{\partial x_i} \, \mathrm{d}\alpha.$$

However, gradient-based methods often generate noisy explanations. Although interpretable methods highlight the important regions of the image, the noise affects the interpretability's clarity. To address this issue, SmoothGrad[16] initially generates multiple perturbed versions of the image of interest by adding noise and subsequently averages the importance scores over these perturbed instances. Kapishnikov *et al.*[17] found that high-magnitude gradient regions in the Integrated Gradients method can cause noisy explanations. To address this, they proposed the Guided Integrated Gradients method, which adapts the gradient integration path based on the input, baseline, and model. The saliency-guided training method[18] iteratively masks input features with noisy gradients while minimizing the KL (Kullbark-Leibler) divergence between the outputs of the original and masked inputs. Additionally, a few studies aim to identify and address theoretical deficiencies in the Integrated Gradients method. Lundström *et al.*[19] observed that the assumptions of function spaces in the Integrated Gradients method are unavailable in the deep learning context. To address this limitation, they introduced an additional axiom, non-decreasing positivity, and rigorously extended it to a diverse deep learning function space.

3.1.2 *Attribution Propagation Based Methods*

Similar to gradient-based methods, attribution propagation techniques use backpropagation to generate explanations. The key difference is that they attribute contributions between adjacent layers rather than relying on gradients.

Layerwise Relevance Propagation (LRP)[20] begins at the output layer and iteratively redistributes the output scores, moving layer by layer, until reaching the input layer. ϵ-LRP is a commonly used imple-

mentation method. It calculates the relevance $r_i^{(l)}$ of unit i in layer l using the following process:

$$r_i^{(l)} = \sum_j \frac{z_{ji}}{\sum_{i'}(z_{ji'}+b_j) + \epsilon \times \text{sign}\left(\sum_{i'}(z_{ji'}+b_j)\right)} r_j^{(l+1)},$$

where z_{ji} represents the weight of the activation from neuron i to neuron j in the subsequent layer. The relevance $r_j^{(l+1)}$ denotes the relevance of unit j in layer l+1, while b_j represents the bias of unit j. Additionally, ϵ is a small value used to mitigate numerical instabilities. However, LRP may underestimate features' importance when they have saturated their contribution to the output. Deep Learning Important FeaTures (DeepLIFT)[21] addresses this by incorporating reference activation. It calculates importance scores by comparing the activation of each neuron with its reference activation. This ensures significant signal propagation even in saturated feature regions.

3.1.3 Perturbation-Based Methods

Perturbation-based methods allocate contributions to each feature by examining the effects of perturbing input features on model output. Perturbation operations involve removing, permutating, and altering features. We summarize and analyze existing research on perturbation from two perspectives: feature perturbation techniques and feature importance calculation methods.

Based on the reliance on the data distribution and model to be explained, feature perturbation methods can be classified into the following categories.

Independent of Data Distribution. These methods perturb features without accounting for data distribution. Zeiler and Fergus[22] set the features of interest to zero, whereas Suresh *et al.*[23] replaced them with random noise sampled from a uniform distribution.

Depending on Data Distribution. These methods perturb features based on data distribution. Dabkowski and Gal[24] replaced the features of interest with the mean value of that feature over all instances. On the other hand, Fong and Vedaldi[25] and Fong *et al.*[26] applied a Gaussian blur kernel to generate blurred features as a replacement for the features of interest. Lundberg and Lee[1] and Covert *et al.*[27] marginalized unrelated features based on conditional distribution. Agarwal and Nguyen[28] used a generative model to perturb features and produce instances that are realistic under the true data distribution.

Depending on Data Distribution and Model. To perturb features in alignment with both the data distribution and the model being explained, adversarial perturbations[29] are employed to gain valuable insights into the decision boundaries of black-box models. However, without theoretical guarantees, such replacements may inaccurately represent feature absence, leading to unreliable explanations. To address this, Ren *et al.*[30] proposed a causal pattern-based method to identify optimal replacements and introduced a metric to assess their fidelity.

After perturbing the inputs, the next crucial step is to determine the contribution of each feature. One popular class of methods used for contribution assignment is output difference analysis. These methods assign contributions to features based on the degradation in performance when removing each feature individually. The degradation in performance is defined as follows:

$$\phi_i = \mathbb{E}\left[\ell\left(f_i\left(X_{-i}\right), Y\right)\right] - \mathbb{E}[\ell(f(X), Y)],$$

where ϕ_i denotes the importance of the i-th feature, and ℓ is the performance evaluation function. The first term represents the expected loss after removing the i-th feature, while the second term corresponds to the expected loss using the full set of features.

Zintgraf *et al.*[31] measured the contribution of the i-th feature by using the weight of evidence, which is the difference between the log odds of the original and the perturbed class probabilities $f_c(x)$ and $f_c(x_{-i})$. However, output difference analysis overlooks feature interactions. To address this, Shapley value based methods[1, 27] evaluate all possible feature subsets. These methods frame feature contribution assignment as a cooperative game, where features are players and predictive power represents the profit. The attribution of the i-th feature is computed as:

$$S(i) = \sum_{Z \subseteq F \setminus i} \frac{|Z|!(|F| - |Z| - 1)!}{|F|!} \left(f(Z \cup i) - f(Z)\right),$$

where F is the set of all features, and Z is a subset of features. To analyze higher-order feature interactions, Fel *et al.*[32] employed Sobol indices[33] to decompose the model's prediction variance.

Several studies have attempted to identify a mask that distinguishes unrelated features from those useful for prediction. For constant perturbation, mask-based perturbation is defined as follows:

$$\Phi(X, M) = X \odot M + A \odot (1 - M),$$

where M is the mask that indicates the regions to be

removed or preserved, X represents the original sample, and A is an alternative sample. Subsequently, they optimize objective function $-\log(f_c(\Phi(X, M)))$ to identify regions that can recognize the selected class.

Generally, regions considered informative should meet three requirements[24, 25]. First, they should be free of artifacts. Second, they should be the smallest sufficient regions of the image that, when isolated, allow confident classification. Finally, they should be the smallest destroying regions of the image that, when removed, prevent confident classification.

To obtain a meaningful region free of artifacts, Dabkowski and Gal[24] introduced an additional term to minimize the difference between adjacent values in M, ensuring a smoother region. To identify a small region for removal or preservation, Dabkowski and Gal[24] incorporated the average mask value into the objective function. Similarly, Fong and Vedaldi[25] enforced sparsity by applying the Manhattan norm $\|M\|_1$ to the selected region.

3.1.4 Surrogate Model Based Methods

Surrogate model based methods approximate a complex model using a simple, interpretable surrogate model, from which explanations are derived.

One widely used approach is the Local Interpretable Model-Agnostic Explanations (LIME) method[34]. As shown in Fig.3, LIME starts by sampling instances around the target instance of interest. The corresponding labels of sampled instances are generated by the model f to be explained. Next, an interpretable surrogate model g is optimized by minimizing the following formula:

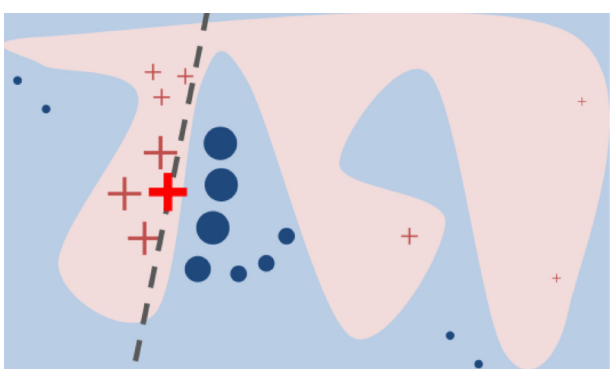


Fig.3. LIME example[34]. The pink/blue background signifies the presence of the black-box model, while the prominent red cross serves to identify the instance undergoing explanation. The remaining red crosses and blue dots represent data points acquired through sampling, with their sizes reflecting their similarity to the specific instance being explained.

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in \mathcal{Z}} \pi_x(z)(f(z) - g(z))^2,$$

where $\mathcal{Z}$ is the set of sampled instances, z is a sampled instance arounding the instance of interest x, and π_x is a function that weights the distance between x and z.

Visani *et al.*[35] found that LIME's stability is not always guaranteed. To address this, Zhou *et al.*[36] proposed Stabilized-LIME, which utilizes the central limit theorem to determine the necessary number of perturbation samples for stable explanations.

3.1.5 Attention-Based Methods

The aforementioned methods are post-hoc approaches designed to explain the original model, raising concerns about explanation fidelity. In contrast, attention-based methods are inherently interpretable[8] and offer higher explanation fidelity than post-hoc methods. Their interpretability stems directly from attention weights, eliminating the need for additional explanatory mechanisms. However, the interpretability of attention weights has been a subject of controversy[37, 38]. Fig.4(a) illustrates the observed model attention and Fig.4(b) illustrates an adversarially generated set of attention weights. Despite the notable differences between these two distributions, they both lead to the same prediction value of 0.01. Jain and Wallace[37] also observed that the attention weights do not exhibit a correlation with feature importance metrics, such as gradient-based measures[14, 15].

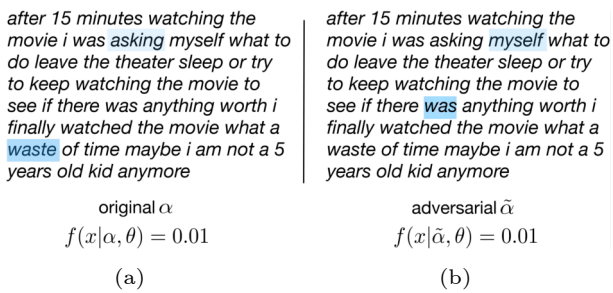


Fig.4. Heatmap illustrating the attention weights generated from a negative movie review[37]. (a) Attention weights. (b) Adversarially constructed set of attention weights.

Serrano and Smith[39] found that attention weights have weak interpretability when the model's output depends on feature interactions. One way to enhance their interpretability is through intelligible downstream operations, such as additive models[40]. Alternatively, Liu *et al.*[41] observed that attention-based explanations lack robustness in determining feature

impact polarity. To address this, they proposed a faithfulness violation test to assess the consistency between explanation weights and impact polarity.

Additionally, intermediate attention scores are often overlooked in complex attention-based models, which may lead to unreliable explanations. Therefore, Chefer *et al.*[42] introduced an attribution-based approach for generating explanations. To enhance the interpretability of attention weights, Ron and Hazan[43] imposed constraints on the position of attention layers, such as placing them near the input layer.

3.2 Concept-Based Explanation

Feature-based explanations rely on raw features, such as pixels, while concept-based methods provide explanations grounded in human-understandable concepts. These concept-based explanations are more intuitive and accessible to human reasoning.

Kim *et al.*[44] proposed Concept Activation Vectors (CAVs) as a means to interpret the internal state of a neural network using human-friendly concepts. A CAV refers to the normal vector of a hyperplane that separates examples containing a concept from those that are not in the activation space of the model. To obtain the CAV of concept c (striped texture), two datasets are constructed: one with examples exhibiting striped textures and the other with random images. These datasets are then used to train a linear classifier to differentiate between their activations. The weight vector $\boldsymbol{v}_c$ of the classifier represents the CAV for concept c. Given $\boldsymbol{v}_c$ as the CAV vector for concept c and $h_k(\boldsymbol{x})$ as the logit for a data point $\boldsymbol{x}$ for class k, the sensitivity of the model prediction for class k concerning concept c can be computed as the directional derivative $S_{c,\,k}(\boldsymbol{x})$:

$$\begin{aligned} S_{c,\,k}(\boldsymbol{x}) &= \lim_{\epsilon \to 0} \frac{h_k\left(f(\boldsymbol{x}) + \epsilon \boldsymbol{v}_c\right) - h_k\left(f(\boldsymbol{x})\right)}{\epsilon} \\ &= \nabla h_k\left(f(\boldsymbol{x})\right) \boldsymbol{v}_c. \end{aligned}$$

Acquiring human-annotated examples for each predefined concept is typically costly. To address this, Self-Explaining Neural Network (SENN)[40] was introduced, which automatically learns concepts by leveraging maximally activated prototypes, eliminating the need for artificially constructed datasets. Automated Concept-based Explanation (ACE)[45] applies image segmentation and clustering to automatically identify concepts. However, ACE suffers from inconsistencies in the learned concept weights. To mitigate this, Zhang *et al.*[46] proposed using matrix factorization for feature maps in concept learning. For faithful explanations, Framework to Learn With INTerpretation (FLINT)[47] employs a joint learning approach, where a predictor network and its interpreter network collaborate to generate concept-based explanations.

After obtaining concepts, another important task is to analyze how these concepts impact the model's predictions. Barbiero *et al.*[48] utilized First-Order Logic to provide insights into how the network uses these concepts to make predictions. Huang *et al.*[49] presented a visual analytics system that allows for the interactive exploration and investigation of concepts, enabling a better understanding of their influence. Tran *et al.*[50] uncovered binary concepts that have a significant causal effect on the model's output.

3.3 Instance-Based Explanation

Instance-based explanation selects representative examples or generates counterfactual instances to explain model predictions. This approach aims to create easily understandable explanations that mimic human reasoning, with instances serving as valuable tools for comprehending complex knowledge[51].

3.3.1 *Counterfactual Instance Based Methods*

Counterfactual instance based methods apply minimal changes to the original instances to generate counterfactual instances with different outputs. These changes provide valuable insights into understanding the decision boundaries within the model's internal mechanisms. We categorize existing methods based on two key components of counterfactual explanation: instance generation and selection. First, we introduce counterfactual instance generation methods, categorized as follows.

Optimization-Based Methods. These methods carefully design loss functions to incorporate desired characteristics into counterfactual instances. Typically, these loss functions can be efficiently optimized using gradient descent algorithms to obtain the desired counterfactual instances. As the pioneering work in introducing counterfactual explanations, Wachter *et al.*[52] uncovered desirable counterfactual instances by minimizing the following objective. This objective seeks to find a counterfactual instance x' that closely resembles the original instance x, while achieving a new and desirable output y' when evaluated by the model f:

$$\arg\min_{x'}\max_{\lambda}\lambda\left(f\left(x'\right)-y'\right)^2+d\left(x,x'\right),$$

where the parameters of model f are fixed, $d(\cdot)$ represents the distance function between instances, and the hyperparameter λ adjusts the weight of the two terms. To identify plausible counterfactual instances, Dhurandhar *et al.*[53] proposed evaluating the L_2 reconstruction error of x' using an autoencoder to ensure its plausibility. For the same objective, Kanamori *et al.*[54] introduced the Mahalanobis distance and local outlier factor. Instead of modifying raw features, DISSECT[55] first identifies and disentangles the concepts used by the model in decision-making, then amplifies their influence to generate counterfactual instances. Similarly, Koh *et al.*[56] built upon concept bottleneck models[57] and manipulated human-defined concepts to alter the model's output and generate counterfactual explanations.

Approximation Algorithm Based Methods. These methods employ heuristic strategies to identify counterfactual instances. While they are often more efficient than optimization-based methods, there exists a trade-off between time consumption and the quality of results[58]. Goyal *et al.*[59] utilized greedy search based approaches to address the minimum-edit counterfactual problem. Russell[60] formulated the problem of obtaining plausible and diverse counterfactual instances as a linear program, which is solved using mixed integer programming. Additionally, Schleich *et al.*[61] introduced a generic algorithmic approach for real-time retrieval of diverse counterfactual instances.

Next, we introduce the existing methods for selecting counterfactual instances. Generally, multiple counterfactual explanations are available for each instance, and selecting good counterfactual instances is another crucial step[62, 63]. The common selection criteria are as follows.

One of the criteria is validity, which states that the output of the counterfactual instance should be close to the predefined output. Mothilal *et al.*[62] defined validity as the proportion of counterfactual examples in which the output matches the predefined output.

Another criterion is feasibility, which states that the counterfactual instance should either closely resemble the original instance or the suggested change should be readily adopted and actionable[63]. For example, if a person is rejected for a loan, a counterfactual explanation advising to change the person's age would not be possible. This property becomes even more crucial when generating counterfactual explanations for recourse in critical domains[64], where recourse involves providing actionable recommendations to individuals to achieve desired outcomes.

3.3.2 *Prototype-Based Methods*

Prototype-based methods generate explanations by using the similarity between prototypes and current instances, where prototypes are representative data instances from the dataset. We present an example of a prototype-based explanation[65] in Fig.5.

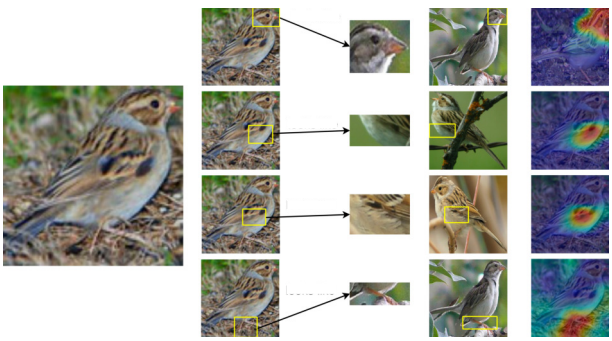


Fig.5. Prototype-based explanation. It features a test image of a clay-colored sparrow in the first column. The third column highlights the similar regions between the test image in the first column and the prototype depicted in the fourth column, while the rightmost column presents the corresponding activation map, illustrating the basis for the model's prediction[65].

Li *et al.*[66] proposed early work on prototype-based interpretable neural networks. They first trained an autoencoder to learn a latent low-dimensional space, offering a better similarity measure than the pixel space. Subsequently, they identified prototypes in that space and made predictions based on their similarity to the encoded inputs. Assuming the existence of m prototypes $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m\}$ and the encoded inputs $\boldsymbol{z}$, the L^2 distance between the inputs and prototypes can be expressed as follows:

$$d(\boldsymbol{z})=\left[\|\boldsymbol{z}-\boldsymbol{p}_1\|_2^2, \|\boldsymbol{z}-\boldsymbol{p}_2\|_2^2, \ldots, \|\boldsymbol{z}-\boldsymbol{p}_m\|_2^2\right].$$

The prediction is then determined using the softmax function as follows:

$$p(\boldsymbol{z})=\operatorname{softmax}\left(\boldsymbol{W}d(\boldsymbol{z})\right),$$

where $p(\boldsymbol{z})$ represents the model's prediction and $\boldsymbol{W}$ is a learnable weight matrix.

Instead of using entire images[66], Prototypical Part Network (ProtoPNet)[65] uses parts of images as prototypes. By utilizing smaller spatial dimensions, ProtoPNet enables more fine-grained comparisons, allowing different parts of an image to be compared with different prototypes.

However, assigning each prototype exclusively to a single class results in a large number of prototypes. To address this, Prototypical Part Shared Network (ProtoShare)[67] merges prototypes with similar semantics and allows them to be shared across classes, thereby reducing the number of prototypes. Neural Prototype Tree (ProtoTree)[68] simplifies model explanations by organizing prototypes hierarchically. To align the perception of similarity between humans and the similarity learned by the classification model in existing methods[65, 68], Nauta *et al.*[69] enhanced visual prototypes by incorporating additional quantitative information, such as the influence of color hue in images. Relying solely on prototypes to explain the behavior of models may lead to over-generalization and misunderstandings. To address this limitation, Kim *et al.*[70] proposed a unified framework called MMD-critic to introduce criticism-based explanations in addition to prototype-based explanations.

4 Temporal Methods: From General Frameworks to Knowledge Tracing

Temporal models like Long Short-Term Memory (LSTM)[71] excel across various domains. However, in critical applications, such as analyzing electronic health records[8] and clinical early warning systems[9, 12], the lack of interpretability may undermine user trust. Early interpretability methods for temporal models often directly adopt approaches originally designed for static models. However, these methods overlook critical aspects, such as temporal dependencies and the abundance of input features unique to temporal models, leading to unreliable and inaccurate explanations[10, 72]. Recently, explanation methods that are specifically designed for temporal models have been proposed.

In this section, we extend the taxonomy used for explanation methods in static models to those in temporal models. They are divided into two classes: feature/time step based methods and prototype-based methods. Additionally, we review the progress of explainable methods in knowledge tracing, a specific application domain for temporal models.

4.1 Feature/Time Step Based Explanation

Feature/time step based explanation methods provide importance scores for features or time steps as explanations. In this scenario, the same feature holds varying importance at different time steps. Therefore, unlike static feature based methods, these approaches account for the influence of the temporal dimension. These methods score features or time steps for explanation. Some are specifically designed for popular models like LSTM and attention-based models, while others are more versatile. Consequently, we categorize them as either model-specific or model-agnostic methods based on their correlation with target models.

4.1.1 *Model-Specific Methods*

Model-specific methods generate explanations tailored to the unique architectures of individual models. Murdoch and Szlam[73] decomposed the output of LSTM into a product of factors, with each term representing the contribution of a specific word. These results are used to construct an interpretable rule based classifier while preserving much of LSTM's accuracy. Not limited to word-based importance scores, contextual decomposition[74] computes both word importance scores and their interactions by decomposing the LSTM output. IMV-LSTM[75] generates both feature importance and time step importance by learning feature-wise hidden states. In contrast, attention-based approaches inherently provide explanations by leveraging attention weights, without requiring additional methods[76]. However, the interpretability of attention mechanisms remains a subject of debate within the machine learning community[37–39].

4.1.2 *Model-Agnostic Methods*

Different from model-specific methods that generate explanations for specific model architectures, model-agnostic methods are capable of providing explanations for any type of black-box model. This versatility makes them applicable in a broader range of scenarios.

Ismail *et al.*[10] argued that directly applying static explanation methods to temporal models results in explanations of poor quality due to the ignoring of time and feature domains. To address this issue, they proposed the Temporal Saliency Rescaling (TSR) approach to compute the importance scores of time steps and features, separately. TSR determines the importance score of the i-th feature at time t, denoted as $R^{\mathrm{TSR}}_{i,\,t}$, through the product of the time-relevance score Δt and the feature-relevance score Δi, defined as follows:

$$R^{\mathrm{TSR}}_{i,\,t} = \big|\underbrace{(R(x) - R(x_{-t}))}_{\Delta t}\underbrace{(R(x) - R(x_{-i}))}_{\Delta i}\big|,$$

where $R(\cdot)$ represents a static interpretation method, such as Integrated Gradients[14], or DeepSHAP[1], used to calculate the saliency values, and x_{-t} represents the input x without time step t. x_{-i} represents the input x without the i-th feature. The results are valid only when Δt exceeds a certain threshold.

Tonekaboni *et al.*[11] proposed the Feature Importance in Time (FIT) method to assess feature importance over time by quantifying each feature's contribution to temporal shifts in the model's output distribution. In contrast to FIT, which computes the importance of a feature set at a single time step, the Windowed Feature Importance in Time (WinIT)[13] method evaluates feature importance over multiple time steps, capturing temporal patterns in feature importance.

Perturbation-based methods have also been recently introduced into temporal settings. For building perturbation with time dependency of data, Crabbé and Schaar[72] proposed dynamic perturbation that constructs a perturbation for each feature at each time step by utilizing the value of the same feature at adjacent time steps. Inspired by the excellent performance and solid theoretical fundamental of KernelSHAP[1], TimeSHAP[77] incorporates sequence-wide perturbations, which are specifically designed for temporal models. To provide diverse explanations, TIME[78] applies various perturbation methods to allow the identification of important features, temporal windows, and the influence of value ordering within those windows. When using perturbation-based techniques in temporal models, generating out-of-distribution inputs may lead to explanations that are socially misaligned[79, 80]. To address this, Parvatharaju *et al.*[81] proposed Prioritized Replacement Selector, which employs weighted dataset sampling to identify the most appropriate temporal instance for perturbing the target instance.

4.2 Prototype-Based Explanation

Prototype-based methods have been applied in temporal models by emulating human problem-solving processes to generate explanations that are easily understandable to humans. Drawing inspiration from case-based reasoning[82], Prototype Sequence Network (ProSeNet)[83] extracts a compact set of prototypical sequences from the original dataset, subsequently inferring new inputs by comparing them with the prototypes. Building on the foundations of ProSeNet, Prototype Steering (ProtoSteer)[84] enables end-users to integrate their knowledge into ProSeNet without developer intervention. This is achieved through the addition, deletion, or revision of prototypes.

To offer concise explanations, $\mathrm{SCN}_{\mathrm{pro}}$[12] employs segments (sub-sequences) of sequences as prototypes, diverging from using the entire sequence. For similar purposes, Multi-Level Attention-Based Prototype Network (MapNet)[85] incorporates an attention mechanism to learn short-term and long-term prototypes, separately, with the aim of acquiring more representative prototypes for both features and time intervals.

In this subsection, we have discussed the key findings related to temporal explanation methods, emphasizing the main trends and insights from the literature. As shown in Table 1, we compare these methods based on their explanation types. These results provide a solid foundation for further exploration in future studies.

4.3 Explanation for Knowledge Tracing

This subsection focuses on the development of explanation methods in knowledge tracing. Although knowledge tracing is a type of temporal model, the explanation methods in current knowledge tracing research remain underdeveloped and exhibit notable limitations. We categorize and summarize explanation methods for knowledge tracing into two distinct types: feature-based explanations, and educational concept based explanations where features primarily refer to interactions between students and exercises.

4.3.1 Feature-Based Explanation

These methods mainly analyze interaction influences on current predictions to generate explanations.

Attention-based methods represent a valuable approach for offering intrinsic explanations within knowledge tracing models. Self-Attentive Knowledge Tracing (SAKT)[86] is the first knowledge tracing model based on attention mechanisms. Subsequently, the Attentive Knowledge Tracing (AKT) model[87] incorporates time intervals and problem similarity to compute attention weights. Additionally, Pandey and Srivastava[88] leveraged the similarity among exercise texts to enhance the rationale behind the attention weights. Here is an example of an attention weight used to explain the model's prediction. As shown in

Table 1. Summary of Temporal Explanation Methods

Method	Explanation Type				Intrinsic or Post-Hoc	Model Type
	Feature	Temporal	Subsequence	Prototype		
IMV-LSTM[75]	✔	✔			Intrinsic	LSTM
[8, 76]	✔				Intrinsic	Attention
[73, 74]	✔				Post-hoc	LSTM
TSR[10]	✔	✔			Post-hoc	Temporal model
FIT[11]		✔			Post-hoc	Temporal model
WinIT[13]		✔	✔		Post-hoc	Temporal model
TimeSHAP[77]	✔	✔			Post-hoc	Temporal model
Dynamask[72]	✔	✔			Post-hoc	Temporal model
TIME[78]	✔	✔	✔		Post-hoc	Temporal model
ProSeNet[83]				✔	Intrinsic	Temporal model
ProtoSteer[84]				✔	Intrinsic	Temporal model
SCN_{pro}[12]			✔	✔	Intrinsic	Temporal model
MapNet[85]			✔	✔	Intrinsic	Temporal model

Fig.6, we present the three attention heads in AKT[87], which are used to predict student performance and operate on distinct time scales, each with a unique attention window width.

Apart from attention-based explanations, [89, 90] apply GAM-based methods to provide intrinsic explanations for knowledge tracing. Here, the Generalized Additive Model (GAM)[91] is considered interpretable because its predictions depend linearly on functions of the inputs. Yang and Cheung[92] employed a decision tree to preprocess features and analyze feature importance. Wang *et al.*[93] proposed a knowledge tracing method based on tensor factorization[94], which imposes constraints on the tensor to yield interpretable parameters, thereby facilitating the discovery of relationships between problems and exercises.

Rather than incorporating interpretable structures into knowledge tracing models, post-hoc methods such as LIME[34], LRP[20], and KernelSHAP[1] have been employed to explain these models[95–97]. Furthermore, Ding and Larson[98] applied post-hoc methods to uncover the underlying mechanisms of knowledge tracing. Their study suggests that the superior performance of RNN-based models primarily results from mapping original inputs to a high-dimensional space, rather than effectively modeling temporal information.

4.3.2 *Educational Concept Based Explanation*

Feature-based explanations offer a valuable means of understanding the inner workings of black-box models by generating explanations based on raw features. Recent efforts have focused on providing explanations grounded in educational concepts, such as student ability and exercise difficulty.

To infer student abilities, DKT-DSC[99] divides the interaction sequence into multiple time intervals, with the student's ability being iteratively updated after each interval. Ghosh *et al.*[87] encoded exercises diffi-

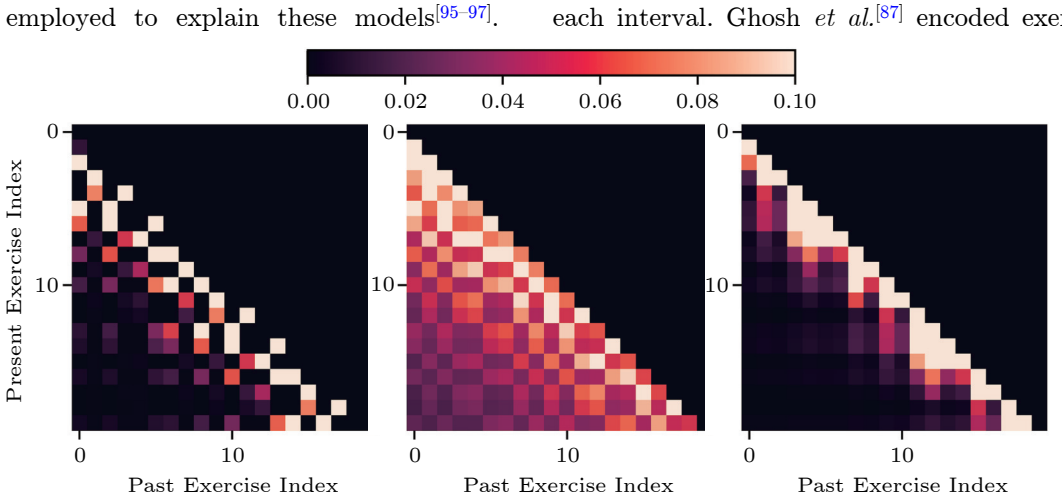


Fig.6. Attention weights of the three attention heads in AKT[87].

culty into exercise embedding based on Rash model based embeddings[100]. The exercise embedding is defined by the following equation:

$$\boldsymbol{x}_t = \boldsymbol{c}_{c_t} + \mu_{q_t} \times \boldsymbol{d}_{c_t},$$

where $\boldsymbol{c}_{c_t}$ denotes the embedding of the concept associated with an exercise, $\boldsymbol{d}_{c_t}$ means the variations present in exercises containing the same concept, and μ_{q_t} is a difficulty parameter representing the extent to which an exercise deviates from the associated concept.

In addition to meaningful educational parameters, Deep-IRT[101] and QIKT[102] design a transparent and interpretable component based on Item Response Theory (IRT)[100], which takes as input both students' mastery of knowledge concepts and exercise difficulties over time, obtained by processing students' learning interactions. Furthermore, Su *et al.*[103] incorporated Multidimensional Item Response Theory (MIRT) into knowledge tracing models for situations where multiple concepts are required to answer an exercise correctly. Minn *et al.*[104] extracted skill mastery, student ability, and exercise difficulty from raw data, and then used the Tree-Augmented Naive Bayes classifier[105] to forecast student performance.

Mining the relationships between educational items is another way to provide explanations. Educational items refer to concepts, exercises, etc. DKT[106] extracts relationships between knowledge concepts from trained knowledge tracing models. This is conducted to assess whether DKT can accurately discern the correct knowledge structure. Beyond considering the knowledge structure merely as an evaluation metric to judge the rationality of the underlying mechanisms, Graph-based Knowledge Tracing (GKT)[107] introduces knowledge structures as a form of relational inductive bias. This incorporation aims to enhance both the performance and inherent interpretability of knowledge tracing. A method based on the multi-head attention mechanism to introduce the relationship between knowledge concepts is defined as follows:

$$f_{\text{neighbor}}\left(\boldsymbol{h}_i^t, \boldsymbol{h}_j^t\right) = \frac{1}{M} \sum_{k \in M} \alpha_{ij}^k f_k\left(\boldsymbol{h}_i^t, \boldsymbol{h}_j^t\right),$$

where k represents the head index among a total of M heads. α_{ij}^k is the attention weight that represents relationship from concept i to j, and f_k represents the neural network of the k-th head. Similarly, Song *et al.*[108] employed a graph convolutional network[109, 110] to learn relationships between concepts and exercises. Furthermore, Tong *et al.*[111] introduced the Hierarchical Graph Knowledge Tracing (HGKT) model, which constructs a hierarchical graph to represent intricate knowledge structures among knowledge concepts, exercises, and problem schemas[112], thereby enhancing interpretability.

To conclude, this subsection provides a detailed examination of explanation methods for knowledge tracing from various perspectives, including feature-based approaches and educational concept based approaches. Table 2 summarizes the key findings and comparisons, offering a concise overview of their underlying educational theories and evaluation metrics.

5 Evaluation of Explanation Methods

Evaluating the quality of explanations is a subsequent critical concern that arises after generating explanations for static or temporal models[113]. The desirable and widely recognized properties of explanations include robustness[114], sensitivity[115–117], brevity[72], and legibility[118], among others. To assess explanation methods based on these properties, various evaluation techniques have been proposed. We first summarize the commonly used datasets, and then, based on the involvement of human feedback, categorize these methods into two groups: subjective and objective evaluation.

5.1 Datasets

When evaluating the performance of explanation methods, certain datasets are frequently employed across various explanation methods. We summarize the commonly used datasets as follows.

The MIMIC-III dataset[119] is a multivariate clinical time series dataset containing various vital signs and laboratory measurements collected over time from approximately 40 000 patients at the Beth Israel Deaconess Medical Center in Boston. Temporal explanation methods often leverage this dataset to evaluate their performance, including FIT[11], TIME[78], WinIT[13], and others.

The Caltech-UCSD Birds-200-2011 dataset① is an image dataset containing annotations for 200 bird species. The images were sourced from Flickr and curated by workers on Amazon Mechanical Turk. Each

①https://www.vision.caltech.edu/datasets/cub_200_2011, Feb. 2025.

Table 2. Summary of Explanation Methods in Knowledge Tracing

Method	Intrinsic or Post-Hoc	Feature-Based Explanation	Concept-Based Explanation	Education Theory	Explanation Evaluation
DKT[106]	Post-hoc		✔		a), b)
AKT[87]	Intrinsic	✔	✔	Rash model	a), b), c)
RKT[88]	Intrinsic	✔			c)
SAKT[86]	Intrinsic	✔			b), c)
GKT[107]	Intrinsic	✔			a)
HGKT[111]	Intrinsic	✔			a)
JKT[108]	Intrinsic		✔		d), e)
QIKT[102]	Intrinsic		✔	IRT	a)
Deep-IRT[101]	Intrinsic		✔	IRT	f), g)
TC-MIRT[103]	Intrinsic		✔	MIRT	d)
DPFA[90]	Intrinsic	✔		PFA	a), b)
KTM[89]	Intrinsic	✔			d)
LRP-based[97]	Post-hoc	✔			c), d)
SHAPley-based[95]	Post-hoc	✔			h)
LIME-based[96]	Post-hoc	✔			h)

Note: a) Visualization of student knowledge state change over time. b) Mining of exercise relationships. c) Visualization of historical interaction influence. d) Mining of concept relationships. e) Mining of relationships between concepts and exercises. f) Mining of exercise difficulty. g) Mining of student ability. h) Analysis of important features.

image is annotated with a bounding box, rough bird segmentation, and a set of attribute labels. This dataset is commonly used to evaluate static explanation methods, including ProtoPNet[65], FLINT[47], and ProtoShare[67], and others.

MNIST② is a dataset containing a large collection of handwritten digits. It is widely used in the field of machine learning for training various image processing systems. Although the images in the MNIST dataset do not include a temporal dimension, this dataset was also used to evaluate temporal methods in addition to static methods. These studies include KernelSHAP[1], TSR[10], FLINT[47], and MMD-critic[70], and others.

Since the important features of real-world datasets are often unknown, researchers typically construct synthetic datasets based on application scenarios. These simulated datasets, where important features that impact the output of an instance are known, help improve the objectivity of the evaluation process. Studies that adopt synthetic datasets to evaluate the performance of their explanation methods include LRP[20], TIME[78], WinIT[13], DISSECT[55], Dynamask[72], and others.

5.2 Subjective Evaluation

Subjective evaluation depends on human feedback or judgments and plays a pivotal role in assessing explanation methods. This is because the primary goal of explanations is to help humans understand the decision-making processes of black-box models, aiding in debugging or preventing errors caused by these models. We classify existing subjective evaluation methods into two categories: evaluation based on explanation examples and evaluation based on crowdsourcing.

5.2.1 Explanation Examples Based Evaluation

These methods utilize easily comprehensible formats, such as visualizations, to present explanations and evaluate the quality of those explanations based on certain prior information. For instance, in the context of a classification task, the efficacy of a saliency map[14–16] is evaluated by ascertaining its ability to effectively emphasize the relevant portions of the original image pertaining to the prediction, as illustrated in Fig.2. For prototype-based explanations, Li *et al.*[66] employed visualization of learned prototypes to ascertain their alignment with real-world instances. For concept-based explanations, visualizing learned concepts is a crucial evaluation technique. Kim *et al.*[44] organized images based on their relevance to the concept being examined. Then, the most and least similar images are shown to confirm how well the learned

②https://yann.lecun.org/exdb/mnist/index.html, Feb. 2025.

concept aligns with human intuition.

5.2.2 Crowd Sourcing Based Evaluation

These methods evaluate the quality of explanations by collecting feedback from individuals through proxy tasks. Ribeiro *et al.*[34] designed proxy tasks that require participants to utilize explanations to complete them. For instance, one task involves selecting the classifier with better generalization based on the provided explanations. Parekh *et al.*[47] conducted a survey in which participants evaluated how well the visualizations of explanations aligned with the corresponding manually generated textual descriptions. Yeh *et al.*[120] designed a proxy task in which participants were shown explanations of a given concept and asked to select the image that best represents it.

5.3 Objective Evaluation

While subjective evaluation provides direct human feedback on explanations, it is prone to biases inherent in human judgment[115, 117]. Therefore, objective evaluation is essential to complement subjective assessment. We categorize existing methods into four categories: ground truth based, removal-based, proxy task based, and rule-based approaches.

5.3.1 Ground Truth Based Methods

An obstacle in objectively evaluating explanations is the lack of access to the ground truth decision process of black-box models. This limitation is particularly pronounced in feature-based explanations, where determining the true importance of features is often infeasible[13]. Therefore, ground truth based methods evaluate explanations based on the utilization of white-box models or datasets with known feature importance. The explanation can be quantified by comparing the important features from the explanation and the ground truth important features[78].

5.3.2 Removal-Based Methods

In contrast to white-box models or datasets where feature importance is known, black-box models lack prior information about critical dataset features or the features on which the models depend. Removal-based methods remove critical features highlighted by explanation techniques and measure the resulting performance change to assess whether the explanations accurately highlight critical features[26, 72, 78]. However, solely removing the crucial value introduces ambiguity as to whether the drop in performance is caused by the absence of essential information or by the introduction of anomalies outside the original training distribution. To eliminate the latter cause, RemOve and Retrain (ROAR)[121] retrains new models on a modified dataset where the significant features are removed, ensuring that all inputs conform to the distribution encountered by the models. To achieve a more comprehensive evaluation, the progressive perturbation approach[10] gradually perturbs features from the most to the least important, constructing a performance degradation curve. A smaller area under this curve indicates higher explanation quality.

5.3.3 Proxy Task Based Methods

These methods carefully design proxy tasks to indirectly evaluate explanation techniques by measuring their performance on these specific tasks. Shrikumar *et al.*[21] designed a task to evaluate whether removing specific important features identified by explanation methods results in the reclassification of the instance into a different class. Zhang *et al.*[122] introduced Point Game as a metric to quantify the localization accuracy of saliency maps. The process begins by identifying the highest-scoring point on the saliency map. A "hit" is recorded when this peak point matches the annotated labels, while a "miss" occurs when no match is found. The degree of localization accuracy is then quantified by the ratio of the number of hits to the total count.

5.3.4 Axiom-Based Methods

These methods assess explanations by examining whether interpretable methods adhere to desired properties. One form of verification involves examining the theoretical foundations of these explanation methods. Sundararajan and Najmi[123] discovered that utilizing SHAP[1] in a multiplicative manner for model explanation leads to the absence of uniqueness in the results. In response to this issue, they introduced an axiomatic framework aimed at comparing different methods based on SHAP. This comprehensive framework incorporates axioms such as the dummy axiom, efficiency axiom, and proportionality axiom, among others. Sundararajan *et al.*[14] argued that ex-

planation methods must adhere to the Implementation Invariance axiom, which states that explanations should remain consistent for two functionally equivalent networks, meaning that these networks produce identical outputs for the same inputs, despite having distinct implementations.

In conclusion, we highlight the key differences and similarities among evaluation methods for explanations, with Table 3 providing a comprehensive summary that emphasizes the metrics of various evaluation methods.

6 Performance of Explainable Methods on Knowledge Tracing

In this section, we first summarize the development of deep learning based knowledge tracing models. Then, we evaluate explanation methods on these knowledge tracing models.

6.1 Deep Learning Based Knowledge Tracing

Initial knowledge tracing models predominantly rely on probabilistic graphical models or logistic regression-based techniques, such as Bayesian knowledge tracing[124]. While these approaches offer interpretability, they yield suboptimal performance. With the progression of deep learning, Deep Learning Based Knowledge Tracing (DLKT) has obtained remarkable performance[125–127]. The development of DLKT is closely related to the advancement of deep learning technologies. Therefore, we classify and summarize them based on the deep learning techniques employed by knowledge tracing models.

Recurrent Neural Network Based Methods. Deep Knowledge Tracing (DKT)[106] is the pioneering application of deep learning models to the knowledge tracing task. It utilizes recurrent neural networks[71] and their variants as the primary components.

Memory Neural Network Based Methods. Despite DKT's notable improvements in predictive accuracy, it lacks explicit modeling of concept mastery levels, as it uses a single hidden state to encode all knowledge states. Inspired by the memory neural network[128], Dynamic Key-Value Memory Network (DKVMN)[129] introduces two external memory modules: the key matrix for storing specific concepts and the value matrix for storing corresponding mastery levels.

Graph Neural Network Based Methods. Given the relationships between concepts, coursework concepts can be structured as a graph. Incorporating this graph structure presents a promising approach to improving both the performance and interpretability of knowledge tracing. GKT[107] utilizes Graph Neural Networks (GNN)[130, 131] to incorporate the relationships between concepts. When an interaction with an exercise involving a concept is observed, GKT updates not only the mastery of that specific concept, but also that of its neighboring concepts by leveraging their relationships.

Attention-Based Methods. The Self-Attention Knowledge Tracing (SAKT) model[86] is an earlier knowledge tracing model that adopts the attention mechanism. SAKT employs this mechanism to identify relevant interactions from a student's historical data, thereby predicting knowledge mastery. However,

Table 3. Summary of Evaluation Methods for Explanations

	Method	Reference	Metric			
			Accuracy	Fidelity	Robustness	Sensitivity
Subjective	Proxy task	[34, 45]	✔	✔		
	Proxy task	[44, 70, 120]	✔			
	Survey	[120]	✔			
	Visualization	[8, 65, 66]	✔			
Objective	Quantitative metrics	[63]			✔	
	Proxy task	[21, 116, 122]	✔			
	Proxy task	[62]	✔	✔		
	Randomization test	[115–117]				✔
	Game theory	[1]	✔		✔	
	Theoretical proof	[14]				✔
	Quantitative metrics	[72]	✔			
	Quantitative metrics	[40]	✔		✔	
	Quantitative metrics	[10, 45]	✔	✔		
	Synthesized dataset	[44]	✔			
	Synthesized dataset	[11, 78]	✔	✔		

Ghosh *et al.*[87] found that SAKT underperforms compared with DKT and DKVMN in their experiments. In response, they introduced the Attentive Knowledge Tracing (AKT) model, which utilizes adapted attention networks to generate context-aware representations that encompass a learner's entire practice history. As a result, AKT demonstrates superior performance in predicting learner outcomes.

6.2 Experiments

Subsections 4.3 and 6.1 summarize the development of knowledge tracing and interpretability in this field. However, most of these interpretability approaches explain only certain parts of a model's structure and fail to clarify the rationale behind its decisions. Furthermore, as they are designed for specific models, their evaluations lack objectivity. Therefore, in this subsection, we focus on model-agnostic explanation methods from the broader field of explainability and analyze their effectiveness in the context of knowledge tracing. To ensure a rigorous evaluation, we selectively identify representative knowledge tracing models and explanation methods, which are then thoroughly assessed using public datasets.

Datasets. We assess the performance of knowledge tracing models and interpretable methods using two benchmark datasets: ASSISTment 2009[③] and ASSISTment 2017[④]. These datasets are gathered from an online tutoring platform. In the ASSISTment 2009 dataset, records without corresponding knowledge concepts are removed, resulting in 4 151 students, 16 891 exercises, and 110 associated concepts. This processed dataset contains 325 637 records. The ASSISTment 2017 dataset includes 1 709 students, 3 162 exercises, and 102 associated concepts, with 942 816 interaction records for these students.

Knowledge Tracing Models. To comprehensively examine the landscape of knowledge tracing architectures, we select four diverse models: DKT[106], DKVMN[129], GKT[107], and AKT[87].

Explanation Methods. For explanation methods, we select FO[22], LIME[34], and KernelSHAP[1] as representatives of static explanation methods, while FIT[11], TimeSHAP[77], and WinIT[13] represent temporal explanation methods. Additionally, we include two other comparison methods: the Random method, which assigns importance to each interaction randomly, and the LastK method, which assigns importance based on temporal proximity, giving higher importance to more recent interactions. When generating explanations for knowledge tracing, we treat each interaction as a feature, which includes exercise, concept, and response.

Evaluation. Due to the lack of ground-truth explanations for real data, we conduct a performance deterioration test to evaluate the quality of explanations generated by different methods. Specifically, we systematically remove the top K most important interactions from each student's sequence. Across all experiments, we report the AUC (Area Under the Curve) drop and Loss Raise when important features are removed. The AUC drop measures the reduction in AUC after removing critical features, while Loss Raise quantifies the increase in loss following their removal.

Result. We first train knowledge tracing models and select optimal parameters, ensuring their performance aligns with results reported in the literature. The AUC and accuracy (ACC) metrics are presented in Table 4. Subsequently, we evaluate the performance of various explanation methods across all knowledge tracing models, as shown in Tables 5 and 6. On the ASSISTment 2009 dataset, KernelSHAP emerges as the optimal explanation method for each model. However, on the ASSISTment 2017 dataset, various models exhibit distinct optimal explanation methods. This suggests that current explanation methods face challenges in consistently achieving high performance across diverse models and datasets, indicating limited generalization ability. Furthermore, although the WinIT and FIT methods are designed for temporal models, their performance remains suboptimal across different datasets and models. One possible explanation is that in knowledge tracing, temporal dependencies encompass not only the time interval between interactions but also the relationship within the knowledge structure across interactions[87]. However, the temporal method TimeSHAP achieves

Table 4. Performance of Knowledge Tracing Model

KT Model	ASSISTment 2009		ASSISTment 2017	
	AUC	ACC	AUC	ACC
DKT	0.803 1	0.753 6	0.712 8	0.690 5
DKVMN	0.794 1	0.748 8	0.711 6	0.690 2
GKT	0.809 9	0.753 8	0.734 1	0.702 2
AKT	0.826 6	0.768 1	0.760 0	0.713 1

[③]https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/skill-builder-data-2009-2010, Jun. 2025.
[④]https://sites.google.com/view/assistmentsdatamining/dataset, Jun. 2025.

Table 5. Performance Comparison of Explanation Methods for the DKT and DKVMN Models

Method	DKT				DKVMN			
	ASSISTment 2009		ASSISTment 2017		ASSISTment 2009		ASSISTment 2017	
	AUC Drop	Loss Raise	AUC Drop	Loss Raise	AUC Drop	Loss Raise	AUC Drop	Loss Raise
Random	0.058 39	1.131 05	0.023 69	–0.021 73	0.045 85	0.511 24	0.018 49	–0.011 93
LastK	0.011 23	0.306 42	0.001 04	–0.083 31	0.004 40	–0.150 25	–0.001 93	–0.141 38
FO	0.061 51	1.241 16	**0.116 48**	0.777 45	0.089 13	1.840 79	**0.117 01**	**0.892 26**
LIME	0.156 55	2.805 19	0.103 53	**0.915 14**	0.152 42	2.534 59	0.094 36	0.591 96
KernelSHAP	0.169 21	**2.880 89**	0.104 54	0.893 83	**0.158 07**	**2.547 96**	0.094 21	0.574 22
WinIT	0.067 61	1.240 72	0.066 64	0.273 84	0.073 04	1.098 08	0.066 67	0.327 44
FIT	0.133 29	2.259 78	0.069 68	0.624 82	0.133 03	2.105 74	0.070 24	0.576 04
TimeSHAP	**0.170 42**	2.785 45	0.085 85	0.877 38	0.146 63	2.514 50	0.072 55	0.622 36

Table 6. Performance Comparison of Explanation Methods for the GKT and AKT Models

Method	GKT				AKT			
	ASSISTment 2009		ASSISTment 2017		ASSISTment 2009		ASSISTment 2017	
	AUC Drop	Loss Raise	AUC Drop	Loss Raise	AUC Drop	Loss Raise	AUC Drop	Loss Raise
Random	0.013 98	0.198 19	0.001 69	0.082 15	0.051 70	0.880 62	0.018 30	0.299 36
LastK	0.009 25	0.153 23	0.000 07	–0.018 77	0.009 99	–0.026 46	–0.001 14	–0.109 00
Attention	—	—	—	—	0.104 87	1.791 33	0.063 42	0.653 49
FO	0.058 67	1.070 70	0.007 38	–0.007 17	0.080 17	1.574 78	0.082 53	0.927 99
LIME	**0.074 96**	1.121 58	0.032 72	0.082 12	0.121 62	1.923 62	0.085 98	1.060 70
KernelSHAP	0.067 56	**1.161 50**	**0.033 83**	**0.132 65**	**0.133 75**	**2.211 88**	0.088 83	1.102 56
WinIT	0.060 08	0.931 21	0.007 70	–0.050 52	0.070 08	0.917 57	0.051 52	0.756 12
FIT	0.030 95	0.549 67	0.005 72	0.101 61	0.117 01	1.854 93	0.067 91	0.893 02
TimeSHAP	0.063 25	1.111 17	0.009 01	–0.042 07	0.121 70	1.964 66	**0.093 57**	**1.106 40**

optimal performance in explaining the AKT model on the ASSISTment 2017 dataset. We attribute this to treating the learning record as temporal data with a single feature, interaction, making TimeSHAP and KernelSHAP roughly equivalent for interpreting knowledge tracing models. The key difference lies in TimeSHAP's pruning strategy, which accounts for the observed performance variation.

Impact of Hyperparameter K on Explanation Methods. The hyperparameter K refers to the number of important features removed. We analyze its impact on the performance of various explanation methods applied to AKT. In addition to the explainability methods mentioned earlier, we include the performance without feature removal as the Baseline method to illustrate the relative performance of other methods. As shown in Fig.7, we observe that, in most cases, other explanation methods generally outperform the Random method, demonstrating their effectiveness on ASSISTment 2017. However, the LastK method, which can be considered as a feature removal approach based on temporal proximity, performs even worse than the Random method. This finding suggests that, for knowledge tracing models, temporal proximity is not a reliable measure of feature importance.

Case Study. We use KernelSHAP and the Attention method to analyze AKT's predictions within the ASSISTment 2009 dataset. As shown in Fig.8, to simplify the analysis, we present the concepts and their corresponding responses, omitting the exercise component. For example, "87:1" indicates that the student answer correctly on an exercise involving concept 87, whereas "87:0" indicates an incorrect response on an exercise involving the same concept. Additionally, the final value of 0.98 indicates the probability that the AKT model predicts the student will correctly answer the exercise in the last interaction. The explanation provided by KernelSHAP aligns with the educational theory, indicating that the model relies on similar interactions to predict the current one. Conversely, attention-based explanations overwhelmingly emphasize the most recent interaction, neglecting earlier ones. This observation suggests that attention weights may not accurately reflect the model's decision-making process[132].

7 Future Research Directions

Despite significant progress in explainable methods for knowledge tracing, which holds promise for educational research, several critical challenges re-

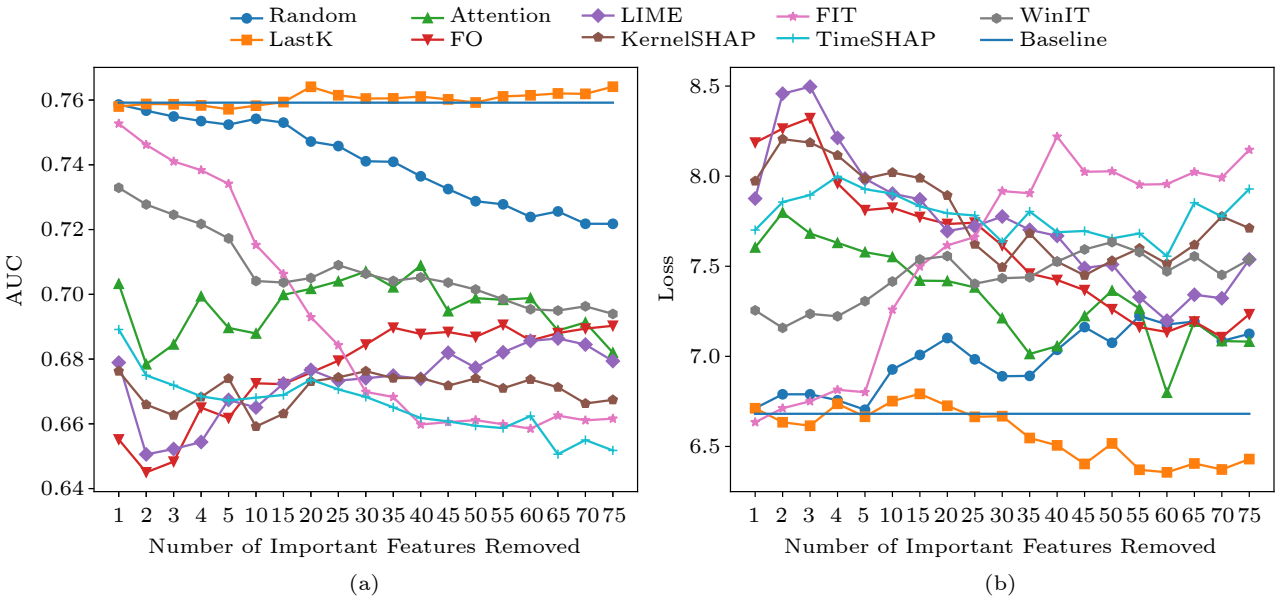


Fig.7. Performance of explanation methods varies with the increase in K on ASSISTment 2017, where the Baseline refers to the model performance with no features removed. (a) AUC. (b) Loss.

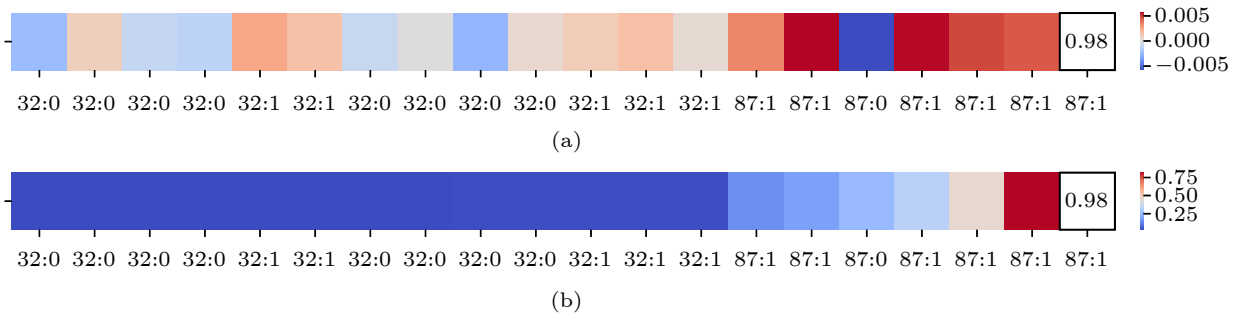


Fig.8. Visualization of explanation in knowledge tracing on ASSISTment 2009. (a) KernelSHAP. (b) Attention.

main. This section explores potential avenues for future research.

Knowledge Structures Enhanced Explanation. Currently, explanations predominantly rely on raw features like historical interactions, overlooking the underlying knowledge structure between exercises. Consequently, they fail to elucidate the mutual influence between two different exercises sharing the same knowledge concepts or problem schemas[111]. Both educators and learners prefer explanations that incorporate finer-grained knowledge structures in knowledge tracing. For example, when a student provides an incorrect answer to an exercise, a more insightful explanation would attribute the error to a lack of mastery in a related concept, going beyond simply highlighting past incorrect exercises. Despite these preferences, current explanations primarily depend on raw data, leaving the integration of the educational theory into the explanation generation process largely unaddressed.

Multimodal Data Fusion Enhanced Explanation. Presently, explanations predominantly rely on interactions between students and exercises, pinpointing which previous interactions significantly influence current predictions. However, the singular sources of these explanations pose a risk of unreliability. Additional data, including exercise text, time intervals between interactions, relationships between exercises or knowledge concepts, and exercise difficulty, all have the potential to contribute valuable insights. Explanations sourced from diverse channels can mutually reinforce each other, thereby amplifying the credibility of the overall explanations. Consequently, the strategic fusion of multimodal data emerges as a potent approach to elevating the credibility of explanations.

Evaluation in Knowledge Tracing Explanation. Assessing explanations is inherently challenging, primarily due to the lack of a reliable ground truth for a model's internal operations. This challenge is particularly pronounced in knowledge tracing, where expla-

nations aim to elucidate the learning process. Even if explanations accurately capture the internal mechanisms of knowledge tracing models, they may deviate significantly from human perceptions of learning. This discrepancy complicates objective evaluation, highlighting the need for robust evaluation methods for explanations in knowledge tracing.

Recourse for Knowledge Tracing. Recourse[2] describes a user's ability to counteract the adverse effects of model decisions. When a student is identified with deficient understanding of certain concepts, current explainable methods merely offer reasons for these outcomes. Unfortunately, these reasons are often complex and not easily comprehensible, failing to guide students in enhancing their weak knowledge states. Currrently, counterfactual explanations are extensively employed to generate recourse. Hence, it becomes imperative to incorporate explainable methods to effectively produce recourse for knowledge tracing.

8 Conclusions

In this survey, we explored explanation methods in depth. We first provided a comprehensive overview of the development of explanation methods for static and temporal models, highlighting key factors driving their evolution. By analyzing the differences between static and temporal explanation methods and the unique challenges each faces, we underscored the necessity of developing specialized methods for both types. Based on theoretical analysis and experimental findings, we observed the need for explanation methods tailored to specific tasks. Additionally, existing evaluation methods for explanation techniques remain controversial, making the establishment and refinement of evaluation frameworks a crucial area of future research. Such advancements will contribute to standardizing performance comparisons and guiding the selection of appropriate methods. By outlining future research directions, we aim to inspire further advancements and offer readers a valuable guide to explanation methods and their applications.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Lundberg S M, Lee S I. A unified approach to interpreting model predictions. In *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp.4768–4777. DOI: 10.5555/3295222.3295230.

[2] Karimi A H, Barthe G, Schölkopf B, Valera I. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 2023, 55(5): Article No. 95. DOI: 10.1145/3527848.

[3] Kokhlikyan N, Miglani V, Martin M, Wang E, Alsallakh B, Reynolds J, Melnikov A, Kliushkina N, Araya C, Yan S, Reblitz-Richardson O. Captum: A unified and generic model interpretability library for PyTorch. arXiv: 2009.07896, 2020. https://arxiv.org/abs/2009.07896, May 2025.

[4] Li X, Cao C C, Shi Y, Bai W, Gao H, Qiu L, Wang C, Gao Y, Zhang S, Xue X, Chen L. A survey of data-driven and knowledge-aware explainable AI. *IEEE Trans. Knowledge and Data Engineering*, 2022, 34(1): 29–49. DOI: 10.1109/TKDE.2020.2983930.

[5] Yuan H, Yu H, Gui S, Ji S. Explainability in graph neural networks: A taxonomic survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2023, 45(5): 5782–5799. DOI: 10.1109/TPAMI.2022.3204236.

[6] Arrieta A B, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-Lopez S, Molina D, Benjamins R, Chatila R, Herrera F. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 2020, 58: 82–115. DOI: 10.1016/J.INFFUS.2019.12.012.

[7] Samek W, Montavon G, Lapuschkin S, Anders C J, Müller K R. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 2021, 109(3): 247–278. DOI: 10.1109/JPROC.2021.3060483.

[8] Choi E, Bahadori M T, Kulas J A, Schuetz A, Stewart W F, Sun J. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Proc. the 30th International Conference on Neural Information Processing Systems*, Dec. 2016, pp.3512–3520. DOI: 10.5555/3157382.3157490.

[9] Hardt M, Rajkomar A, Flores G, Dai A, Howell M, Corrado G, Cui C, Hardt M. Explaining an increase in predicted risk for clinical alerts. In *Proc. the 2020 ACM Conference on Health, Inference, and Learning*, Apr. 2020, pp.80–89. DOI: 10.1145/3368555.3384460.

[10] Ismail A A, Gunady M, Bravo H C, Feizi S. Benchmarking deep learning interpretability in time series predictions. In *Proc. the 34th International Conference on Neural Information Processing System*, Dec. 2020, Article No. 540. DOI: 10.5555/3495724.3496264.

[11] Tonekaboni S, Joshi S, Campbell K R, Duvenaud D, Goldenberg A. What went wrong and when? Instance-wise feature importance for time-series black-box models. In *Proc. the 34th International Conference on Neural Information Processing Systems*, Dec. 2020, Article No. 68. DOI: 10.5555/3495724.3495792.

[12] Ni J, Chen Z, Cheng W, Zong B, Song D, Liu Y, Zhang X, Chen H. Interpreting convolutional sequence model by learning local prototypes with adaptation regularization. In *Proc. the 30th ACM International Conference on*

Information & Knowledge Management, Nov. 2021, pp.1366–1375. DOI: 10.1145/3459637.3482355.

[13] Leung K K, Rooke C, Smith J, Zuberi S, Volkovs M. Temporal dependencies in feature importance for time series prediction. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[14] Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In *Proc. the 34th International Conference on Machine Learning*, Aug. 2017, pp.3319–3328. DOI: 10.5555/3305890.3306024.

[15] Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proc. the 2nd International Conference on Learning Representations*, Apr. 2014.

[16] Smilkov D, Thorat N, Kim B, Viégas F B, Wattenberg M. SmoothGrad: Removing noise by adding noise. arXiv: 1706.03825, 2017. https://arxiv.org/abs/1706.03825, May 2025.

[17] Kapishnikov A, Venugopalan S, Avci B, Wedin B, Terry M, Bolukbasi T. Guided integrated gradients: An adaptive path method for removing noise. In *Proc. the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2021, pp.5048–5056. DOI: 10.1109/CVPR46437.2021.00501.

[18] Ismail A A, Feizi S, Bravo H C. Improving deep learning interpretability by saliency guided training. In *Proc. the 35th International Conference on Neural Information Processing Systems*, Dec. 2021, pp.26726–26739. DOI: 10.5555/3540261.3542308.

[19] Lundström D D, Huang T, Razaviyayn M. A rigorous study of integrated gradients method and extensions to internal neuron attributions. In *Proc. the 39th International Conference on Machine Learning*, Jul. 2022, pp.14485–14508.

[20] Bach S, Binder A, Montavon G, Klauschen F, Müller K R, Samek W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One*, 2015, 10(7): e0130140. DOI: 10.1371/journal.pone.0130140.

[21] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. In *Proc. the 34th International Conference on Machine Learning*, Aug. 2017, pp.3145–3153. DOI: 10.5555/3305890.3306006.

[22] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks. In *Proc. the 13th European Conference on Computer Vision*, Sept. 2014, pp.818–833. DOI: 10.1007/978-3-319-10590-1_53.

[23] Suresh H, Hunt N, Johnson A, Celi L A, Szolovits P, Ghassemi M. Clinical intervention prediction and understanding using deep networks. arXiv: 1705.08498, 2017. https://arxiv.org/abs/1705.08498, May 2025.

[24] Dabkowski P, Gal Y. Real time image saliency for black box classifiers. In *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp.6970–6979. DOI: 10.5555/3295222.3295440.

[25] Fong R C, Vedaldi A. Interpretable explanations of black boxes by meaningful perturbation. In *Proc. the 2017 IEEE International Conference on Computer Vision*, Oct. 2017, pp.3449–3457. DOI: 10.1109/ICCV.2017.371.

[26] Fong R, Patrick M, Vedaldi A. Understanding deep networks via extremal perturbations and smooth masks. In *Proc. the 2019 IEEE/CVF International Conference on Computer Vision*, Oct. 27–Nov. 2, 2019, pp.2950–2958. DOI: 10.1109/ICCV.2019.00304.

[27] Covert I C, Lundberg S, Lee S I. Understanding global feature contributions with additive importance measures. In *Proc. the 34th International Conference on Neural Information Processing Systems*, Dec. 2020, Article No. 1444. DOI: 10.5555/3495724.3497168.

[28] Agarwal C, Nguyen A. Explaining image classifiers by removing input features using generative models. In *Proc. the 15th Asian Conference on Computer Vision*, Nov. 30–Dec. 4, 2020, pp.101–118. DOI: 10.1007/978-3-030-69544-6_7.

[29] Akhtar N, Jalwana M A A K, Bennamoun M, Mian A. Attack to fool and explain deep networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2022, 44(10): 5980–5995. DOI: 10.1109/TPAMI.2021.3083769.

[30] Ren J, Zhou Z, Chen Q, Zhang Q. Can we faithfully represent absence states to compute shapley values on a DNN? In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[31] Zintgraf L M, Cohen T S, Adel T, Welling M. Visualizing deep neural network decisions: Prediction difference analysis. In *Proc. the 5th International Conference on Learning Representations*, Apr. 2017.

[32] Fel T, Cadène R, Chalvidal M, Cord M, Vigouroux D, Serre T. Look at the variance! Efficient black-box explanations with sobol-based sensitivity analysis. In *Proc. the 35th International Conference on Neural Information Processing Systems*, Dec. 2021, pp.26005–26014. DOI: 10.5555/3540261.3542252.

[33] Sobol' I M. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, 2001, 55(1/2/3): 271–280. DOI: 10.1016/S0378-4754(00)00270-6.

[34] Ribeiro M T, Singh S, Guestrin C. "Why should I trust you?": Explaining the predictions of any classifier. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp.1135–1144. DOI: 10.1145/2939672.2939778.

[35] Visani G, Bagli E, Chesani F. OptiLIME: Optimized LIME explanations for diagnostic computer algorithms. In *Proc. the 2020 CIKM Workshops*, Oct. 2020.

[36] Zhou Z, Hooker G, Wang F. S-LIME: Stabilized-LIME for model explanation. In *Proc. the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Aug. 2021, pp.2429–2438. DOI: 10.1145/3447548.3467274.

[37] Jain S, Wallace B C. Attention is not explanation. In *Proc. the 2019 Conference of the North American Chap-*

ter of the Association for Computational Linguistics: Human Language Technologies, Jun. 2019, pp.3543–3556. DOI: 10.18653/V1/N19-1357.

[38] Wiegreffe S, Pinter Y. Attention is not not explanation. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Nov. 2019, pp.11–20. DOI: 10.18653/V1/D19-1002.

[39] Serrano S, Smith N A. Is attention interpretable? In *Proc. the 57th Conference of the Association for Computational Linguistics*, Jul. 2019, pp.2931–2951. DOI: 10.18653/V1/P19-1282.

[40] Alvarez-Melis D, Jaakkola T S. Towards robust interpretability with self-explaining neural networks. In *Proc. the 32nd International Conference on Neural Information Processing Systems*, Dec. 2018, pp.7786–7795. DOI: 10.5555/3327757.3327875.

[41] Liu Y, Li H, Guo Y, Kong C, Li J, Wang S. Rethinking attention-model explainability through faithfulness violation test. In *Proc. the 39th International Conference on Machine Learning*, Jul. 2022, pp.13807–13824.

[42] Chefer H, Gur S, Wolf L. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proc. the 2021 IEEE/CVF International Conference on Computer Vision*, Oct. 2021, pp.387–396. DOI: 10.1109/ICCV48922.2021.00045.

[43] Ron T, Hazan T. Dual decomposition of convex optimization layers for consistent attention in medical images. In *Proc. the 39th International Conference on Machine Learning*, Jul. 2022, pp.18754–18769.

[44] Kim B, Wattenberg M, Gilmer J, Cai C, Wexler J, Viégas F, Sayres R. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proc. the 35th International Conference on Machine Learning*, Jul. 2018, pp.2668–2677.

[45] Ghorbani A, Wexler J, Zou J, Kim B. Towards automatic concept-based explanations. In *Proc. the 33rd International Conference on Neural Information Processing Systems*, Dec. 2019, pp.9277–9286.

[46] Zhang R, Madumal P, Miller T, Ehinger K A, Rubinstein B I P. Invertible concept-based explanations for CNN models with non-negative concept activation vectors. In *Proc. the 35th AAAI Conference on Artificial Intelligence*, Feb. 2021, pp.11682–11690. DOI: 10.1609/AAAI.V35I13.17389.

[47] Parekh J, Mozharovskyi P, d'Alché-Buc F. A framework to learn with interpretation. In *Proc. the 35th Conference on Neural Information Processing Systems*, Dec. 2021, pp.24273–24285.

[48] Barbiero P, Ciravegna G, Giannini F, Lió P, Gori M, Melacci S. Entropy-based logic explanations of neural networks. In *Proc. the 36th AAAI Conference on Artificial Intelligence*, Feb. 22–Mar. 1, 2022, pp.6046–6054. DOI: 10.1609/AAAI.V36I6.20551.

[49] Huang J, Mishra A, Kwon B C, Bryan C. ConceptExplainer: Interactive explanation for deep neural networks from a concept perspective. *IEEE Trans. Visualization and Computer Graphics*, 2023, 29(1): 831–841. DOI: 10.1109/TVCG.2022.3209384.

[50] Tran T Q, Fukuchi K, Akimoto Y, Sakuma J. Unsupervised causal binary concepts discovery with VAE for blackbox model explanation. In *Proc. the 36th AAAI Conference on Artificial Intelligence*, Feb. 22–Mar. 1, 2022, pp.9614–9622. DOI: 10.1609/AAAI.V36I9.21195.

[51] Renkl A. Toward an instructionally oriented theory of example-based learning. *Cognitive Science*, 2014, 38(1): 1–37. DOI: 10.1111/COGS.12086.

[52] Wachter S, Mittelstadt B, Russell C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. arXiv: 1711.00399, 2017. https://arxiv.org/abs/1711.00399, May 2025.

[53] Dhurandhar A, Chen P Y, Luss R, Tu C C, Ting P, Shanmugam K, Das P. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Proc. the 32nd International Conference on Neural Information Processing Systems*, Dec. 2018, pp.590–601. DOI: 10.5555/3326943.3326998.

[54] Kanamori K, Takagi T, Kobayashi K, Arimura H. DACE: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *Proc. the 29th International Joint Conference on Artificial Intelligence*, Jan. 2021, pp.2855–2862. DOI: 10.24963/IJCAI.2020/395.

[55] Ghandeharioun A, Kim B, Li C L, Jou B, Eoff B, Picard R W. DISSECT: Disentangled simultaneous explanations via concept traversals. In *Proc. the 10th International Conference on Learning Representations*, Apr. 2022.

[56] Koh P W, Nguyen T, Tang Y S, Mussmann S, Pierson E, Kim B, Liang P. Concept bottleneck models. In *Proc. the 37th International Conference on Machine Learning*, Jul. 2020, pp.5338–5348.

[57] Lampert C H, Nickisch H, Harmeling S. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp.951–958. DOI: 10.1109/CVPR.2009.5206594.

[58] Williamson D P, Shmoys D B. The Design of Approximation Algorithms. Cambridge University Press, 2011. DOI: 10.1017/CBO9780511921735.

[59] Goyal Y, Wu Z, Ernst J, Batra D, Parikh D, Lee S. Counterfactual visual explanations. In *Proc. the 36th International Conference on Machine Learning*, Jun. 2019, pp.2376–2384.

[60] Russell C. Efficient search for diverse coherent explanations. In *Proc. the 2019 Conference on Fairness, Accountability, and Transparency*, Jan. 2019, pp.20–28. DOI: 10.1145/3287560.3287569.

[61] Schleich M, Geng Z, Zhang Y, Suciu D. GeCo: Quality counterfactual explanations in real time. *Proceedings of*

the VLDB Endowment, 2021, 14(9): 1681–1693. DOI: 10.14778/3461535.3461555.

[62] Mothilal R K, Sharma A, Tan C. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proc. the 2020 Conference on Fairness, Accountability, and Transparency*, Jan. 2020, pp.607–617. DOI: 10.1145/3351095.3372850.

[63] Slack D, Hilgard S, Lakkaraju H, Singh S. Counterfactual explanations can be manipulated. In *Proc. the 35th International Conference on Neural Information Processing Systems*, Dec. 2021, pp.62–75. DOI: 10.5555/3540261.3540267.

[64] Karimi A H, von Kügelgen J, Schölkopf B, Valera I. Algorithmic recourse under imperfect causal knowledge: A probabilistic approach. In *Proc. the 34th International Conference on Neural Information Processing Systems*, Dec. 2020, Article No. 23. DOI: 10.5555/3495724.3495747.

[65] Chen C, Li O, Tao D, Barnett A J, Su J, Rudin C. *This* looks like *that*: Deep learning for interpretable image recognition. In *Proc. the 33rd International Conference on Neural Information Processing Systems*, Dec. 2019, pp.8930–8941. DOI: 10.5555/3454287.3455088.

[66] Li O, Liu H, Chen C, Rudin C. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, Feb. 2018, pp.3530–3537. DOI: 10.1609/AAAI.V32I1.11771.

[67] Rymarczyk D, Struski Ł, Tabor J, Zielinski B. ProtoPShare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proc. the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Aug. 2021, pp.1420–1430. DOI: 10.1145/3447548.3467245.

[68] Nauta M, van Bree R, Seifert C. Neural prototype trees for interpretable fine-grained image recognition. In *Proc. the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2021, pp.14928–14938. DOI: 10.1109/CVPR46437.2021.01469.

[69] Nauta M, Jutte A, Provoost J C, Seifert C. This looks like that, because ... Explaining prototypes for interpretable image recognition. In *Proc. the 2021 International Workshops of ECML PKDD 2021 on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Sept. 2021, pp.441–456. DOI: 10.1007/978-3-030-93736-2_34.

[70] Kim B, Khanna R, Koyejo O. Examples are not enough, learn to criticize! Criticism for interpretability. In *Proc. the 30th International Conference on Neural Information Processing Systems*, Dec. 2016, pp.2288–2296. DOI: 10.5555/3157096.3157352.

[71] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, 1997, 9(8): 1735–1780. DOI: 10.1162/NECO.1997.9.8.1735.

[72] Crabbé J, van der Schaar M. Explaining time series predictions with dynamic masks. In *Proc. the 38th International Conference on Machine Learning*, Jul. 2021, pp.2166–2177.

[73] Murdoch W J, Szlam A. Automatic rule extraction from long short term memory networks. In *Proc. the 5th International Conference on Learning Representations*, Apr. 2017.

[74] Murdoch W J, Liu P J, Yu B. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *Proc. the 6th International Conference on Learning Representations*, Apr. 30–May 3, 2018.

[75] Guo T, Lin T, Antulov-Fantulin N. Exploring interpretable LSTM neural networks over multi-variable data. In *Proc. the 36th International Conference on Machine Learning*, Jun. 2019, pp.2494–2504.

[76] Zhang Y, Yang X, Ivy J S, Chi M. ATTAIN: Attention-based time-aware LSTM networks for disease progression modeling. In *Proc. the 28th International Joint Conference on Artificial Intelligence*, Aug. 2019, pp.4369–4375. DOI: 10.24963/IJCAI.2019/607.

[77] Bento J, Saleiro P, Cruz A F, Figueiredo M A T, Bizarro P. TimeSHAP: Explaining recurrent models through sequence perturbations. In *Proc. the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Aug. 2021, pp.2565–2573. DOI: 10.1145/3447548.3467166.

[78] Sood A, Craven M. Feature importance explanations for temporal black-box models. In *Proc. the 36th AAAI Conference on Artificial Intelligence*, Feb. 22–Mar. 1, 2022, pp.8351–8360. DOI: 10.1609/AAAI.V36I8.20810.

[79] Jacovi A, Goldberg Y. Aligning faithful interpretations with their social attribution. *Trans. Association for Computational Linguistics*, 2021, 9: 294–310. DOI: 10.1162/TACL_A_00367.

[80] Hase P, Xie H, Bansal M. The out-of-distribution problem in explainability and search methods for feature importance explanations. In *Proc. the 35th International Conference on Neural Information Processing Systems*, Dec. 2021, pp.3650–3666. DOI: 10.5555/3540261.3540540.

[81] Parvatharaju P S, Doddaiah R, Hartvigsen T, Rundensteiner E A. Learning saliency maps to explain deep time series classifiers. In *Proc. the 30th ACM International Conference on Information & Knowledge Management*, Nov. 2021, pp.1406–1415. DOI: 10.1145/3459637.3482446.

[82] Kolodner J L. An introduction to case-based reasoning. *Artificial Intelligence Review*, 1992, 6(1): 3–34. DOI: 10.1007/BF00155578.

[83] Ming Y, Xu P, Qu H, Ren L. Interpretable and steerable sequence learning via prototypes. In *Proc. the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug. 2019, pp.903–913. DOI: 10.1145/3292500.3330908.

[84] Ming Y, Xu P, Cheng F, Qu H, Ren L. ProtoSteer: Steering deep sequence model with prototypes. *IEEE*

Trans. Visualization and Computer Graphics, 2020, 26(1): 238–248. DOI: 10.1109/TVCG.2019.2934267.

[85] Ma D, Wang Z, Xie J, Yu Z, Guo B, Zhou X. Modeling multivariate time series via prototype learning: A multi-level attention-based perspective. In *Proc. the 2020 IEEE International Conference on Bioinformatics and Biomedicine*, Dec. 2020, pp.687–693. DOI: 10.1109/BIBM49941.2020.9313406.

[86] Pandey S, Karypis G. A self attentive model for knowledge tracing. In *Proc. the 12th International Conference on Educational Data Mining*, Jul. 2019.

[87] Ghosh A, Heffernan N, Lan A S. Context-aware attentive knowledge tracing. In *Proc. the 26th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Jul. 2020, pp.2330–2339. DOI: 10.1145/3394486.3403282.

[88] Pandey S, Srivastava J. RKT: Relation-aware self-attention for knowledge tracing. In *Proc. the 29th ACM International Conference on Information & Knowledge Management*, Oct. 2020, pp.1205–1214. DOI: 10.1145/3340531.3411994.

[89] Vie J, Kashima H. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proc. the 33rd AAAI Conference on Artificial Intelligence*, Jan. 27–Feb. 1, 2019, pp.750–757. DOI: 10.1609/AAAI.V33I01.3301750.

[90] Pu S, Converse G, Huang Y. Deep performance factors analysis for knowledge tracing. In *Proc. the 22nd International Conference on Artificial Intelligence in Education*, Jun. 2021, pp.331–341. DOI: 10.1007/978-3-030-78292-4_27.

[91] Chiang A Y. Generalized additive models: An introduction with R. *Technometrics*, 2007, 49(3): 360–361. DOI: 10.1198/TECH.2007.S505.

[92] Yang H, Cheung L P. Implicit heterogeneous features embedding in deep knowledge tracing. *Cognitive Computation*, 2018, 10(1): 3–14. DOI: 10.1007/S12559-017-9522-0.

[93] Wang C, Sahebi S, Zhao S, Brusilovsky P, Moraes L O. Knowledge tracing for complex problem solving: Granular rank-based tensor factorization. In *Proc. the 29th ACM Conference on User Modeling, Adaptation and Personalization*, Jun. 2021, pp.179–188. DOI: 10.1145/3450613.3456831.

[94] Papalexakis E E. Automatic unsupervised tensor mining with quality assessment. In *Proc. the 2016 SIAM International Conference on Data Mining*, May 2016, pp.711–719. DOI: 10.1137/1.9781611974348.80.

[95] Wang D, Lu Y, Zhang Z, Chen P. A generic interpreting method for knowledge tracing models. In *Proc. the 23rd International Conference on Artificial Intelligence in Education*, Jul. 2022, pp.573–580. DOI: 10.1007/978-3-031-11644-5_51.

[96] Mandalapu V, Gong J, Chen L. Do we need to go deep? Knowledge tracing with big data. arXiv: 2101.08349, 2021. https://arxiv.org/abs/2101.08349, May 2025.

[97] Lu Y, Wang D, Chen P, Meng Q, Yu S. Interpreting deep learning models for knowledge tracing. *International Journal of Artificial Intelligence in Education*, 2023, 33(3): 519–542. DOI: 10.1007/S40593-022-00297-Z.

[98] Ding X, Larson E C. On the interpretability of deep learning based models for knowledge tracing. arXiv: 2101.11335, 2021. https://arxiv.org/abs/2101.11335, May 2025.

[99] Minn S, Yu Y, Desmarais M C, Zhu F, Vie J J. Deep knowledge tracing and dynamic student classification for knowledge tracing. In *Proc. the 2018 IEEE International Conference on Data Mining*, Nov. 2018, pp.1182–1187. DOI: 10.1109/ICDM.2018.00156.

[100] Johns J, Mahadevan S, Woolf B. Estimating student proficiency using an item response theory model. In *Proc. the 8th International Conference on Intelligent Tutoring Systems*, Jun. 2006, pp.473–480. DOI: 10.1007/11774303_47.

[101] Yeung C K. Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory. In *Proc. the 12th International Conference on Educational Data Mining*, Jul. 2019.

[102] Chen J, Liu Z, Huang S, Liu Q, Luo W. Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In *Proc. the 37th AAAI Conference on Artificial Intelligence*, Feb. 2023, pp.14196–14204. DOI: 10.1609/AAAI.V37I12.26661.

[103] Su Y, Cheng Z, Luo P, Wu J, Zhang L, Liu Q, Wang S. Time-and-concept enhanced deep multidimensional item response theory for interpretable knowledge tracing. *Knowledge-Based Systems*, 2021, 218: 106819. DOI: 10.1016/J.KNOSYS.2021.106819.

[104] Minn S, Vie J J, Takeuchi K, Kashima H, Zhu F. Interpretable knowledge tracing: Simple and efficient student modeling with causal relations. In *Proc. the 36th AAAI Conference on Artificial Intelligence*, Feb. 22–Mar. 1, 2022, pp.12810–12818. DOI: 10.1609/AAAI.V36I11.21560.

[105] Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. *Machine Learning*, 1997, 29(2): 131–163. DOI: 10.1023/A:1007465528199.

[106] Piech C, Bassen J, Huang J, Ganguli S, Sahami M, Guibas L, Sohl-Dickstein J. Deep knowledge tracing. In *Proc. the 29th International Conference on Neural Information Processing Systems*, Dec. 2015, pp.505–513. DOI: 10.5555/2969239.2969296.

[107] Nakagawa H, Iwasawa Y, Matsuo Y. Graph-based knowledge tracing: Modeling student proficiency using graph neural networks. *Web Intelligence*, 2021, 19(1/2): 87–102. DOI: 10.3233/WEB-210458.

[108] Song X, Li J, Tang Y, Zhao T, Chen Y, Guan Z. JKT: A joint graph convolutional network based deep knowledge tracing. *Information Sciences*, 2021, 580: 510–523. DOI: 10.1016/J.INS.2021.08.100.

[109] Liang K, Meng L, Liu M, Liu Y, Tu W, Wang S, Zhou S, Liu X, Sun F, He K. A survey of knowledge graph

reasoning on graph types: Static, dynamic, and multimodal. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2024, 46(12): 9456–9478. DOI: 10.1109/TPAMI.2024.3417451.

[110] Liu A, Zhang Y. Spatial-temporal dynamic graph convolutional network with interactive learning for traffic forecasting. *IEEE Trans. Intelligent Transportation Systems*, 2024, 25(7): 7645–7660. DOI: 10.1109/TITS.2024.3362145.

[111] Tong H, Wang Z, Zhou Y, Tong S, Han W, Liu Q. Introducing problem schema with hierarchical exercise graph for knowledge tracing. In *Proc. the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2022, pp.405–415. DOI: 10.1145/3477495.3532004.

[112] Zhang D, Wang L, Zhang L, Dai B T, Shen H T. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2020, 42(9): 2287–2305. DOI: 10.1109/TPAMI.2019.2914054.

[113] Hoffman R R, Mueller S T, Klein G, Litman J. Metrics for explainable AI: Challenges and prospects. arXiv: 1812.04608, 2018. https://arxiv.org/abs/1812.04608, May 2025.

[114] Alvarez-Melis D, Jaakkola T S. On the robustness of interpretability methods. arXiv: 1806.08049, 2018. https://arxiv.org/abs/1806.08049, May 2025.

[115] Adebayo J, Gilmer J, Muelly M, Goodfellow I, Hardt M, Kim B. Sanity checks for saliency maps. In *Proc. the 32nd International Conference on Neural Information Processing Systems*, Dec. 2018, pp.9525–9536. DOI: 10.5555/3327546.3327621.

[116] Rebuffi S A, Fong R, Ji X, Vedaldi A. There and back again: Revisiting backpropagation saliency methods. In *Proc. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp.8836–8845. DOI: 10.1109/CVPR42600.2020.00886.

[117] Nie W, Zhang Y, Patel A. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *Proc. the 35th International Conference on Machine Learning*, Jul. 2018, pp.3809–3818.

[118] Narayanan M, Chen E, He J, Kim B, Gershman S, Doshi-Velez F. How do humans understand explanations from machine learning systems? An evaluation of the human-interpretability of explanation. arXiv: 1802.00682, 2018. https://arxiv.org/abs/1802.00682, May 2025.

[119] Johnson A E W, Pollard T J, Shen L, Lehman L W H, Feng M, Ghassemi M, Moody B, Szolovits P, Anthony Celi L, Mark R G. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 2016, 3(1): 160035. DOI: 10.1038/sdata.2016.35.

[120] Yeh C K, Kim B, Arik S Ö, Li C L, Pfister T, Ravikumar P. On completeness-aware concept-based explanations in deep neural networks. In *Proc. the 34th International Conference on Neural Information Processing Systems*, Dec. 2020, Article No. 1726. DOI: 10.5555/3495724.3497450.

[121] Hooker S, Erhan D, Kindermans P J, Kim B. A benchmark for interpretability methods in deep neural networks. In *Proc. the 33rd International Conference on Neural Information Processing Systems*, Dec. 2019, pp.9737–9748. DOI: 10.5555/3454287.3455160.

[122] Zhang J, Bargal S A, Lin Z, Brandt J, Shen X, Sclaroff S. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 2018, 126(10): 1084–1102. DOI: 10.1007/S11263-017-1059-X.

[123] Sundararajan M, Najmi A. The many shapley values for model explanation. In *Proc. the 37th International Conference on Machine Learning*, Jul. 2020, pp.9269–9278. DOI: 10.5555/3524938.3525797.

[124] Yudelson M V, Koedinger K R, Gordon G J. Individualized Bayesian knowledge tracing models. In *Proc. the 16th International Conference on Artificial Intelligence in Education*, Jul. 2013, pp.171–180. DOI: 10.1007/978-3-642-39112-5_18.

[125] Huang C Q, Huang Q H, Huang X, Wang H, Li M, Lin K J, Chang Y. XKT: Towards explainable knowledge tracing model with cognitive learning theories for questions of multiple knowledge concepts. *IEEE Trans. Knowledge and Data Engineering*, 2024, 36(11): 7308–7325. DOI: 10.1109/TKDE.2024.3418098.

[126] Huang C, Wei H, Huang Q, Jiang F, Han Z, Huang X. Learning consistent representations with temporal and causal enhancement for knowledge tracing. *Expert Systems with Applications*, 2024, 245: 123128. DOI: 10.1016/J.ESWA.2023.123128.

[127] Wu Z, Huang L, Huang Q, Huang C, Tang Y. SGKT: Session graph-based knowledge tracing for student performance prediction. *Expert Systems with Applications*, 2022, 206: 117681. DOI: 10.1016/J.ESWA.2022.117681.

[128] Weston J, Chopra S, Bordes A. Memory networks. In *Proc. the 3rd International Conference on Learning Representations*, May 2015.

[129] Zhang J, Shi X, King I, Yeung D. Dynamic key-value memory networks for knowledge tracing. In *Proc. the 26th International Conference on World Wide Web*, Apr. 2017, pp.765–774. DOI: 10.1145/3038912.3052580.

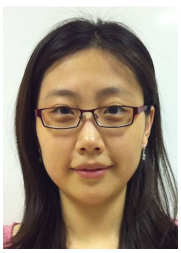
[130] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In *Proc. the 5th International Conference on Learning Representations*, Apr. 2017.

[131] Liu M, Liang K, Zhao Y, Tu W, Zhou S, Gan X, Liu X, He K. Self-supervised temporal graph learning with temporal and structural intensity alignment. *IEEE Trans. Neural Networks and Learning Systems*, 2025, 36(4): 6355–6367. DOI: 10.1109/TNNLS.2024.3386168.

[132] Lopardo G, Precioso F, Garreau D. Attention meets post-hoc interpretability: A mathematical perspective. In *Proc. the 41st International Conference on Machine Learning*, Jul. 2024, Article No. 1331. DOI: 10.5555/3692070.3693401.

Fan Li received his M.S. degree in computer science and technology from Northeastern University, Shenyang, in 2019. He is currently a Ph.D. candidate at the School of Computer Science and Engineering, Northeastern University, Shenyang. His research interests include machine learning, deep learning, knowledge tracing, and explainable artificial intelligence.

Tian-Cheng Zhang received his Ph.D. degree in computer software and theory from Northeastern University, Shenyang, in 2008. He is currently a professor at the School of Computer Science and Engineering, Northeastern University, Shenyang. His research interests include big data analysis, spatiotemporal data management, and deep learning.



Yi-Fang Yin received her B.E. degree in computer science and technology from Northeastern University, Shenyang, in 2011, and her Ph.D. degree in computer science from National University of Singapore, Singapore, in 2016. She is currently a scientist in the Machine Intellection Department, Institute for Infocomm Research, A*STAR. Her research interests include machine learning, spatiotemporal data mining, and multimodal analysis in multimedia.

Di Fan is currently a Ph.D. candidate at the School of Computer Science and Engineering, Northeastern University, Shenyang. Her research interests include data science, knowledge graphs, recommendation systems, and AI for science.

Ming-He Yu received her B.S. degree in computer science and technology from Northeastern University, Shenyang, in 2012, and her Ph.D. degree in computer science and technology from Tsinghua University, Beijing, in 2018. She is currently an associate professor with Software College, Northeastern University, Shenyang. Her research interests include databases, information retrieval, and intelligent education.

Ge Yu received his Ph.D. degree in computer science from Kyushu University, Fukuoka, in 1996. He is currently a professor and a Ph.D. supervisor at Northeastern University, Shenyang. His research interests include distributed and parallel databases, OLAP and data warehousing, data integration, and graph data management.