

A Survey on Behavioral Data Representation Learning

Xi Chen^{ID}, Siwei Zhang^{ID}, Kangxiang Jia^{ID}, Sijia Peng^{ID}, Yuxuan Wang^{ID}, Liang Su^{ID}, Xinyi Huang^{ID},
Jiarong Xu^{ID}, Yun Xiong^{ID} and Philip S. Yu^{ID}, *Fellow, IEEE*

Abstract—Behavioral data, reflecting dynamic and complex interactions among entities, are pivotal for advancing multidisciplinary research and practical applications. Effective modeling and representation of behavioral data facilitate enhanced understanding, predictive analytics, and informed decision-making across diverse domains. This paper presents a comprehensive taxonomy of behavioral data representation learning methods, categorized by data modalities: tabular data, event sequences, dynamic graphs, and natural language. Within each category, we further dissect methods based on distinct modeling strategies and capabilities, and provide detailed reviews of their developments. Additionally, we extensively discuss significant downstream applications, datasets, and benchmarks, highlighting their roles in guiding methodological development and evaluating performance. To support further exploration in behavioral data representation learning, we release a continuously maintained repository at [GitHub](#) that curates the methods and papers covered in this survey.

Index Terms—Behavioral Data, Representation Learning, Time Series, Dynamic Graph, LLM

I. INTRODUCTION

BEHAVIORAL data capture intricate and dynamic interactions between individuals and their environments, serving as foundational evidence across multiple disciplines such as psychology, neuroscience, social sciences, economics, and artificial intelligence. They are inherently rich, relational, and dynamic, often described in various data modalities such as tabular data, event sequences, dynamic graphs, and textual data.

Accurate modeling of behavioral data is crucial as it enables a deeper understanding of underlying behavioral mechanisms, facilitates predictive analytics, and informs practical interventions in diverse areas such as marketing, policy-making, financial risk control, and social network analysis. Given its complexity, behavioral data often exhibit high dimensionality, temporal dependencies, and relational structures, necessitating sophisticated representation learning methods.

Xi Chen, Siwei Zhang, Kangxiang Jia, Sijia Peng, Liang Su, Xinyi Huang, Yun Xiong are with the Shanghai Key Laboratory of Data Science, College of Computer Science and Artificial Intelligence, Fudan University, Shanghai, China. E-mail: {x_chen21, siweizhang, kangxiangjia, sijiapeng, liangs, xinyihuang25}@m.fudan.edu.cn, yunx@fudan.edu.cn,

Yuxuan Wang is with the Shanghai University of Finance and Economics, Shanghai, China. E-mail: wang.yuxuan1@stu.sufe.edu.cn

Jiarong Xu is with the School of Management, Fudan University, Shanghai, China. E-mail: jiarongxu@fudan.edu.cn

Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, USA. E-mail: psyu@uic.edu

(Corresponding author: Yun Xiong.)

Representation learning of behavioral data involves transforming raw behavioral inputs into structured, compact representations, i.e., embeddings, that capture essential behavioral patterns suitable for computational models. Unlike traditional feature extraction methods that rely heavily on manual intervention, representation learning leverages advanced machine learning and deep learning techniques to automatically distill informative features directly from complex datasets.

To provide a systematic understanding of behavioral data representation learning, we propose a taxonomy structured around the different forms in which behavioral data can be represented: **(1) Tabular Data:** This form represents behaviors as structured tables, characterized by static or aggregated features such as demographic attributes, user preferences, transaction frequencies, or sensor readings. Tabular data often enables efficient processing and straightforward interpretability, making it ideal for applications like intelligent monitoring, risk assessment, and market analytics. Tabular data modeling has undergone significant evolution, progressing from early tree-based methods to a competitive landscape dominated by both deep learning approaches and tree-based models. More recently, the emergence of large language models has introduced a new paradigm, further advancing the field. **(2) Event Sequences:** Behaviors are frequently recorded as sequential events, each associated with a timestamp, providing insights into temporal dependencies, user trajectories, and action progressions. Common examples include clickstreams from websites, transaction logs, medical events, and sensor data sequences. Representation learning methods for event sequences usually utilize sequential modeling approaches, such as Recurrent Neural Networks (RNNs) [1] and Transformers [2], to capture inherent sequential dynamics and temporal patterns. **(3) Dynamic Graphs:** Many behavioral phenomena naturally form dynamic relational structures, representing interactions among entities evolving over time, such as social networks, financial transactions, and collaborative systems. Dynamic graphs capture complex relational dynamics, enabling detailed analysis of interactions and structural changes. Representation learning methods for dynamic graphs employ Graph Neural Networks (GNNs) [3], integrated with recurrent models [1], [4], attention mechanisms [2], and memory architectures [5], to accurately encode evolving graph structures and temporal dependencies. **(4) Textual Data:** Behavioral data often manifest through textual interactions and documents, including online reviews, chat logs, social media content, and textual financial disclosures. Textual data inherently contains

rich semantic information reflecting sentiments, intents, and contextual interactions. Representation learning in this domain leverages advanced large-scale pre-trained Language Models (LMs), such as BERT, GPT [6], and specialized variants like FinBERT [7], to effectively extract contextual and semantic insights, facilitating sentiment analysis, intention recognition, and behavioral prediction tasks.

Within each form, we further subdivide representation learning methods based on specific architectural choices, temporal modeling strategies, or the integration of external domain knowledge and multi-modal fusion capabilities.

Additionally, we extensively explore significant downstream applications, including personalized recommendation systems, financial risk assessment, anomaly and fraud detection, social network analytics, adaptive educational technologies, health-care monitoring, and behavioral interventions. Each application domain underscores the practical implications, efficacy, and adaptability of behavioral data representation learning methodologies. To facilitate research and practical applications, we further provide an overview of commonly used datasets and benchmarks, outlining their characteristics, scales, domains of applicability, and evaluation standards. This comprehensive survey approach enables clearer alignment between methodological choices, application-specific goals, and available data resources. We summarize our contributions as follows:

- **Novel Taxonomy:** We introduce a new and comprehensive taxonomy for behavioral data representation learning. First, from the perspective of data modalities, we categorize representation methods into distinct classes based on the form of the input behavioral data, including tabular data, event sequences, dynamic graphs, and textual data. This modality-based taxonomy helps researchers efficiently select appropriate methods according to data characteristics and application scenarios. Second, we classify methods according to various downstream tasks and applications, enabling clear alignment between methodological choices and specific analytical goals.
- **Comprehensive Methodological Reviews:** We provide a thorough and detailed review of the state-of-the-art methods within each category, highlighting their core principles, underlying assumptions, key innovations, and limitations. Our comprehensive analysis includes critical discussions of model architectures, training paradigms, and evaluation metrics, facilitating a deep understanding of existing approaches and guiding future methodological innovations.
- **Integration of Emerging Technologies:** We critically analyze the impact and potential integration of emerging technologies on behavioral data representation learning. Furthermore, we outline promising future directions and open challenges, emphasizing how interdisciplinary advancements can further enhance behavioral representation learning and expand its applicability and effectiveness across various practical and scientific domains.

II. DEFINITIONS

A. Human Behavior

Human behavior refers to the observable actions or reactions of a person in response to its environment or internal state. Behavior is not a static or isolated event; it emerges from a complex interplay of factors and unfolds over time [8]. This highlights that behavior is inherently relational, arising through interactions between an individual (i.e., the behavioral subject) and its surroundings (i.e., target entities, including other people), and is dynamic, progressing as a sequence of events rather than a single event. For example, in social situations multiple individuals may dynamically interact and display species-specific behaviors, with each person's actions influencing the others' in real time. Behavior is also high-dimensional in nature. Even a single behavioral episode can be described by numerous variables (positions of limbs, neural signals, environmental context, etc.) [9]. This high dimensionality means behavioral data can be extremely rich, but it also poses challenges for analysis and interpretation.

B. Behavioral Data

Behavioral data are the recorded or measured manifestations of behavior, referring to data collected to represent what an individual does. Such data are grounded in observable actions or outputs of the subject. Behavioral data can take many forms depending on how the behavior is observed and logged. The key aspect is that the behavior has been externalized into a digital or structured form that can be stored and analyzed. In early behavioral science, data often came from manual observation notes or simple counts of events, whereas today they are frequently complex digital datasets (potentially combining multiple channels) [10]. Regardless of the format, behavioral data serve as the empirical evidence of behavior, which researchers can analyze to draw inferences about underlying processes or to predict future actions.

C. Representation Learning

Representation learning aims to automatically learn effective and generalizable feature representations from raw data through machine learning or deep learning methods. It transforms high-dimensional, complex, and difficult-to-process raw data into more compact, low-dimensional, and semantically rich vector representations. These learned representations can not only significantly improve the performance of downstream tasks, such as classification and relation prediction, but also reduce the reliance on manual feature engineering during analysis. Representation learning has achieved remarkable success across a wide range of domains, including Computer Vision (CV), Speech Processing, and Natural Language Processing (NLP).

In recent years, with the rapid development and maturation of deep learning research, particularly the emergence of Convolutional Neural Networks (CNNs) [11], RNNs [1], GNNs [3], and self-attention mechanisms [2], the expressive power of representation learning for modeling diverse data types has been greatly enhanced. This progress has further provided powerful tools for the analysis of behavioral data.

D. Behavioral Data Representation Learning

Behavioral data representation learning is the process of encoding raw behavioral data into structured, compact representations (often low-dimensional vectors or embeddings) that are suitable for computational modeling and analysis. It enables models to automatically learn the salient features or abstract factors of the data directly from the data itself, rather than relying on manual feature selection or domain-specific heuristics [12], [13].

In conventional analyses, human experts must design specific behavioral metrics or features *a priori* (for example, deciding to measure the frequency of certain actions or the duration of particular events). As behavioral datasets grow more complex and high-dimensional, this manual process becomes a bottleneck: many hand-crafted features do not capture the rich, hierarchical structure of behavior, and creating new features for each problem is labor-intensive and not scalable. Along with the development of machine learning and deep learning, researchers have started to use these emerging techniques to discover a latent encoding of behavioral patterns without explicit human supervision, which enables models to automatically capture complex, non-linear temporal dependencies, reduce reliance on handcrafted features, and generalize across diverse behavioral scenarios [11], [14].

E. Forms of Behavioral Data and Their Representations

Behavioral data can be represented or constructed in various forms or modalities depending on how it is measured and what aspects of behavior are of interest. Each modality may require different representation learning techniques. We outline common data modalities for behavior and how they capture behavioral information.

Static Feature Vectors: One straightforward way to represent behavior is as a vector of features summarizing an individual's behavioral tendencies. In this modality, the behavior is reduced to a defined set of attributes or metrics. Such a representation treats behavioral data as a point in a multi-dimensional feature space. Historically, many behavioral studies relied on this kind of summarization – for example, recording the number of online purchases, the average session duration on a platform, or the frequency of logins, which yields numeric descriptors that are relatively easy to analyze statistically. The advantage of static feature vectors is their simplicity and interpretability; however, they may omit the temporal dynamics or context of the behavior.

Event Sequences: Human behavior often unfolds as a sequence of discrete, temporally ordered events, which may occur sporadically or in bursts. This could be a sequence of timestamped events or state changes generated over time by one or more sources. Representing behavior as a sequence preserves the temporal dynamics like the order and timing of events, which are crucial for understanding patterns such as routines, progressions of actions, or cause-and-effect in interactions. Event sequence representations enable the use of sequence-modeling techniques, such as Markov Chain Models [15], RNNs [1], [4] or Transformer and its variants [16], to learn temporal patterns in the behavior.

Dynamic Graphs: Some behavioral data are best described not just as a linear sequence but as a set of entities and their interactions evolving over time. Graph-based representations are useful when behavior involves relational structure, such as social interactions or complex event dependencies. In a dynamic graph, nodes usually represent entities, and edges represent interactions or temporal relations between these nodes [17]. By applying dynamic graphs, a complex action sequence becomes a structured graph rather than a simple timeline, and analyzing the behavior can involve graph matching or graph neural network algorithms to identify patterns. Dynamic graph representations can naturally capture concurrent events (when multiple things happen at once) and long-range dependencies.

Multi-modal Representations: Real-world behavior often spans multiple data modalities at once. Many forms of behavioral data are inherently linguistic, such as dialogue transcripts, social media interactions, textual evaluations, or narrative logs of user interactions. These textual behavioral data inherently embed rich semantic information including emotional states, intentions, motivations, and subjective experiences [18]. Moreover, recent advancements in large-scale pre-trained language models, such as GPT models [6], [19], [20], have profoundly changed the landscape [21]. These models enable not only direct analysis of inherently textual behavioral data but also allow the transformation of traditionally non-linguistic behavioral datasets into natural language [18].

III. TAXONOMY OF BEHAVIORAL DATA REPRESENTATION LEARNING

Based on the different modalities of behavioral data introduced in Sec. II-E, this section presents representation learning methods tailored to each modality and provides a finer-grained taxonomy within each category.

A. Tabular Data

Behavioral data can be instantiated in different forms, one of which is tabular data organized in rows and columns, where each row corresponds to an individual instance and each column represents a specific feature or attribute [73]–[76]. In representation learning, such data are typically modeled as static feature vectors, with each instance encoded as a fixed-dimensional vector that aggregates all attribute values. As one of the most prevalent and widely used data types in real-world information systems, tabular data are central to a broad range of domains, including medicine [77], [78], finance [79], [80], e-commerce [81], [82]. Compared to unstructured data such as images or text, tabular data offer clear and interpretable structure, facilitating efficient storage, processing, and analysis. For example, in digital platforms, tabular data are widely used to record user behaviors such as login frequency, browsing histories, purchase transactions, or content interactions; in online education, they capture attendance records, assignment submissions, and test outcomes. With the growing emphasis on data-driven decision-making across industries, tabular behavioral data play an increasingly critical role in tasks such as user profiling, anomaly detection, risk assessment, and personalized

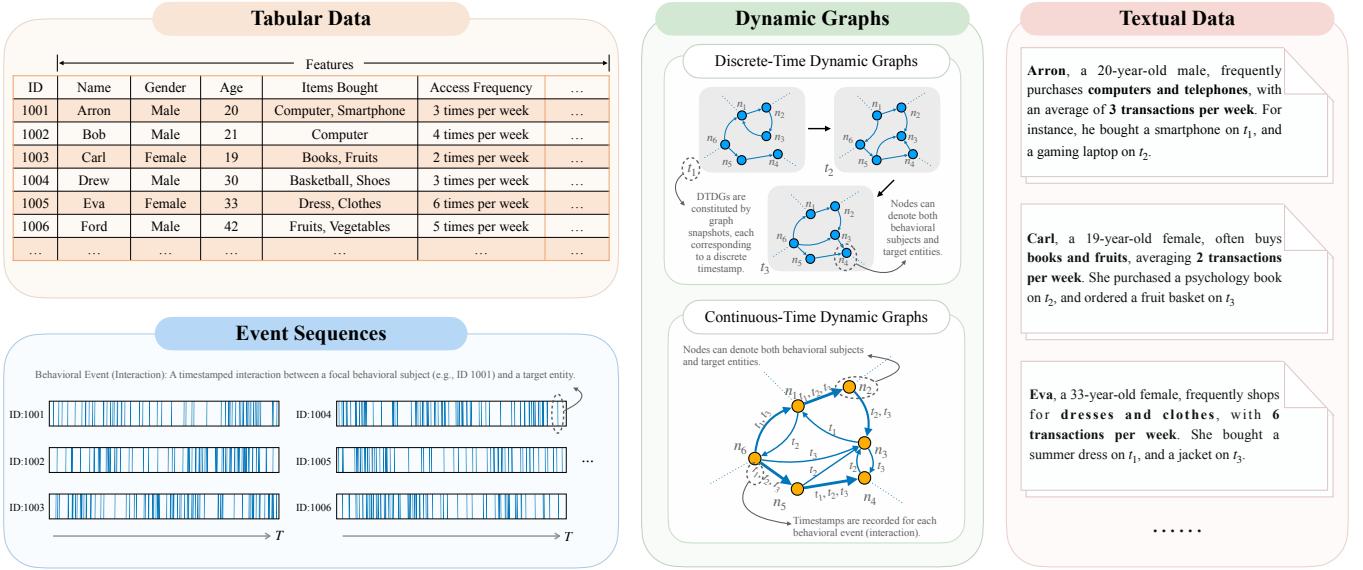


Fig. 1: Taxonomy of behavioral data: behavioral data can take various forms, including tabular data, event sequences, dynamic graphs, and textual data.

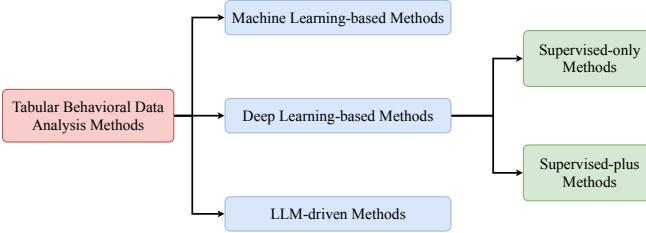


Fig. 2: The dissection of the tabular data representation learning methods.

recommendation. However, despite its seemingly regular structure, behavioral tabular data often exhibit high heterogeneity, sparsity, and domain-specific characteristics, posing significant challenges for effective modeling and analysis. Therefore, a systematic investigation into the representation, modeling strategies, and practical applications of tabular behavioral data are of both theoretical importance and practical relevance for advancing intelligent decision systems and enabling digital transformation across sectors.

Modeling methods for tabular data have seen a progression—from traditional machine learning to deep learning, and more recently, to large foundation models. Traditional tree-based models provided a solid balance between interpretability and robustness, establishing them as a strong baseline for tabular data learning. With the rise of deep learning, neural networks have been explored for tabular data [32], [35], [83], showing promise in areas like feature learning and multi-modal integration despite inherent challenges. More recently, Large Language Models (LLMs) have introduced a new paradigm, enabling tasks such as reasoning and generation over tabular inputs. Overall, tabular data modeling is evolving toward more generalizable and automated solutions.

In this section, we systematically review representative

modeling approaches for tabular behavioral data. Specifically, we categorize existing methods into three major families based on their underlying modeling paradigms: **(1) Machine Learning-based Methods**, **(2) Deep Learning-based Methods**, and **(3) LLM-driven Methods**. For each category, we highlight the core modeling techniques, representative architectures, and recent advancements. All methods are concluded in the Tab. I.

1) Machine Learning-based Methods

While classical non-tree methods such as Support Vector Machines (SVMs) and linear models remain relevant in specific niches—particularly for small-sample regimes or high-dimensional sparse data—their sensitivity to data pre-processing and scalability constraints have restricted their broader adoption. Consequently, over the past decade, tree-based ensemble algorithms such as eXtreme Gradient Boosting (XGBoost) have established themselves as the go-to solutions for tabular data modeling in real-world applications [22], [24], [84]. Beyond their empirical dominance, researchers have actively investigated the theoretical underpinnings of their superiority on structured data [73], [85]. These tree-based ensembles—most notably Gradient Boosting Decision Trees (GBDT) and their optimized implementations like XGBoost [22], LightGBM [23], and CatBoost [24]—are favored for their exceptional accuracy, training efficiency, and robustness against missing values and unscaled categorical features. Even amidst the rapid proliferation of deep learning, GBDT-based models continue to be regarded as the state-of-the-art for general tabular tasks [86], [87].

Recent studies have sought to explain the superior performance of tree-based models on tabular data from both theoretical and empirical perspectives [85]. Works such as Unmasking Trees [88] have shifted attention toward understanding and improving the interpretability of tree-based models by analyzing their internal decision mechanisms in tabular settings.

TABLE I: Tabular data representation learning methods.

Method	Training Paradigm	Model Architecture	Year
Tree-based Methods			
XGBoost [22]	Supervised learning	Tree	2016
LightGBM [23]	Supervised learning	Tree	2017
CatBoost [24]	Supervised learning	Tree	2018
Deep Learning-based Methods			
Wide&Deep [25]	Supervised learning	GLM+MLP	2016
DeepFM [26]	Supervised learning	FM+MLP	2017
xDeepFM [27]	Supervised learning	FM+MLP+CIN	2018
TabNN [28]	Supervised learning	GBDT+MLP	2018
RLN [29]	Supervised learning	MLP	2018
NODE [30]	Supervised learning	NODE	2019
SuperTML [31]	Supervised learning	CNN	2019
TabNet [32]	Supervised + Self-supervised learning	TabNet	2019
DeepGBM [33]	Supervised + Online learning	DeepGBM	2019
NON [34]	Supervised learning	NON	2020
DNF-Net [35]	Supervised learning	DNF-Net	2020
VIME [36]	Self-supervised + Semi-supervised learning	VIME	2020
TabTransformer [37]	Supervised + Semi-supervised learning	Transformer	2020
ARM-Net [38]	Supervised learning	ARM-Net	2021
NPT [39]	Supervised learning	Transformer	2021
Regularized DNNs [40]	Supervised learning	MLP	2021
Boost-GNN [41]	Supervised learning	GBDT+GNN	2021
DNN2LR [42]	Supervised learning	DNN2LR	2021
IGTD [43]	Supervised learning	IGTD+CNN	2021
FT-Transformer [44]	Supervised learning	Transformer	2021
SAINT [45]	Supervised + Self-supervised learning	SAINT	2021
SCARF [46]	Self-supervised learning	MLP	2021
GANDALF [47]	Supervised learning	GFLU	2022
TabDDPM [48]	Unsupervised learning	Diffusion Model	2022
PTab [49]	Supervised + Self-supervised learning	BERT	2022
TabCBM [50]	Supervised learning	MLP	2024
Trompt [51]	Supervised + Prompt learning	MLP	2023
HYTREL [52]	Supervised + Self-supervised learning	HYTREL	2023
ReConTab [53]	Self-supervised	Transformer	2023
XTab [54]	Supervised + Self-supervised learning	Transformer	2023
IATTN [55]	Supervised learning	Transformer	2024
MambaTab [56]	Supervised learning	Mamba	2024
GCondNet [57]	Supervised learning	MLP	2024
BiSShop [58]	Supervised learning	BiSShop	2024
LF-transformer [59]	Supervised learning	Transformer	2024
TabR [60]	Supervised learning	TabR	2024
TP-BERTa [61]	Supervised + Self-supervised learning	BERT	2024
CARTE [62]	Supervised + Self-supervised learning	Transformer	2024
SwitchTab [63]	Self-supervised learning	Transformer	2024
DP-2Stage [64]	Supervised learning	Transformer	2025
TARTE [65]	Supervised + Self-supervised learning	Transformer	2025
TDCoLER [66]	Supervised + Unsupervised learning	Distillation	2025
LLM-driven Methods			
TAPAS [67]	Self-supervised learning	Bert	2020
TAPEX [68]	Supervised + Self-supervised learning	Transformer	2022
TabLLM [69]	Supervised + Self-supervised learning	Transformer	2022
cTBLS [70]	Supervised learning	Transformer	2023
TST-LLM [71]	Self-supervised learning	Transformer	2025
Tabby [72]	Supervised learning	Transformer	2026

On the one hand, tabular datasets often exhibit characteristics such as non-stationarity, feature heterogeneity, and sparsity, which pose significant challenges for traditional neural networks [74]. Tree-based models, compared to neural networks, naturally adapt to these properties, performing implicit feature selection and capturing complex nonlinear relationships. These models also have demonstrated strong robustness in handling categorical variables, outliers, and missing data, and are less prone to overfitting when trained on small datasets [89]. These

advantages have contributed to the widespread adoption and continuous development of tree-based models in both industry and academia.

2) Deep Learning-based Methods

While traditional machine learning methods continue to demonstrate strong performance on structured data, deep learning models are increasingly gaining traction due to their powerful representation learning capabilities. Motivated by their remarkable success in domains such as CV and NLP,

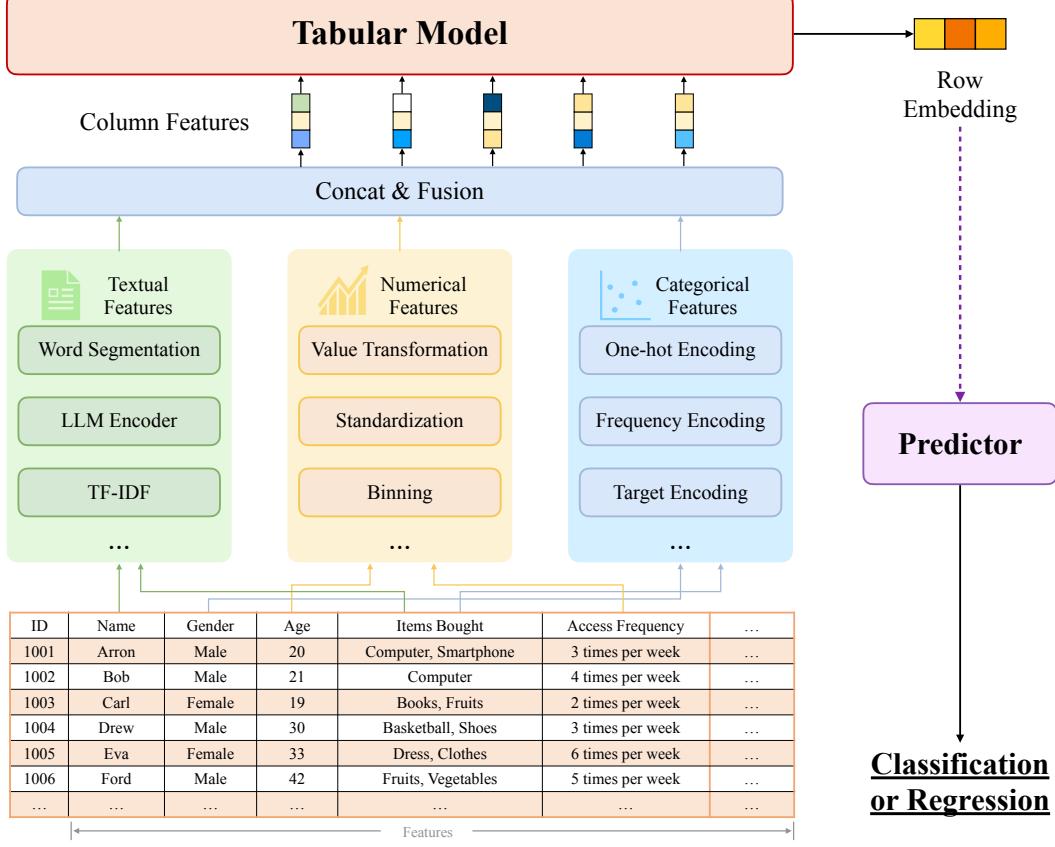


Fig. 3: The pipeline of supervised tabular data representation learning methods.

deep learning approaches have been actively explored for tabular data modeling and analysis. In this section, we provide a comprehensive overview of recent advances in applying deep learning to tabular data, with a particular focus on two key methodological paradigms: **(a) Supervised-only Methods** and **(b) Supervised-plus Methods**.

a) Supervised-only Methods

Supervised learning has long been the dominant paradigm in modeling behavioral tabular data. By leveraging manually annotated labels, supervised approaches are capable of learning explicit mappings between input features and target variables, and have been widely applied in tasks such as user behavior prediction, risk assessment, and recommendation systems. We show the pipeline of supervision methods in Fig. 3.

In the early attempts to apply deep learning to tabular data, researchers primarily focused on integrating traditional machine learning methods with deep learning architectures. For instance, Wide & Deep [25] jointly trains a generalized linear model and a deep neural network, aiming to combine the strengths of memorization and generalization for improved performance in recommendation systems. DeepFM [26] integrates the feature interaction modeling capabilities of Factorization-Machine (FM) with the representation learning power of deep neural networks. xDeepFM [27] further extends this architecture by incorporating three components: an FM module for capturing low-order feature interactions, a Multilayer Perceptron (MLP) module for implicitly modeling

high-order interactions, and a Compressed Interaction Network (CIN) module for explicitly learning high-order feature interactions.

Moreover, TabNN [28] leverages GBDT to identify meaningful feature combinations, treating the features used in each decision tree as a group, which are then encoded using an MLP. Experimental results demonstrate that these hybrid approaches achieve strong performance on a variety of tabular data learning tasks. In addition, SuperTML [31] leverages two-dimensional CNNs by transforming tabular data into image-like representations, demonstrating the potential of CNNs for structured data classification. Furthermore, Boost-GNN [41] integrates GBDTs with GNNs in an end-to-end framework, where GBDT models heterogeneous tabular features and GNN captures graph structures, achieving superior performance on graph-based tasks with tabular attributes.

Following the phenomenal success of Transformer [2] and BERT [90] in NLP, researchers have turned their attention to applying these architectures to structured tabular data. Several representative approaches have emerged in this context. NPT [39] jointly models the entire dataset using self-attention to capture inter-sample dependencies and enable cross-sample reasoning. FT-Transformer [44] emphasizes the importance of standardized evaluation and proposes a simplified yet effective Transformer variant, highlighting the role of architecture and training strategy. LF-Transformer [59] incorporates ideas from matrix factorization by introducing a latent attention factor

matrix to enhance feature representation. Collectively, these methods demonstrate the growing potential of Transformer-based models as powerful alternatives to traditional approaches in tabular learning. Interpretable Additive Tabular Transformer Networks (IATTN) [55] combine self-attention with an additive modeling structure, allowing Transformer-based models to provide feature-wise contribution explanations for tabular predictions.

In addition, many works have proposed architectures specifically tailored to the unique characteristics of tabular data. Regularization Learning Networks (RLN) [29] address the challenge that traditional MLPs require extensive hyperparameter tuning to perform well on tabular tasks, by assigning a separate regularization coefficient to each weight, significantly improving performance. Regularized DNNs [40] explore 13 different regularization techniques and jointly optimize both the selection of regularizers and their auxiliary hyperparameters for each dataset. NODE [30] introduces differentiable decision trees by employing techniques such as the α -entmax transformation, enabling gradient-based optimization of split decisions and tree routing. It further stacks multiple layers and supports end-to-end training via backpropagation, allowing the model to capture complex data patterns.

NON [34] designs a triple-network architecture that explicitly models intra-field and inter-field relationships, effectively boosting classification performance on tabular data. DNF-Net [35] proposes the Disjunctive Normal Neural Form (DNNF) module, which integrates feature selection and spatial localization components, and can be trained end-to-end using standard gradient-based methods. ARM-Net dynamically models feature interactions in the exponential space and incorporates a multi-head gated attention mechanism, capturing complex dependencies in structured data while enhancing both predictive performance and interpretability.

IGTD [43] transforms tabular inputs into image-like representations and leverages CNNs for downstream processing, making it particularly suitable for domains such as biomedical data, where latent spatial relationships among features may exist. DNN2LR [42] addresses the trade-off between performance and interpretability by extracting nonlinear feature interactions from deep neural networks and converting them into explicit cross features for a logistic regression model, resulting in a lightweight yet interpretable model suitable for real-world applications. TabCBM [50] further introduces a concept-based neural network for tabular data, where human-understandable concepts are learned as an explicit intermediate layer, enabling transparent and concept-level explanations of model predictions. GANDALF [47] relies on a novel table processing unit called the Gated Feature Learning Unit (GFLU), which features a gating mechanism and built-in feature selection for learning feature representations. The superiority of GANDALF is demonstrated through experiments on multiple established public benchmarks. DeepGBM [33] introduces a hybrid deep learning framework that combines neural networks and gradient boosting decision trees to efficiently handle sparse categorical and dense numerical features, enabling accurate and adaptive online prediction for tabular data.

In recent studies, researchers have continued to explore

similar directions in deep learning for tabular data, while introducing novel architectural innovations. Trompt [51] is motivated by prompt learning in NLP and has been adapted for tabular data. It optimizes column embeddings through prompt embeddings, thereby generating specific feature importance for each sample. MambaTab [56] presents a new approach based on Structured State Space Models (SSMs), leveraging Mamba [91], a recent SSM variant, for end-to-end supervised learning on tabular data. Experimental results across a variety of benchmark datasets demonstrate that MambaTab achieves superior performance compared to state-of-the-art baselines, while requiring significantly fewer parameters. GCondNet [57] focuses on small, high-dimensional tabular datasets by introducing a tailored conditioning mechanism that improves representation learning and generalization under limited data regimes.

BiSHop [58] addresses two fundamental challenges in deep tabular learning: the lack of rotation invariance in data structures and the inherent feature sparsity of tabular data. Inspired by the connection between associative memory and attention mechanisms, BiSHop adopts a dual-component architecture comprising two interlinked directional learning modules that process data sequentially along both column-wise and row-wise dimensions. TabR [60] introduces a Feed-forward Neural Network (FNN) architecture that integrates a customized k-Nearest Neighbors (k-NN) retrieval module. This module identifies and retrieves training samples most similar to a given query instance, and incorporates their features and labels to enhance the prediction quality of the FNN. This hybrid design allows TabR to combine the strengths of neural networks and instance-based learning for improved performance on tabular tasks. DP-2Stage [64] adapts large language models for synthetic tabular data generation through a two-stage pipeline with differential privacy guarantees, enabling privacy-preserving data release for supervised learning.

b) Supervised-plus Methods

Supervised-plus methods integrate self-supervised pre-training with supervised fine-tuning, combining the representational strengths of unlabeled-data learning with the task-specific precision of supervised models. Motivated by the success of self-supervised learning in domains such as vision and language, recent research has applied this paradigm to tabular and behavioral data, where labeled samples are often scarce or noisy. By first learning generalizable representations from large-scale unlabeled data and then refining them through supervised adaptation, hybrid approaches enhance model generalization, robustness, and data efficiency in complex real-world scenarios.

Researchers have explored the potential of unsupervised feature representation from multiple perspectives, including graph structure modeling, feature reconstruction, embedding learning, and cross-modal conversion. One prominent line of work focuses on modeling the structural relationships among features. Given that tabular features often exhibit complex interdependencies, several studies have proposed representing tables as graphs to capture these interactions. For instance, CARTE [62] introduces an innovative approach by modeling each row of a table as a star-like graph, where individual cell

values are treated as nodes and column headers serve as edge labels. By integrating the strengths of Transformers and Graph Attention Networks (GAT), CARTE leverages self-attention mechanisms to capture contextual dependencies within tabular data. Its self-supervised learning strategy is designed within a contrastive learning framework, enabling the model to extract informative representations from large-scale background data, which facilitates effective transfer learning for downstream tasks.

In contrast, HYTREL [52] models tabular data using a hypergraph structure to capture its inherent relational characteristics. Specifically, each cell is represented as a node, while rows, columns, and the entire table are treated as hyperedges. A structure-aware Transformer is employed to encode this hypergraph representation, and the model is trained through a self-supervised objective to enhance its understanding of table structure.

Another widely adopted strategy for self-supervised learning on tabular data is based on masking and reconstruction, inspired by masked language modeling in NLP. VIME [36] facilitates the learning of informative feature representations from unlabeled tabular data by jointly estimating feature vectors and corresponding mask vectors. Tabbie [92] generates corrupted cell candidates by swapping two cells within the same table and trains the model to predict whether each cell has been swapped. SCARF [46], on the other hand, randomly selects a subset of features and replaces them with values sampled from their empirical marginal distributions to create corrupted inputs.

SAINT [45] and TabNet [32], adopt Transformer-based architectures. SAINT employs contrastive learning to encourage the model to capture similarities and differences across various “views” of the same data point. It also introduces a denoising task to help the model recover original data from corrupted inputs, thereby enhancing its ability to leverage unlabeled data. TabNet [32] randomly masks a portion of the input features and attempts to reconstruct the masked features from the unmasked ones. This task encourages the model to learn the underlying structure of the data and the relationships among features. Building upon the masking-based approach, TabTransformer [37] introduces a replacement token detection task, in which a subset of feature values is randomly replaced, and a binary classifier is trained to predict whether a given feature value has been replaced.

Recontab [53] and XTab [54] also utilize reconstruction loss as a self-supervised learning signal. Notably, ReConTab [53] further incorporates label information to compute a semi-supervised classification loss as well as a contrastive loss, enabling the model to better capture discriminative features in the data. SwitchTab [63] decomposes the feature representation of each data sample into mutual information and salient information, and reconstructs the data by swapping the mutual information components between different samples. This process not only encourages the model to learn more structured embeddings, but also enables it to explicitly distinguish between shared characteristics across samples and sample-specific distinctive features.

In addition, PTab [49] transforms tabular data into textual

format and trains the model using BERT-style pre-training objectives, achieving competitive performance across various tasks. TabDDPM [48] explores the applicability of diffusion models to general-purpose tabular problems and introduces TabDDPM—a diffusion-based framework capable of handling arbitrary tabular datasets and feature types. TP-BERTa [61] proposes a novel relative magnitude tokenization approach that converts continuous numerical feature values into finely discretized high-dimensional tokens. It further incorporates an intra-feature attention mechanism to integrate feature values with their corresponding feature names. This unique self-supervised objective enhances the model’s adaptability to tabular data by explicitly modeling feature semantics and structure. More broadly, TARTE [65] advocate large-scale pre-training on diverse tabular corpora to acquire transferable knowledge and generalizable representations for downstream tabular tasks. In addition, TDCoLER [66] investigates tabular data distillation, focusing on transforming complex tabular structures into compact and distillable representations suitable for efficient downstream learning.

In summary, self-supervised learning for tabular data has witnessed a surge of diverse and creative approaches. Whether through graph-based modeling, feature masking and reconstruction, embedding optimization, or modality transformation, these methods aim to extract meaningful supervisory signals from unlabeled data. Moving forward, future research may further integrate these strategies to develop more robust and transferable pre-training frameworks for a wide range of structured data analysis tasks.

3) LLM-driven Methods

Recently, LLMs have attracted significant attention for their impressive performance across a wide range of tasks, demonstrating generalization capabilities far beyond traditional NLP applications [93], [94]. With their powerful language understanding and generation abilities, LLMs are increasingly regarded as a promising pathway toward Artificial General Intelligence (AGI) [95], [96]. Naturally, efforts have begun to investigate the potential of LLMs in tabular data processing, leveraging their language interfaces to unify the modeling of structured information. These approaches have shown promising results in tasks such as prediction, question answering, reasoning, and data augmentation [69], [72], [97]. Recent work has also explored fairness-aware in-context learning for tabular data, where tabular foundation models are adapted to mitigate bias during inference [98]. In this section, we provide a brief overview of LLM-driven techniques for tabular data modeling.

A natural approach to leveraging LLMs for tabular data is to first convert the structured data into a textual format that the model can understand. This can be achieved by representing the table using programming-language-friendly data structures—such as Python’s Pandas DataFrame loaders, line-separated JSON files, nested lists for data matrices, or HTML table representations. Alternatively, tables can be serialized into delimiter-separated values, where delimiters such as commas or tabs are commonly used. Some studies further transform tables into human-readable sentences by applying templates that incorporate column headers and corresponding cell values, enabling LLMs to process the data in a more

context-aware manner [99]–[101]. Several studies further analyze the robustness of LLMs for tabular question answering by examining internal attention patterns, providing insights into how models align questions with tabular evidence [102].

Another key direction for optimizing LLM-based tabular data modeling lies in how to effectively structure and present tabular information to maximize the model’s comprehension. One common strategy is context compression, which addresses the input length limitations of LLMs. For instance, researchers propose truncation techniques based on maximum sequence length [67], [68]. Gpt4table [103] introduce predefined constraints to ensure that the input remains within the acceptable range for LLMs.

Another effective approach involves selective retrieval, where only the most relevant tables, rows, columns, or cells are extracted and fed into the model. In this context, Retrieval-Augmented Generation (RAG) has also emerged as a viable solution, combining retrieval mechanisms with generative models to enhance the model’s ability to leverage structured information [70], [104].

Beyond raw table content, several studies have explored enriching prompts with auxiliary tabular information. TAP4LLM [105] demonstrate that incorporating metadata such as table dimensions, measurement units, and semantic field types can significantly improve model performance across six benchmark datasets. Furthermore, statistical features—particularly in datasets with a high proportion of quantitative cells, such as fever records [106]—have been shown to enhance task accuracy. Additional contextual signals, such as document references and terminology explanations, help inject semantic meaning into tabular inputs. However, not all augmentations yield positive results: while statistical and semantic cues are beneficial, features like table size offer limited gains, and header hierarchies may introduce unnecessary complexity, even degrading performance.

In addition to structural input optimization, two widely adopted strategies for enhancing LLM performance on tabular data tasks are prompt engineering and fine-tuning. Prompt engineering techniques aim to guide the model’s reasoning and output by crafting effective input prompts. A common baseline is to concatenate a task description with a serialized table, allowing the model to interpret and respond accordingly. Research has shown that well-defined and explicitly stated task instructions significantly improve performance [107]. In addition, TST-LLM [71] leverages LLMs to generate task-aware pretext objectives, enabling more effective representation learning without manual labels. Furthermore, incorporating external context, such as relevant questions or statements alongside the table, can enhance the model’s understanding and accuracy [103].

Advanced prompting strategies such as Chain-of-Thought (CoT) prompting [108] encourage step-by-step reasoning, enabling the model to decompose complex tasks into intermediate steps. Variants like program-aided CoT [109] and Self-Consistency (SC) [110] further boost performance by sampling multiple reasoning paths and aggregating the most consistent outcomes. These methods are particularly effective in numerical reasoning, table-based QA, and fact verification

tasks.

In contrast to prompt-based approaches, fine-tuning adapts LLMs to specific tabular tasks by updating model parameters with labeled data, often resulting in improved performance and tighter integration with external tools such as SQL and Python. Recent studies have actively explored how LLMs can interact with these tools to enhance reasoning over structured data. For instance, an iterative framework is proposed in [111], where execution errors from SQL queries are fed back into the LLM to enable successive refinements that significantly improve query generation accuracy. In the realm of analytical platforms, the work of [112] introduces a no-code solution that leverages LLMs to generate data summaries, formulate relevant analytical questions, and produce structured queries.

Furthermore, a survey by [113] reviews natural language interfaces for tabular data, highlighting recent advances in Text-to-SQL and Text-to-Vis techniques. Together, these fine-tuning and tool-integrated approaches showcase the growing potential of LLMs in enabling more intuitive and effective interaction with tabular data. A recent survey [114] also reviews how LLMs are adapted for tabular prediction, data generation, and reasoning, highlighting emerging trends and open challenges.

B. Event Sequences

Rooted in the foundational principles of sequential decision-making [154], human behavior can be naturally modeled as a sequential process. Unlike traditional feature-based analysis, sequence-based structural modeling captures temporal dynamics, enabling more accurate behavior representation by leveraging historical patterns to inform present decision-making. Typically, a behavior sequence consists of a continuous series of ordered interactions, each associated with a timestamp and relevant contextual features. This formulation allows for the application of powerful sequential representation learning models to behavioral data analysis. Broadly, sequential modeling approaches can be categorized into **(1) Symbolic Modeling Methods**, and **(2) Deep Learning-based Methods**.

a) Symbolic Modeling Methods

Symbolic modeling methods include **(a) Sequential Pattern Mining** and **(b) Markov Chain Models**.

a) Sequential Pattern Mining

Sequential Pattern Mining follows a paradigm that first extracts frequent patterns from sequences and then utilizes them to predict the next action. Several studies have adopted this simple yet effective approach for behavioral data analysis [116], [155]. Although this solution is feasible, it not only incurs high computational costs but also predominantly relies on frequent patterns for prediction, thereby overlooking the informative value of infrequent patterns [156].

b) Markov Chain Models

Markov Chain have been widely employed to model behavior sequence transitions. Existing approaches can be categorized into two main paradigms: (1) direct computation of transition probabilities from historical sequences [157], [158], and (2) probability estimation through Euclidean distance between action embeddings [117], [159]. However, due to the

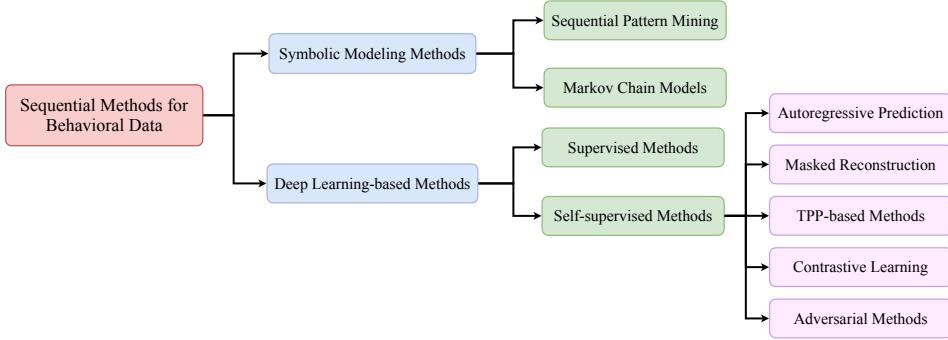


Fig. 4: The dissection of the event sequence representation learning methods.

fundamental Markov assumption, where the next state depends solely on the current state, these methods exhibit two critical limitations: they struggle to capture long-term dependencies and fail to model joint probabilities where multiple actions collectively influence the next action [117].

2) Deep Learning-based Methods

In recent years, deep learning has become the dominant approach for sequential modeling in behavioral data analysis. These methods can be broadly categorized into **(a) Supervised** and **(b) Self-supervised** learning paradigms, each with distinct frameworks that we will examine separately.

a) Supervised Methods

Supervised learning methods for behavior sequence representation utilize annotated datasets to optimize task-specific objectives such as classification or regression through direct minimization of supervised loss functions (e.g., cross-entropy or mean squared error). Although dependent on manual annotation, these methods achieve strong empirical performance when sufficient labeled data are available, with architectural flexibility across domains. The pipeline of supervised methods is shown in Fig. 5.

To illustrate the progression of this field, we structure the review chronologically, tracing the evolution of sequential supervised models from early RNN-based encoders to the latest LLM-driven architectures for behavior modeling.

The inherent sequential nature of event data found a natural match in RNNs. The core idea of an RNN is to maintain a hidden state, h_t , which acts as a compressed memory of the entire history seen up to event e_t . This state is updated recursively with each new event:

$$h_t = f(h_{t-1}, x_t)$$

where x_t is the embedding of the current event e_t , and $f(\cdot)$ denotes the state update function that integrates the previous hidden state and the current input. To predict the next event, the final hidden state is passed through a prediction layer (e.g., a softmax layer over the item vocabulary). Beyond RNN-based approaches, Memory-Augmented Neural Networks (MANN) [126] enhanced collaborative filtering through external memory modules, while neuromorphic event modeling techniques like Event2Vec [121] adapted Word2Vec's [160] embedding paradigm for supervised classification of benchmark datasets such as the ASL-DVS gesture

recognition dataset [161]. SeqLink [146] addresses sparsely and irregularly observed behavior sequences by modeling continuous-time latent dynamics with Neural Ordinary Differential Equations (ODEs), aggregating multiple linked latent trajectories to obtain robust representations under long observation gaps.

RNN-based models are often criticized for their computational inefficiency, as their autoregressive nature requires sequential processing and prevents parallelization. In contrast, CNNs overcome this limitation by processing entire sequences in parallel. Beyond efficiency gains, CNN-based models excel at capturing local temporal patterns in user behavior. Caser [124] adopted CNN as encoder to capture local temporal dynamics in user-item interactions. Another notable approach is NextItNet [131], which employs stacked holed convolutional layers to efficiently expand receptive fields for sequential modeling.

More recently, Transformers have gained widespread adoption for sequential behavior modeling. SASRec was the first to explore Transformer architectures for sequential recommendation tasks, inspiring numerous follow-up improvements. Many subsequent Transformer-based methods are developed. TiSASRec [135] incorporated personalized time interval processing into attention mechanism to better model temporal patterns in recommendation sequences. AttRec [125] integrated self-attention with metric learning for next-item prediction and AS-Rep [138] proposed reverse-trained Transformers to augment short behavior sequences, further advancing recommendation accuracy. PhASER [150] further improves supervised classification by explicitly modeling non-stationarity through phase-aware representations, decomposing sequences into magnitude and phase components and applying phase-driven augmentation to enhance robustness under distribution shifts.

Recent advancements have also leveraged LLMs, given their remarkable capabilities in sequence understanding. For example, AlmostRec [162] enhanced predictive modeling in sequential recommendation by aligning multi-modal user behavior (e.g., IDs, text, images) using a large multi-modal model and an ID prediction objective for controllable and accurate recommendations.

The effectiveness of each architecture hinges on its alignment with the domain's inductive biases: RNNs and their gated variants (e.g., Long Short-Term Memory, LSTM [4]

TABLE II: Event sequence representation learning methods.

Method	Training Paradigm	Model Architecture	Year
Symbolic Modeling Methods			
FPMC [115]	Self-supervised learning	Markov Chain	2010
PSPM [116]	Self-supervised learning	Sequential pattern mining	2012
PRME [117]	Self-supervised learning	Markov Chain	2015
Deep Learning-based Methods			
GRU4Rec [118]	Supervised learning	GRU	2015
RMTPP [119]	Self-supervised learning	RNN	2016
NHP [120]	Self-supervised learning	LSTM	2017
Event2Vec [121]	Supervised learning	Transformer	2017
IRGAN [122]	Semi-supervised learning	GAN	2017
HRNN [123]	Supervised learning	RNN	2017
Caser [124]	Supervised learning	CNN	2018
AttRec [125]	Supervised learning	Transformer	2018
MANN [126]	Supervised learning	Memory network	2018
RecGAN [127]	Semi-supervised learning	GAN+RNN	2018
RPPN [128]	Self-supervised learning	RNN	2019
BERT4Rec [16]	Self-supervised learning	BERT	2019
DTCDR [129]	Supervised learning	MLP	2019
FDSA [130]	Supervised learning	Transformer	2019
NextItNet [131]	Supervised learning	CNN	2019
SAHP [132]	Self-supervised learning	Transformer	2020
THP [133]	Self-supervised learning	Transformer	2020
BEHRT [134]	Self-supervised learning	BERT	2020
TiSASRec [135]	Supervised learning	Transformer	2020
RAPT [136]	Self-supervised learning	Transformer	2021
CoSeRec [137]	Self-supervised learning	GAN+CL	2021
ASReP [138]	Supervised learning	Transformer	2021
UniSRec [139]	Self-supervised learning	BERT+Transformer	2022
RecGURU [140]	Self-supervised learning	Transformer	2022
promptTPP [141]	Continual learning	Transformer+ Prompts	2023
Meta TPP [142]	Meta learning	Transformer	2023
BERT4ETH [143]	Self-supervised learning	BERT	2023
PrimeNet [144]	Self-supervised learning	Transformer	2023
ECGAN-Rec [145]	Semi-supervised learning	GAN	2023
SeqLink [146]	Supervised learning	Neural ODE	2024
Chronos [147]	Self-supervised learning	Transformer	2024
Player2Vec [148]	Self-supervised learning	BERT	2024
TOTEM [149]	Self-supervised learning	VQ-VAE	2024
PhASER [150]	Supervised learning	Multiple	2025
Residual TPP [151]	Self-supervised learning	Hawkes + Neural TPP	2025
IOCLRec [152]	Self-supervised learning	Transformer+CL	2025
HORAE [153]	Self-supervised learning	Transformer	2025

and Gated Recurrent Unit, GRU [163]) are well-suited for modeling sequential dependencies through recursive state updates, enabling effective modeling of short- to medium-range user behavior; memory networks facilitate iterative refinement of user state via external storage; CNNs excel at extracting shift-invariant local patterns in short-term interactions; Transformers capture global contextual dependencies through fully parallelizable computation; and LLMs generalize across symbolic and multi-modal behavior via large-scale pre-training. Beyond architectural choices, recent work also studies training principles for supervised sequence representations. In particular, Mohapatra et al. [164] show that properly regularizing sequence-specific embeddings are crucial to prevent co-adaptation with shared layers, thereby improving representation quality and transferability.

b) Self-supervised Methods

Self-supervised learning for behavior sequence modeling typically learns sequence representations through five key paradigms: Autoregressive Prediction, Masked Reconstruction,

TPP-based Methods, Contrastive Learning, and Adversarial Methods. In the following sections, we categorize and summarize existing work according to these paradigms.

i) Autoregressive Prediction: Autoregressive Prediction is the framework that learns sequence representations by predicting the future values based on observed historical data, thus encouraging the model to learn the underlying temporal structure and behavior patterns. Such models treat the event sequence as a unidirectional process, where the future is generated step-by-step conditioned on the past. This formulation aligns naturally with many real-world tasks such as next-click prediction [131], [157], stock price forecasting [143], and sequential decision-making [148]. For instance, inspired by the parallels between generative modeling and financial tasks such as churn prediction, credit default, and expenditure forecasting, NPPR [165] employs next event prediction adapted from language modeling to handle multivariate transaction events. Chronos [147] further formulates time series modeling as a language-modeling-like problem by discretizing continuous

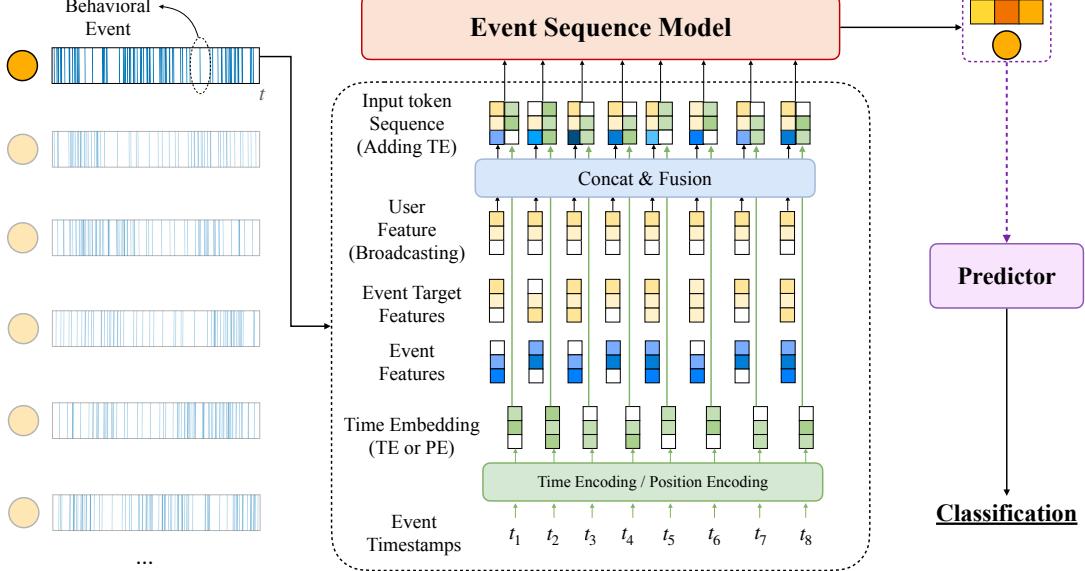


Fig. 5: The pipeline of supervised event sequence representation learning methods.

values into tokens and training transformer models with a next-token prediction objective. By autoregressively modeling token sequences via cross-entropy loss, Chronos learns general-purpose time series representations that transfer across domains without task-specific architectures.

ii) Masked Reconstruction: Masked Reconstruction learns sequence representations by reconstructing masked portions of historical records (e.g., actions or features). During training, the model processes partially masked input sequences and learns to predict the original data using the remaining unmasked tokens as context. This reconstruction objective forces the model to capture both temporal patterns and structural dependencies in the data, which enhances its effectiveness for downstream classification tasks.

This paradigm closely resembles the successful BERT model in NLP. A substantial body of research directly adapts BERT’s encoder architecture and Masked Language Modeling (MLM) objective for behavior sequences, repurposing its bidirectional attention mechanism to capture complex temporal dependencies in actions and features. For example, BERT4Rec [16] leverages a bidirectional Transformer encoder for masked item prediction, enabling robust information aggregation across entire historical sequences. Similarly, BEHRT [134] builds upon BERT’s framework, adopting a two-stage pre-training and fine-tuning approach while integrating multi-modal data from five heterogeneous data sources to enhance depression prediction accuracy. Furthermore, BERT4ETH [143] improves Ethereum transaction pattern recognition using a Transformer architecture with masked address modeling and domain-specific strategies to address data repetitiveness, skewness, and heterogeneity. Meanwhile, Player2Vec [148] adapts long-context Transformer architectures to model player behavior from sessionized interaction logs, producing enriched behavior representations.

Beyond BERT-like methods, some works adopt

reconstruction-based objectives without explicit masking to learn general-purpose behavioral representations. For instance, TOTEM [149] learns discrete time series representations by training a Vector-Quantized Autoencoding Reconstruction (VQ-VAE) tokenizer to reconstruct unmasked historical sequences in a self-supervised manner, producing a fixed codebook of temporal tokens shared across domains and tasks.

Conversely, an alternative research direction argues for architectural modifications to address inherent limitations of vanilla Transformers in modeling behavioral data. Innovations such as RAPT [136] augment self-attention with explicit temporal encoding to handle irregular inter-event intervals, while SAGE [166] introduces noisy reconstruction, injecting deletions, substitutions, and insertions into the masked sequence, leading to notable performance improvements.

This methodological divergence highlights distinct design philosophies: while BERT-like models provide powerful general-purpose sequence representations, specialized Transformer variants offer tailored solutions for temporal dynamics and data quality challenges in behavior analytics.

iii) TPP-based Methods: Temporal Point Processes (TPPs) are a class of probabilistic models designed to characterize the timing of events over continuous time by explicitly preserving the original timestamp of each occurrence. Unlike traditional sequence models which predict the next token in a discrete sequence, TPPs model the conditional intensity function to estimate the probability of an event occurring at any time t , given its event history. This unique formulation makes TPPs particularly effective for handling irregular, asynchronous event sequences as commonly found in behavioral data streams such as click logs, social media actions, or medical visit records.

While traditionally formulated in the context of generative modeling, TPPs can also be viewed as a specialized form

of self-supervised learning. They optimize a likelihood-based objective by predicting the timing of future events solely based on past observations, without requiring any external labels.

Early deep learning approaches for TPPs naturally leveraged RNNs [1] to encode the event history. For instance, RMTPP [119] uses an RNN to learn a representation of the event history, embedding the sequence of past event times and their associated markers into a hidden state vector which then parameterizes the intensity function. Similarly, NHP [120] introduces a novel continuous-time LSTM architecture where the hidden state evolves continuously between events, allowing it to capture more complex, non-linear patterns of excitation and inhibition than traditional Hawkes models. RPPN [128] jointly model asynchronous event sequences and synchronous time series by parameterizing the endogenous and exogenous intensity functions with separate RNNs.

However, RNN-based TPPs struggle to effectively capture long-term dependencies, subsequent research drew inspiration from the success of attention mechanisms in NLP to address this issue. For example, SAHP [132] and THP [133] employ self-attention to summarize the influence of history events and compute the probability of the next event, which not only models long-range dependencies more effectively but also allows for parallel computation over the event sequence, breaking the sequential processing bottleneck of RNNs [1]. There has also been research in creative ways of training temporal point processes, such as in continual learning framework [141] whose basis is a continuous-time retrieval prompt pool for modeling streaming event sequences and meta learning framework [142] to quickly adapt to new, heterogeneous sequences from few examples.

While attention-based TPPs improve expressiveness, they often incur high computational overhead and face difficulty in training on long sequences. To address this, the residual TPP [151] proposes a hybrid framework that first employs a lightweight Hawkes process to capture dominant temporal patterns, and then delegates only the residual, irregular dynamics to a neural TPP, significantly reducing training complexity while preserving accuracy.

iv) *Contrastive Learning*: Contrastive Learning approaches aim at maximizing similarity between semantically related sequences while minimizing similarity between unrelated ones, thus improving generalization across tasks. Contrastive learning has been widely used in many fields including computer vision and sequential modeling. To make it more powerful in modeling behavioral data, many works have proposed modifications and focuses mainly on three aspects: *sample selection, loss function, and data augmentation*.

- *Sample Selection*: A critical challenge in modeling behavioral data lies in the irregular timing of events, which requires specialized sampling strategies to preserve temporal dynamics. For example, PrimeNet [144] tackles this issue via time-sensitive stratified sampling and fixed-time masking, explicitly capturing irregular patterns in multivariate time series during augmentation. Beyond temporal irregularity, another line of research focuses on mitigating semantic bias in sampled sequences. In recommendation systems, IOCLRec [152] reduces false

negatives in contrastive pairs by aligning user behavior sequences with their latent intents, thereby improving both robustness and recommendation accuracy.

- *Loss Function*: Another line of research focuses on enhancing contrastive learning through novel loss function designs to improve adaptability to specific scenarios. For instance, HORAE [153] explicitly incorporates temporal dynamics into its contrastive learning framework, enabling multi-interest pre-training by aligning user behaviors with their evolving time-aware representations. Similarly, S3-Rec [167] adapts contrastive learning to recommendation systems by introducing four auxiliary self-supervised objectives, which jointly capture multi-view semantic correlations (e.g., item attributes, sequential dependencies) to improve representation learning. These approaches demonstrate that tailored loss functions can effectively disentangle complex behavioral semantics.
- *Data Augmentation*: Contrastive learning can also be enhanced through innovative data augmentation strategies. CL4SRec [168], a pioneering work in sequential recommendation, proposes three atomic augmentation operations (crop, mask, reorder) to generate self-supervised signals from interaction sequences. Subsequent studies further develop this paradigm: CoSeRec [137] introduces substitute and insert operations to create more informative augmented views, while DuoRec [169] proposes model-level augmentation via Dropout and a supervised positive sampling strategy to address representation degeneration in sequential modeling.

v) *Adversarial Methods*: Adversarial Methods employ a discriminator to distinguish between generated fake samples and real samples, while the generator improves the quality of its outputs to deceive the discriminator. This framework has been effectively adapted for behavioral data processing. For example, RecGAN [127] integrates RNNs and GANs through adversarial training to model temporal preferences for recommendations. Similarly, IRGAN [122] unifies generative and discriminative retrieval paradigms via a minimax adversarial game, where the generative model produces challenging samples and the discriminative model refines its ranking, achieving significant improvements in search, recommendation, and QA tasks. Further advancing this approach, ECGAN-Rec [145] combines contrastive learning with adversarial training to enhance sequential recommendation by balancing data sparsity and noise, while CoSeRec [137] also leverages adversarial training alongside contrastive learning to boost performance of downstream tasks.

C. Dynamic Graphs

Human behaviors frequently manifest as evolving interactions among individuals or entities over time, naturally lending themselves to representation as dynamic graphs. Specifically, a dynamic graph captures temporal variations in connections between entities (e.g., people, items), thereby providing a structured framework to analyze complex, time-dependent relational patterns.

Formally, behavioral data can be modeled as a dynamic graph $G = (V, E, T)$, where V is a set of nodes representing

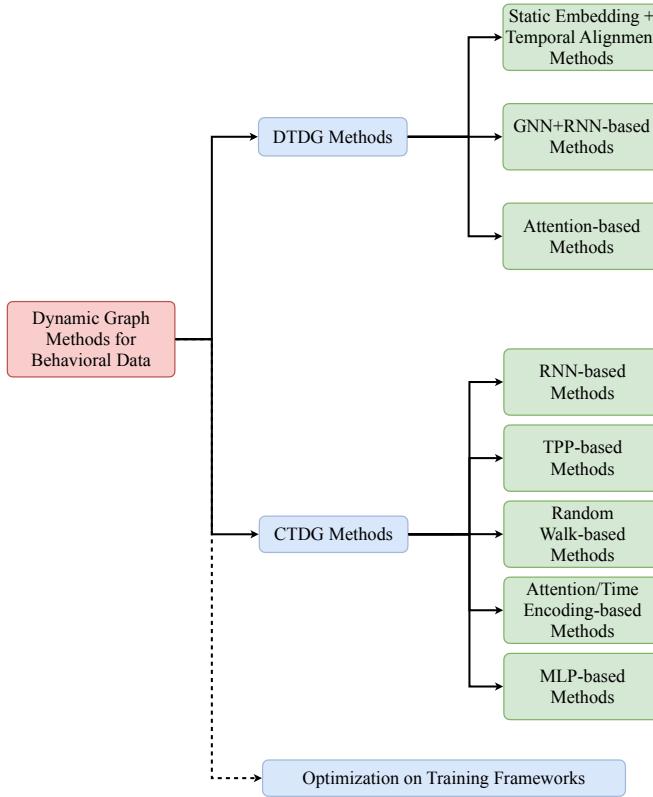


Fig. 6: The dissection of dynamic graph representation learning methods.

individuals or entities, E denotes the set of interactions or relationships (edges) among nodes, and T explicitly encodes temporal information. Based on how the temporal dimension T is defined, dynamic graphs are typically classified into two distinct types:

- **Discrete-Time Dynamic Graphs:** In Discrete-Time Dynamic Graphs (DTDGs), temporal dynamics are captured via a series of graph snapshots at discrete time intervals [170]: $\mathcal{G} = \{G_1, G_2, \dots, G_T\}$, $G_t = (V_t, E_t)$, $t = 1, 2, \dots, T$, where each snapshot G_t describes the structure of interactions within a fixed time interval, representing aggregated relationships.
- **Continuous-Time Dynamic Graphs:** Continuous-Time Dynamic Graphs (CTDGs) record each interaction as individual events at precise timestamps, thus offering finer granularity [5]: $G = (V, \mathcal{E})$, $\mathcal{E} = \{(u, v, t) \mid u, v \in V, t \in \mathbb{R}^+\}$, where each edge (u, v, t) indicates a direct interaction between nodes u and v occurring at a continuous timestamp t .

Dynamic graphs typically facilitate two primary downstream analytical tasks: Dynamic Link Prediction and Node Classification. While a variety of other downstream tasks have also been explored, such as anomalous link detection [171], they are comparatively less studied and often application-specific; therefore, we focus primarily on the two dominant tasks in this paper. The primary output of representation learning methods for both DTDG and CTDG is typically dynamic node embeddings, denoted as $\mathbf{z}_v(t)$ for a node v at

time t . These embeddings encode both structural and temporal information accumulated up to time t , and serve as the basis for various downstream tasks.

- **Dynamic Link Prediction:** Predicting future interactions between nodes based on historical graph structures. Dynamic Link prediction is typically formulated as a self-supervised task, where the model is trained to predict future interactions based on historical data. Specifically, the future edges serve as implicit supervision signals, allowing the model to learn temporal patterns without requiring manual labels. Given a node pair (u, v) and a query time t , the objective is to estimate the likelihood of a future edge:

$$\hat{y}_{uv}(t) = \sigma(f_{\text{link}}(\mathbf{z}_u(t), \mathbf{z}_v(t))), \quad (1)$$

where f_{link} denotes a scoring function (e.g., dot product), and $\sigma(\cdot)$ is a sigmoid function. For behavioral data, link prediction corresponds to real-world tasks such as forecasting social connections, recommending potential friends, or predicting future collaborative partnerships.

- **Node Classification:** Identifying node attributes or categories based on their temporal relational context. Node classification is usually treated as a supervised task, where a set of nodes is associated with ground-truth labels. In practice, the node embeddings $\mathbf{z}_v(t)$ learned from the link prediction task are used, and an additional classification head is trained on top of these embeddings using the available label data:

$$\hat{y}_v = \text{softmax}(f_{\text{cls}}(\mathbf{z}_v(t))), \quad (2)$$

where f_{cls} is typically a Multi-Layer Perceptron (MLP) or linear classifier. Practical tasks for behavioral data include detecting influential users in social networks, inferring user demographics, or identifying community roles based on interaction patterns.

In the following subsections, we systematically review and analyze representation learning methods designed for discrete-time and continuous-time dynamic graphs, respectively, followed by a discussion of the **Optimization on Training Frameworks** that are applicable across dynamic graph models. We highlight the core methodologies, recent advancements, and existing limitations of each category. It is important to note that this section focuses on introducing how the backbone models modeling behavioral data, while their associated downstream applications will be discussed in detail in Sec. IV.

1) Discrete-Time Dynamic Graphs

DTDGs capture evolving networks by recording the graph at successive fixed time intervals as a sequence of static snapshots [213]. This snapshot-based view makes it easy to leverage traditional static graph methods (by treating each snapshot as independent static graph), while still preserving the coarse temporal ordering across snapshots.

Most representation learning methods for DTDG follow a two-step modeling paradigm that first encodes the structural information within each snapshot and then captures temporal dependencies across snapshots. Given that the core distinction

TABLE III: Discrete-time dynamic graph representation learning methods.

Method	Structural Encoding	Temporal Encoding	Year
Static Embedding + Temporal Alignment Methods			
Chakrabarti et al. [172], Chi et al. [173], Kim & Han [174], Gupta et al. [175], Yao et al. [176], Zhou et al. [177]	Matrix factorization	Smoothness regularization or alignment	2006- 2018
Hisano [178]	Matrix factorization	Time window aggregation	2018
Sharan & Neville [179]	Matrix factorization	Time-weighted adjacency matrices	2008
Ibrahim et al. [180]	Matrix factorization	Exponential decay	2015
Ahmed et al. [181]	Low-rank adjacency	Temporal sampling strategies	2016
Singer et al. [182]	Random walk	Init. from previous step + fine-tuning	2019
DynGEM [183]	Deep autoencoder	Regularization across snapshots	2018
DynamicTriad [177]	Triadic closure	Temporal smoothness	2018
GNN+RNN-based Methods			
GCRN [184]	GCN	LSTM	2018
Narayan & Roe [185]	GraphSAGE	LSTM	2018
TGCN [186]	GCN	GRU	2019
TNA [187]	GCN	GRU	2019
VGRNN [188]	VGAE	LSTM	2019
LRGCN [189]	R-GCN	LSTM	2019
E-LSTM-D [190]	Autoencoder	LSTM	2019
EvolveGCN [191]	GCN	GRU	2020
dyngraph2vec [192]	graph2vec	LSTM/GRU	2020
TeMP [193]	GCN	GRU or Attention	2020
WD-GCN/CD-GCN [194]	GCN	Modified LSTM	2020
HDGNN [195]	Heterogeneous random walk	Bi-RNN	2020
HTGN [196]	Hyperbolic attention-based GCN	Hyperbolic GRU	2021
GC-LSTM [197]	GCN	LSTM	2022
ROLAND [198]	GCN	Adaptive RNN	2022
RPC [199]	GNN	GRU	2023
SEIGN [200]	GCN-like message passing	GRU parameter adjustments	2023
RETIA [201]	GCN	GRU + LSTM	2023
MegaCRN [202]	Meta-graph learner	Custom GRU	2023
DEFT [203]	GNN	RNN-based parameter evolution + Wavelet	2023
STGNPP [204]	GCN + Transformer	Continuous GRU	2023
WinGNN [205]	GNN	Sliding window	2023
SpikeNet [206]	GNN	SSN	2023
TTGCN [207]	Truss-based GCN	GRU	2024
Attention-based Methods			
DySAT [208]	Graph attention	Graph attention	2020
TEDIC [209]	Graph diffusion	Temporal convolutional network	2021
DyHATR [210]	Hierarchical attention	Temporal attentive RNN	2021
DREAM [211]	Attention	Attention + Reinforcement learning	2023
STGNP [212]	Dilated causal convolution	Cross-set graph convolution	2023
DTFormer [170]	Transformer	Transformer	2024

between DTDGs and static graphs lies in their temporal dynamics, we classify existing methods primarily based on how they model temporal dependencies, while also taking their structural modeling strategies into account. Therefore, we propose to classify existing DTDG representation learning methods into three-way taxonomy. **(a) Static Embedding + Temporal Alignment Methods:** These methods adapt static graph algorithms to each snapshot and then fuse or smooth the results over time. Often, an explicit smoothness or alignment constraint is added so that the embedding of each node changes gradually from one snapshot to the next. This “shallow” approach thus incorporates temporal smoothing but does not learn an explicit temporal model. **(b) GNN+RNN-based Methods:** These hybrid models use a GNN to encode the structure of each snapshot and an RNN to propagate information through time. Concretely, a GNN produces a

node-embedding vector from each graph snapshot, yielding a time-ordered sequence of embeddings for each node. An RNN is then run on that sequence to capture temporal dependencies, so that each node’s final representation reflects both its local topology (via the GNN) and its evolution history (via the RNN). This recurrent approach effectively learns dynamic updates as the graph changes.

(c) Attention-based Methods: These models leverage self-attention mechanisms to capture both temporal and structural dependencies without relying on recurrent updates. By applying attention mechanism these models enable flexible modeling of long-range temporal interactions, addressing common limitations of RNNs such as vanishing gradients and sequential bottlenecks. Almost all existing methods follow a two-step pipeline: first, they employ GNNs, Transformers, or other encoders to capture structural information; then, they apply a sequence model to capture

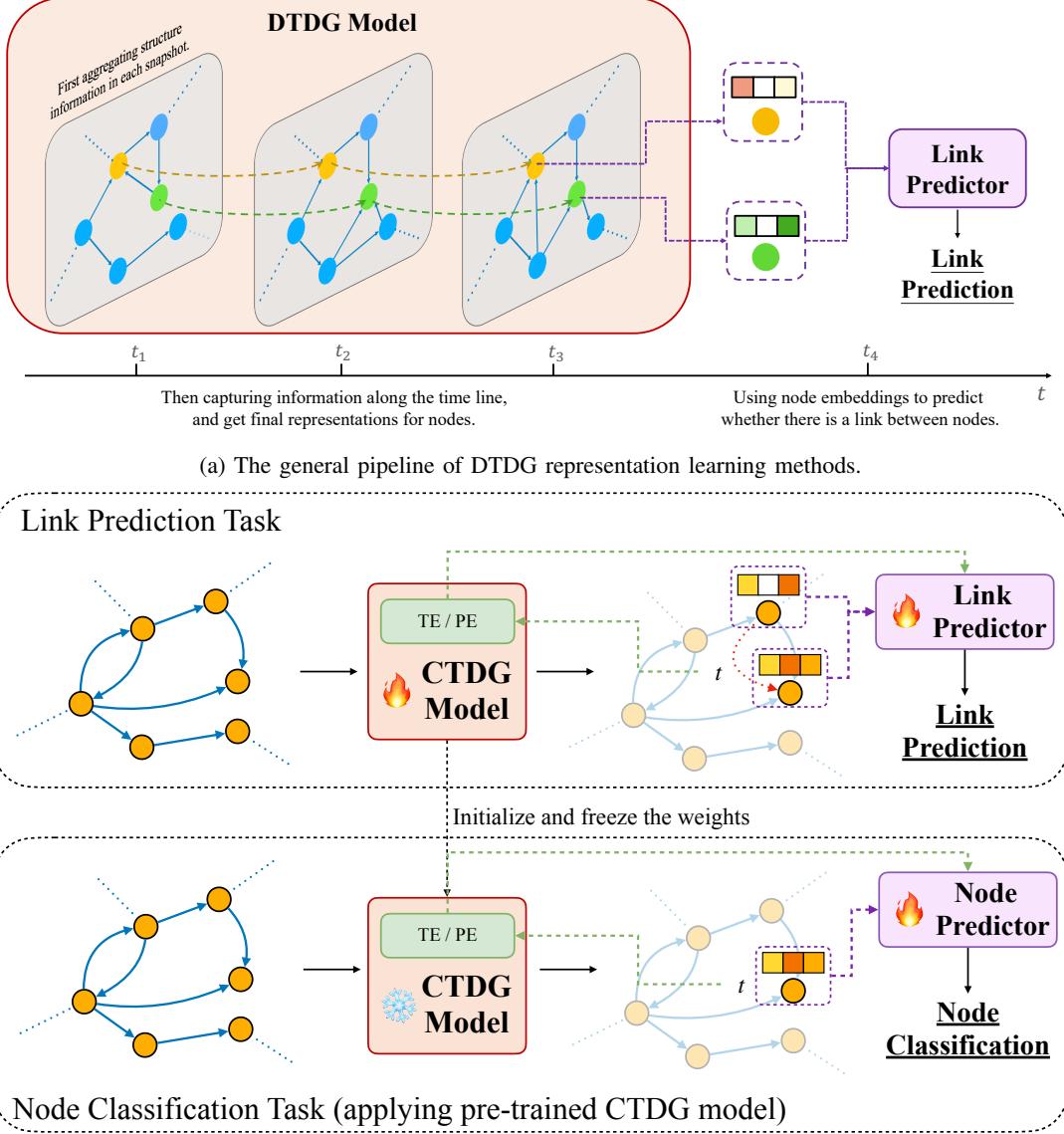


Fig. 7: The pipelines of DTDG and CTDG methods.

temporal dynamics along the timeline and produce the final embedding. This pipeline is illustrated in Fig. 7a.

Note that this taxonomy is defined in a broad sense. For example, methods categorized under the GNN+RNN-based Architecture may not strictly adopt a GNN followed by a conventional RNN module; rather, we group together architectures that share a similar design principle. A comparison among DTDG representation learning methods is shown in Tab III.

a) Static Embedding + Temporal Alignment Methods

Early approaches to DTDG representation learning built upon techniques for static graphs, extending them to handle temporal evolution, as DTDGs are formed with a series of consecutive snapshots [170]. A common strategy is to apply matrix factorization or latent space models to each snapshot of the graph and impose temporal smoothness constraints

between snapshots [172]–[177], [214]. This effectively regularized the embedding trajectory of each node, assuming that node characteristics vary smoothly over time. Some methods aggregated multiple snapshots into a single decomposition: e.g., summing or averaging adjacency matrices over a time window [178], [215]. However, such aggregation can lose temporal information [214], so later refinements introduced time-weighted sums (giving higher weight to recent snapshots) or exponential decay factors to emphasize current graph structure [179]–[181], [216].

Beyond matrix factorization, random walk-based embeddings (e.g., DeepWalk [217] and Node2Vec [218]) are also adapted: one could generate embeddings for each snapshot via truncated random walks and then align or smooth these embeddings over time. For instance, some methods initialized the embedding at time t with the result from $t - 1$ and fine-

tuned it, thereby leveraging past computations to speed up training [173], [182]. Along with the development of deep learning, DynGEM [183] is one of the pioneering methods which utilizes deep autoencoders to learn node embeddings on each snapshot, while incorporating a regularization term to encourage the embeddings of adjacent snapshots to remain similar, thus capturing temporal smoothness. Dynamic-Triad [177] further incorporated triadic closure process into structural modeling. Overall, these pre-GNN methods laid the groundwork by combining static graph embeddings with temporal smoothing or forecasting. They captured evolving structure to some extent, but often in a two-step or loosely coupled manner (separate static embedding and temporal alignment), without end-to-end learning of dynamic patterns.

b) GNN+RNN-based Methods

The emergence of deep learning on graphs introduced end-to-end models for DTDG representation learning that integrate GNNs [3], [219], [220] with sequence models (i.e., RNNs [1]). A dominant early paradigm for DTDGs is the combination of GNNs for structural encoding and RNNs for temporal encoding. We refer to this model design as the GNN+RNN-based architecture. Without loss of generality, and assuming a fixed node set V across time steps for notational simplicity. Given a sequence of graph snapshots $\mathcal{G} = \{G_1, G_2, \dots, G_T\}$, each snapshot is defined as $G_t = (V, E_t, X_t, F_t)$, where $X_t \in \mathbb{R}^{|V| \times d_x}$ is the matrix of node features, and $F_t : E_t \rightarrow \mathbb{R}^{d_e}$ provides edge features such as interaction types and weights. At each time step t , a GNN is applied to capture spatial dependencies within the snapshot. For each node $v \in V$, the GNN aggregates information from its neighborhood $\mathcal{N}_t(v)$, considering both node features and associated edge features:

$$h_v^{(t)} = \text{GNN}(v, \mathcal{N}_t(v); X_t, F_t). \quad (3)$$

The resulting structural embeddings $\{h_v^{(t)}\}_{v \in V}$ serve as input to a RNN, which models temporal evolution by tracking each node's state across time:

$$z_v^{(t)} = \text{RNN}(z_v^{(t-1)}, h_v^{(t)}). \quad (4)$$

Here, $z_v^{(t)}$ denotes the final representation of node v at time t , encoding both current structural context and historical dynamics. This two-stage design, which uses a GNN for intra-snapshot relational modeling and an RNN for inter-snapshot temporal modeling, enables end-to-end learning of evolving node representations in dynamic graphs.

In this architecture, each graph snapshot (at a discrete time step) is first processed by a GNN (such as a graph convolutional network [219]) to produce node embeddings that capture the structural neighborhood information at that time [170]. Then, an RNN (typically an LSTM [4] or GRU [221]) takes these node embeddings from each snapshot as sequential inputs to produce updated node states, thereby propagating information through time [170].

The GNN+RNN-based architecture allows the model to learn how a node's representation should evolve based on its current neighbors and its past states.

Early examples include the model GCRN [184], which applies a GCN followed by an LSTM to capture structured sequences, and the work proposed by Narayan and

Roe [185] using GraphSAGE [222] with an LSTM for representation learning. Subsequent efforts expand on this GNN+RNN paradigm with diverse architectural innovations. TGCN [186] and TNA [187] tightly integrate GCNs with recurrent units to learn temporally-aware node embeddings, while VGRNN [188] brings in variational inference for modeling temporal uncertainty. LRGCN [189], E-LSTM-D [190], and GC-LSTM [197] further enhance temporal modeling through stacked LSTMs or by embedding GCNs into recurrent gating mechanisms.

More recent advances emphasize structural adaptability and computational efficiency: EvolveGCN [191] evolves GCN parameters over time via LSTM; dyngraph2vec [192] combines static embeddings with RNNs for sequential modeling; ROLAND [198] adapts node embeddings via dynamic recurrent controllers. Methods such as TeMP [193], WD-GCN/CD-GCN [194], and HDGNN [195] incorporate attention, hierarchical aggregation, and heterogeneity-aware modules to enhance long-term dynamics modeling.

Additionally, models like HTGN [196], MegaCRN [202], and DEFT [203] explore hyperbolic spaces, meta-structure learning, and spectral wavelets respectively. Others target specific application contexts: RPC [199], RETIA [201], and SEIGN [200] focus on temporal knowledge graphs; STGNPP [204] and WinGNN [205] address spatio-temporal prediction via Transformers and sliding windows; SpikeNet [206] leverages spiking neural dynamics; and TTGCN [207] exploits multi-scale truss structure for hierarchical temporal modeling.

These models effectively treat dynamic graph learning as a synchronous sequence modeling problem, where the RNN absorbs the node's history across snapshots. Despite architectural differences, all such GNN+RNN-based models address temporal dynamics by maintaining a state that carries information from prior snapshots. However, they also inherit limitations of RNNs: difficulty with very long-term dependencies and high memory usage for long sequences [223], [224]. Likewise, stacking many GNN layers per time step can cause over-smoothing of node features [225], [226]. These challenges motivated the next generation of methods that seek to handle time without a standard RNN.

c) Attention-based Methods

Motivated by the success of attention mechanisms in capturing long-range dependencies across various sequence modeling tasks [2], researchers have started to explore their application to DTDG representation learning. This has led to the emergence of attention-based methods.

DySAT [208] is a representative two-stage model that replaces GNNs and RNNs with structural and temporal self-attention, enabling long-range dependency modeling while retaining architectural interpretability. TEDIC [209] uses graph diffusion and Temporal Convolutions (TCN) to model interaction dynamics, supporting permutation invariance and efficient global context capture. DyHATR [210] combines hierarchical attention over heterogeneous relations with temporal attentive RNNs for evolving pattern modeling.

Later models push further toward non-recurrent and scalable designs: DREAM [211], focusing on temporal knowledge

graphs, incorporates multi-faceted attention with imitation learning to enhance reward modeling in multi-hop reasoning tasks. STGNP [212] models extrapolation via a neural latent variable framework with causal temporal convolutions and Bayesian graph aggregation to capture uncertainty and spatial dependencies. DTFormer [170] adopts a fully attention-based architecture, processing structural and temporal patterns jointly with multi-patch encoders and position encodings, achieving strong scalability and predictive performance.

These models collectively shift from recurrent temporal modeling to attention-based or convolutional paradigms, emphasizing scalability, long-term dependencies, and integration of complex spatial-temporal semantics.

2) Continuous-Time Dynamic Graphs

As DTDGs discretize time into a sequence of graph snapshots, each capturing the network state within a fixed interval, this formulation simplifies modeling by leveraging established techniques for static graphs, but it may overlook fine-grained temporal information and introduce artificial boundaries between interactions. In contrast, CTDGs model data as sequences of timestamped interaction events, allowing for more precise capture of asynchronous, real-world dynamics. This finer granularity enables CTDGs to better represent time-sensitive behaviors in domains such as financial transactions [214], [279] and e-commerce [5], [280], [281]. As a result, CTDGs have attracted growing attention as a complementary approach to snapshot-based modeling, particularly in scenarios where modeling temporal precision and event order is critical [5].

Moreover, CTDGs are also commonly referred to as Temporal Interaction Graphs (TIGs) or event-based temporal graphs. In this paper, we regard these terms as equivalent and collectively refer to them as CTDGs. DTDGs can also be transformed into CTDGs [170], [282], [283] by assigning timestamp information from each snapshot to the corresponding edges. Intuitively, this conversion often results in multiple edges sharing the same timestamp, reflecting the coarser temporal granularity inherent in DTDGs.

In CTDGs, structural and temporal information are often intricately intertwined, necessitating joint modeling to accurately capture the evolution of interactions over time. As temporal dependencies critically influence both node behavior and structural dynamics [213], [284], we adopt a temporal modeling perspective as the primary taxonomy dimension, as it highlights the fundamental differences in how models encode, propagate, and reason about time-dependent interactions. Based on this criterion, we categorize existing CTDG methods into five principal classes. **(a) RNN-based Methods:** Use recurrent neural networks (e.g. LSTM/GRU) to sequentially integrate historical interactions, updating node embeddings over time. **(b) TPP-based Methods:** Employ temporal point processes (e.g., Hawkes processes) with conditional intensity functions to model event timing, capturing both historical decay and spontaneous interactions. **(c) Random Walk-based Methods:** Generate temporally constrained random walks that respect chronological order, aggregating spatio-temporal patterns from sequences of timestamped node transitions. **(d) Attention/Time Encoding-based Methods:** Apply self-

attention mechanisms to sequences of event, enabling the model to capture long-range temporal dependencies without strict recurrence. However, since the attention mechanism is inherently position-invariant, time encoding is commonly introduced to explicitly incorporate temporal or positional information. In this category, we group together both methods that use attention-based architectures and those that, while not using attention mechanism, but still incorporate explicit time encoding to model temporal information. **(e) MLP-based Methods:** The necessity of complex temporal architectures has been challenged by adopting streamlined designs for temporal modeling. These methods leverage MLPs as the core building blocks for encoding temporal interactions, often combined with simple aggregation mechanisms and fixed or lightweight time encoding schemes.

In addition to the primary temporal modeling paradigm, another important dimension in CTDG modeling is the use of memory mechanisms [285]. The memory mechanism introduces the idea of maintaining persistent, continuously updated representations of nodes to track their evolving states over time. Rather than recomputing node representations from scratch, memory-based models incrementally update these state vectors, enabling efficient, asynchronous updates and capturing long-term temporal dependencies. Such memory modules often incorporate temporal encoding or attention to regulate the influence of past events, making them well-suited for modeling evolving behaviors in real-world systems. Importantly, the use of memory is orthogonal to the choice of temporal modeling paradigm and can be viewed as a secondary classification dimension. Thus, by organizing models first by their temporal modeling strategy and then by their use (or absence) of memory, we obtain a two-level taxonomy that more precisely characterizes how CTDG models manage evolving temporal and structural information.

A comparison among CTDG representation learning methods is shown in Tab. IV. Nevertheless, regardless of taxonomy, most methods adopt a similar training paradigm [286]: the backbone model is first “pre-trained” via a self-supervised link prediction task, and the resulting node embeddings are then used for downstream tasks, where a projection head is trained. The pipeline is illustrated in Fig. 7b.

a) RNN-based Methods

A seminal work that initiated RNN-based representation learning in CTDGs is DeepCoevolve [227]. DeepCoevolve [227] introduced a co-evolutionary framework with mutually-recursive RNNs and temporal point processes, laying the groundwork for continuous-time event modeling. JODIE [17] extends this by introducing a projection mechanism to estimate future node states given elapsed time, enabling real-time, time-aware predictions.

Subsequent models further enhance this paradigm in various ways: Know-Evolve [228] and RE-Net [229] focus on temporal knowledge graphs, combining RNNs with multivariate point processes or structural aggregation for autoregressive link prediction. HierTCN [230] replaces sequential updates with a hybrid RNN-TCN architecture to capture both short- and long-term user dynamics. DynGESN [231] replaces trainable RNNs with an Echo State Network, offering a lightweight

TABLE IV: Continuous-time dynamic graph representation learning methods.

Method	Structure Encoding	Temporal Encoding	Memory-based	Year
RNN-based Methods				
DeepCoevolve [227]	Implicit (via sequential interactions)	RNN	No	2018
JODIE [17]	Implicit (via sequential interactions)	RNN + Projection	No	2020
Know-Evolve [228]	Implicit (via sequential interactions)	RNN	No	2017
RE-Net [229]	GCN	RNN	No	2020
HierTCN [230]	Implicit (via sequential interactions)	GRU + TCN	No	2019
DynGESN [231]	Implicit (via sequential interactions)	Echo state network	No	2021
DyGNN [232]	GNN	LSTM	No	2020
AER-AD [233]	Local anonymous subgraph	GRU	No	2023
RTRGN [234]	Implicit (via sequential interactions)	RNN	No	2023
TGN [5]	Implicit (via message passing)	RNN + Memory	Yes	2020
NAT [235]	Implicit (via sequential interactions)	RNN	Yes	2022
GDCF [236]	Spatiotemporal GNN	RNN + Memory	Yes	2023
CDGP [237]	Community message passing	Time-aware aggregation	Yes	2023
TIGER [238]	Implicit (via message passing)	RNN + Dual memory	Yes	2023
RDGSL [239]	Implicit (via message passing)	RNN + Memory	Yes	2023
PRES [240]	Implicit (via message passing)	GMM-guided memory correction	Yes	2024
Ada-DyGNN [241]	Reinforced neighbor update	Time-based Policy	Yes	2024
SEAN [242]	Representative neighbor selector	RNN + Temporal-aware aggregation	Yes	2024
MemMap [243]	Latent memory-cell grid	Systematic memory routing	Yes	2024
MSPipe [244]	Implicit (via message passing)	Staleness-aware update	Yes	2024
TPP-based Methods				
HTNE [245]	Historical neighbor modeling	Hawkes process	No	2018
M ² DNE [246]	Micro/Macro temporal co-occurrence	Hierarchical TPP	No	2019
GHN [247]	Entity-level structure modeling	Hawkes process	No	2019
DyRep [248]	Attentive structural encoding	Multi-scale TPP	No	2019
LDG [249]	Edge-quality structure adaptive	Adaptive TPP	No	2021
TREND [250]	Implicit (via sequential interactions)	Hawkes process + Transfer function	No	2022
DynShare [251]	Implicit (via sequential interactions)	Personalized TPP	No	2023
EasyDGL [252]	GAT	TPP + Correlation masking	No	2024
Random Walk-based Methods				
CTDNE [253]	Timestamp-respecting walk	Skip-Gram over walks	No	2018
HNIP [254]	Temporal random walk	Time-decay in walk sequence	No	2020
CAW [255]	Causal Anonymous Walk	Hitting-count encoding	No	2021
NeurTWs [256]	Motif-guided random walk + ODE	ODE over walk path	No	2022
PINT [257]	Implicit (via message passing)	Provable temporal message passing	No	2022
TPNet [258]	Time-decayed walk matrix	Temporal relative encoding	No	2024
Attention/Time encoding-based Methods				
TGAT [259]	Temporal self-attention	Functional time encoding	No	2020
TCL [260]	Transformer	Functional time encoding	No	2021
OTGNet [261]	Open graph attention	Extended time encoding	No	2023
TGRank [262]	Temporal attention ranking	Enhanced time encoding	No	2023
DHGAS [263]	Heterogeneous GNN + Attention	Time encoding	No	2023
DyG2Vec [264]	Temporal edge encoding	Time encoding	No	2023
SimpleDyG [265]	Transformer	Time and position encoding	No	2024
Todyformer [266]	GNN+Transformer	Position encoding	No	2024
DyGFormer [267]	1-hop neighbor + Co-occurrence	Time and position encoding	No	2024
APAN [268]	Mailbox + Attention	Time encoding	Yes	2021
iLoRE [224]	Re-occurrence + Identity attention	Time encoding	Yes	2022
TDGNN [269]	GNN + Time-decay weighting	Exponential decay kernel	No	2020
DGEL [270]	Recent interactions	Time-aware normalization	No	2023
SUPA [271]	Implicit (via sequential interactions)	Time modeling mechanisms	No	2023
FreeDyG [272]	Fourier-enhanced GNN	Functional time encoding	No	2024
CNE-N [273]	Hash table-based memory	Temporal-diverse memory	Yes	2024
TG-Mixer [274]	Clustering patterns	Time encoding	No	2024
DyGMamba [275]	Mamba	Time encoding	No	2025
MLP-based Methods				
GraphMixer [276]	MLP + Mean pooling	Fixed time encoding	No	2024
RepeatMixer [277]	MLP + Repeat-aware sampling	Time-aware aggregation	No	2024
BandRank [278]	Frequency-band MLP	Band-pass time filters	No	2025

alternative for sequential modeling.

Incorporating structural information, DyGNN [232] inte-

grates LSTM updates with GNN-based context under an event-driven scheme. AER-AD [233] focuses on inductive

anomaly detection by learning anonymous edge representations and modeling temporal edge sequences via GRUs. Finally, RTRGN [234] enhances temporal neighbor aggregation through recurrent state updates and a heterogeneous revision module, improving the completeness and accuracy of dynamic representation.

The memory mechanism in CTDG models is first proposed in the paper of TGN [5]. TGN [5] maintains a timestamped memory vector $\mathbf{s}_i(t)$ for each node, which is updated on each interaction event. Upon an event between nodes i and j at time t , TGN computes messages $m_i(t)$ and $m_j(t)$ from their current memory states, the edge attributes, and time intervals. These messages are optionally aggregated (e.g., mean or last event) and passed through an RNN-based memory-updater (e.g., GRU) to produce updated node memories at time t . A node's embedding $\mathbf{z}_i(t)$ is then computed from its memory (and possibly neighbor context) for downstream tasks. This architecture enables long-term historical information retention and online embedding updates with event-level granularity.

The following memory-based methods largely follow the architectural framework established by TGN. These subsequent models retain this core pipeline—maintaining persistent node memories updated upon each interaction—but introduce various enhancements.

NAT [235] introduces a neighborhood cache (N-cache) that stores multi-hop historical context per node, enabling query-time construction of temporally-aware embeddings. GDCF [236] and CDGP [237] adopt memory modules for spatio-temporal prediction and community-aware dynamics, respectively, by maintaining evolving node or community states.

Several works tackle memory staleness and adaptability: TIGER [238] employs dual memory (pre/post-event) and a restarter module to support parallel and context-aware updates; PRES [240] uses a GMM-based correction mechanism guided by gradient trends to adjust memory for robust long-term retention; Ada-DyGNN [241] introduces a learned policy for selective neighbor memory updates, enhancing efficiency and relevance.

Other models focus on addressing robustness and personalization: RDGSL [239] integrates dynamic noise filtering with attention-based edge selection; SEAN [242] proposes a plug-and-play neighborhood encoder with personalized neighbor selection and temporal aggregation. MemMap [243] maps nodes into latent semantic memory cells to encode high-level evolving patterns, while MSPipe [244] decouples memory access from event order using a minimal staleness pipeline and resource-aware scheduling.

These methods highlight the growing trend of memory-centric architectures that move beyond traditional message passing to enable robust, asynchronous, and temporally rich dynamic graph representations.

b) TPP-based Methods

The TPP-based methods directly model event occurrence intensities based on the history of event timestamps and use learned intensity functions or personalized projection operators that condition on past events. The representation of temporal

dependencies is encoded statistically through the intensity functions, thus, they do not require extra memory modules.

HTNE [245] and GHN [247] employ Hawkes processes to model the excitation effects of historical neighbors, capturing inter-event influence in dynamic graphs and TKGs. M²DNE [246] introduces a hierarchical TPP framework for modeling both edge-level (micro) and network-level (macro) dynamics. DyRep [248] unifies communication and association dynamics into a multiscale TPP model with attention-based structural encoding.

Later models extend expressiveness and interpretability: LDG [249] focuses on edge durability over time, while TREND [250] and DynShare [251] combine individual and collective temporal dynamics via learnable intensity functions or personalized projections. EasyDGL [252] integrates TPP-modulated attention, a principled masking-likelihood training scheme, and graph spectral perturbations to support scalability and global interpretability.

These models highlight the strength of TPP-based frameworks in modeling continuous-time dynamics with high temporal precision, offering strong support for event prediction and temporal reasoning in CTDGs.

c) Random Walk-based Methods

In Random Walk-based techniques, the model constructs representations by sampling time-respecting paths (walks) and aggregating information along these trajectories. This path-centric process captures spatio-temporal motifs without storing explicit node states. The node's representation is computed from sampled walks. As such, these methods avoid the complexity and overhead of maintaining per-node memory.

CTDNE [253] is a pioneering approach that generates timestamp-consistent walks and applies Skip-Gram for embedding learning. HNIP [254] enhances this with high-order temporal modeling and a guided autoencoder for supervised walk reconstruction.

CAW [255] introduces causal walks and anonymous encoding to support inductive learning and generalization to unseen nodes, achieving a good balance between expressiveness and scalability. NeurTWs [256] encodes temporal motifs from biased walks using neural ODEs and contrastive learning, capturing fine-grained temporal regularities.

PINT [257] and TPNet [258] formalize temporal walk aggregation via principled message-passing frameworks, enabling provable temporal expressivity and scalable pairwise representation learning through time-decayed walk matrices or random feature propagation.

These methods highlight the path-centric perspective in CTDGs, offering strong generalization, interpretability, and compatibility with transductive and inductive settings by leveraging temporal ordering and walk-based encoding.

d) Attention/Time Encoding-based Methods

TGAT [259] introduces temporal self-attention with functional time encoding based on Bochner's theorem [287], enabling translation-invariant modeling of time intervals and supporting inductive learning. TCL [260] extends this with a two-stream Transformer and co-attention fusion, trained via contrastive learning to enhance robustness. OTGNet [261] and

TGRank [262] further extend temporal attention to handle open dynamic graphs and dynamic link ranking, respectively.

Beyond node-level temporal modeling, DHGAS [263] applies neural architecture search to discover optimal attention-based architectures for dynamic heterogeneous graphs. DyG2Vec [264] introduces a window-based attention-driven encoder that captures temporal motifs via relative time encoding and temporal edge features. It also adopts a non-contrastive self-supervised joint-embedding framework to learn task-agnostic representations. SimpleDyG [265] explores the feasibility of directly applying vanilla Transformers by encoding node interaction sequences with temporal alignment tokens. Todyformer [266] introduces a structure-aware tokenization mechanism that preserves holistic structural and temporal dependencies when converting dynamic graphs into Transformer-compatible sequences. DyGFormer [267] enhances long-history modeling through neighbor co-occurrence encoding and a patching mechanism, effectively adapting standard Transformers for dynamic link predictions.

While several attention-based approaches directly model temporal interaction graphs through self-attention layers—effectively capturing temporal dependencies via time encodings and long-range attention—they typically treat each interaction independently, without retaining historical context beyond attention windows. Several works propose to integrate explicit memory mechanisms into Transformer architectures.

APAN [268] introduces a mailbox-based mechanism, where each node asynchronously receives and queues messages triggered by neighbor events, and performs attention-based summarization only when queried. iLoRE [224] further enhances temporal modeling via a hybrid design: an adaptive short-term updater filters noise, an attention-based long-term updater employs identity-aware attention, and a re-occurrence graph module captures repeated interactions.

Although, attention-based models often employ time encodings to capture temporal ordering, time encoding is not exclusive to the Transformer architecture. A number of methods adopt explicit time encoding strategies within non-Transformer architectures. These models aim to integrate temporal signals into their representations through different time related embeddings.

TDGNN [269] applies exponential decay in its temporal aggregator to weight edges by recency. DGEL [270] updates embeddings in real time for recommendation, aligning structural evolution with normalization constraints. SUPA [271] introduces time-aware short-term decay, time-sensitive propagation, and incremental learning tailored for multiplex heterogeneous graphs.

To capture temporal patterns more expressively, FreeDyG [272] integrates functional time encoding with Fourier-based frequency enhancement for periodicity modeling. CNE-N [273] compresses neighborhoods into hash-based memory and introduces temporal-diverse mechanisms for multi-scale dynamics. TG-Mixer [274] models interaction burstiness via silence decay and clustering-aware sampling of historical links. DyGMamba [275] leverages SSMs to capture long-term temporal dependencies by maintaining compact latent states, thereby enabling efficient and scalable temporal

reasoning over long event sequences.

These methods demonstrate the versatility of temporal modeling in dynamic graphs, incorporating decay, frequency, memory compression, and burstiness to address complex, real-world temporal dynamics.

e) MLP-based Methods

Recent years have witnessed a surge in models leveraging complex architectures such as recurrent neural networks (RNNs) and Transformers to capture fine-grained temporal dependencies. While these approaches have achieved strong empirical performance, they often come at the cost of increased model complexity, slower training, and limited interpretability. As a response to this growing complexity, an emerging line of research has begun to question whether such sophisticated mechanisms are truly necessary. This has led to a re-examination of simpler alternatives—such as MLPs—as potentially sufficient for effective temporal representation learning, thereby inspiring methods like GraphMixer that demonstrate competitive performance through conceptually and technically streamlined designs.

GraphMixer [276] replaces RNNs and attention with MLP-based link encoders, mean-pooling aggregation, and fixed time encoding, offering a fast and effective alternative. RepeatMixer [277] emphasizes repeat behavior modeling by introducing repeat-aware sampling and time-sensitive aggregation across different orders. BandRank [278] adopts a frequency-disentangled approach, decomposing temporal signals into multiple bands using adaptive filters, and employs frequency-enhanced MLPs along with a novel Harmonic Ranking loss to stabilize multi-scale supervision.

These models reflect a growing trend toward simpler yet targeted architectures that leverage signal decomposition, behavioral priors, and lightweight designs to achieve efficient and robust dynamic graph representation.

3) Optimization on Training Frameworks

In contrast to DTDG models, which are typically trained in an end-to-end manner for link prediction task, CTDG models commonly adopt a “pre-train, then fine-tune” training paradigm [286], where a backbone encoder is first trained on temporal interaction data and task-specific predictors are subsequently optimized for different downstream tasks (as illustrated in Fig. 7b). Owing to the fine-grained temporal dependencies and event-driven modeling mechanisms in CTDGs, large-scale training and deployment of CTDG models are generally more complex than their DTDG counterparts. Therefore, in the following, we primarily focus on recent advances in training frameworks tailored for CTDG models.

Beyond fundamental CTDG representation learning methods, new frameworks have emerged to boost both efficiency and effectiveness. These frameworks are generally motivated by two key needs. (1) Scalability across large, evolving graphs: As CTDGs grow and update rapidly, traditional single-GPU or static training pipelines struggle to keep up. Frameworks designed for multi-GPU acceleration address this by parallelizing event processing, optimizing storage structures, and enabling distributed training. (2) Adaptability to downstream tasks under semantic and temporal shifts: CTDGs often feature complex, time-evolving relational patterns that standard pre-

training setups do not fully capture. Prompt-based learning frameworks step in here: by freezing backbone parameters and injecting lightweight, task/time-conditioned prompts, they enable models to adapt dynamically across temporal domains and tasks—without heavy retraining.

Frameworks like TGL [288], DistTGL [289], SPEED [290], GNNFlow [291], and Deep-Graph-Sprints (DGS) [292] address the challenge of training on large-scale CTDGs by optimizing system-level bottlenecks. For instance, TGL introduces a time-sorted Temporal-CSR structure, parallel temporal sampling, and random-chunk scheduling to enable efficient neighbor retrieval and mini-batch updates, achieving large speedups on multi-GPU setups. DistTGL extends this to distributed environments, synchronizing node memory across workers to maintain throughput and convergence. SPEED contributes by offering Streaming Edge Partitioning and Parallel Acceleration Component, while GNNFlow layers on adaptive, time-indexed blocks, GPU/CPU hybrid caching, and distributed training support. DGS introduces a recurrent, sampling-free embedding architecture for CTDGs that updates node states online using vectorized forgetting factors, learnable feature embeddings, and mixed-mode automatic differentiation for efficient end-to-end learning. Collectively, these frameworks tackle storage, sampling, scheduling, caching, and parallelism to scale CTDG training reliably.

TIGPrompt [286] innovatively brings the “pre-train, prompt” training paradigm to CTDG by injecting lightweight, time- and task-aware prompts, which is designed to enhance the adaptability of CTDG models by bridging the temporal and semantic gaps between pre-training and downstream tasks. It introduces three different temporal prompt generator that produces personalized, time-aware prompts for each node, enabling efficient adaptation without modifying the pre-trained model.

D. Textual Data

Behavioral data are often expressed in textual form, arising from user-generated content and textual interactions, such as online reviews, chat logs, social media posts, and conversational records. Such textual behavioral data naturally carry rich semantic information about user intent, preferences, and contextual interactions.

LLMs, such as GPT-4o [293], Gemini [294], PaLM [295], and LLaMA [296], have demonstrated remarkable capabilities in understanding and generating human-readable text, making them particularly well-suited for modeling textual behavioral data. Beyond traditional NLP tasks, their influence has extended to a wide range of deep learning applications, including education [297], healthcare [298], finance [299], and recommendation systems [300].

Consequently, recent research has begun to explore the application of LLMs to behavioral data. By integrating extensive open-world knowledge and emergent abilities such as reasoning, LLMs can effectively analyze entity preferences from textual behavioral traces and enrich the semantic understanding of both entities and behaviors, thereby significantly improving the quality of behavioral data representations. Moreover, LLMs

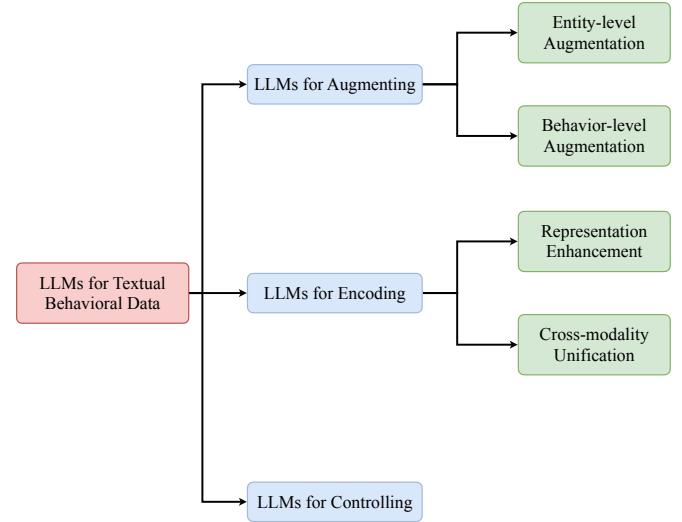


Fig. 8: The dissection of the LLM methods for textual behavioral data.

enable modeling more complex behavioral scenarios, such as conversational behaviors [301], explainable behaviors [302], and real-time behaviors [303], [304].

Based on the role of LLMs within the pipeline, existing LLM-related methods for behavioral data can be broadly categorized into three groups: (1) **LLMs for Augmenting**, (2) **LLMs for Encoding**, and (3) **LLMs for Controlling**. We will introduce these categories in the following subsections.

1) LLMs for Augmenting

LLMs for Augmenting refers to the process of selecting, transforming, and enriching raw behavioral data to create more expressive features/behaviors that are more readily understood by deep learning methods like time series models and dynamic graph approaches. In this process, LLMs take the original data (*e.g.*, entity descriptions, profiles, behavioral features, timestamps) as input and generate auxiliary textual data for augmentation, with various objectives such as enriching the features of entities/behaviors or synthesizing more informative behaviors for both training and evaluation. Based on the ways of augmentation mentioned above, it is easy to further classify these methods into two categories: (a) **Feature Augmentation** and (b) **Behavior Augmentation**.

a) Feature Augmentation

Integrated with exceptional reasoning capabilities and vast parameterized knowledge, LLMs can serve as experts to provide external knowledge during training. This enables them to generate auxiliary features that could promote the understanding of entities and their behaviors. For instance, KAR [305] leverages LLMs to produce user-side behavioral insights and item-side factual information in recommender systems, which are then incorporated as supplementary features into traditional recommendation methods. SAGCN [306] proposes a prompting mechanism to discover semantic, aspect-aware behavioral interactions, providing more fine-grained interpretations of behaviors at the semantic level. In the CUP [307], ChatGPT is used to extract and summarize each user’s behavioral

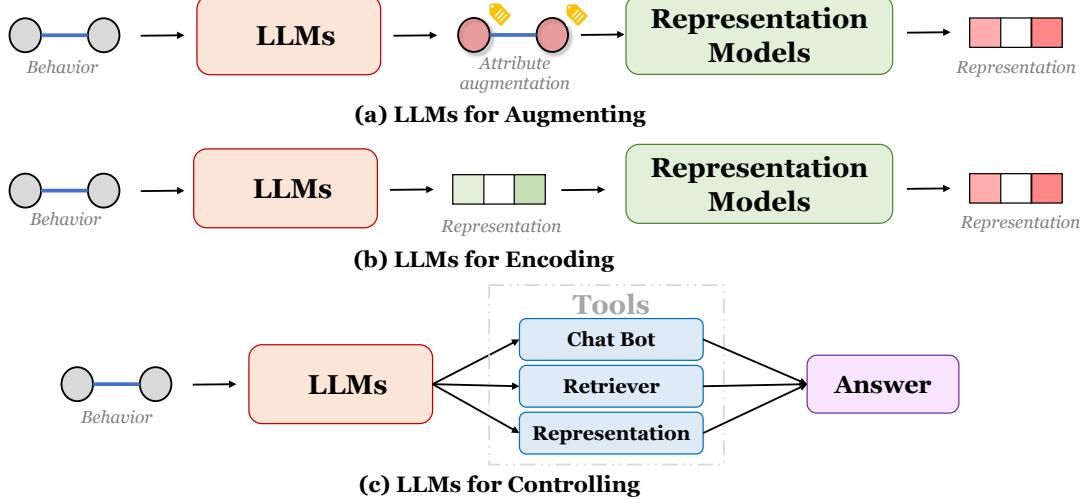


Fig. 9: The pipelines of the LLM-related methods for behavioral data in textual form.

patterns into a set of concise keywords derived from their review history. By condensing user behavioral profiles into just 128 tokens, this method enables efficient encoding with smaller language models that have restricted context windows. In addition to the above methods that rely solely on LLM inference for feature augmentation, some methods [308]–[310] employ instruction fine-tuning with the foundational language models for various tasks across different domains, such as behavior categorization or intent prediction in e-commerce systems. Other methods also leverage LLMs for feature augmentation, including tasks such as text augmentation [311], graph completion [312], [313], and attribute generation [314], [315].

b) Behavior Augmentation

Besides feature-level augmentation, LLMs are also good at generating synthetic behaviors, which can be used to empower the training datasets and subsequently boost the quality of representation generation. For example, in Precious2GPT [316], generative models can be fine-tuned to produce realistic synthetic behavior logs, enriching the training datasets and improving model robustness. In [317], privacy concerns are also being addressed by training LLMs under differential privacy constraints, enabling the generation of synthetic behavioral traces that protect identities while preserving data utility. In [318], prompt engineering techniques are increasingly used to extract richer behavioral summaries, generate user profiles, and even simulate hypothetical interactions, thereby providing more nuanced signals for downstream models. In addition, Trisum [319] facilitates the integration of heterogeneous behavioral datasets by aligning features across domains, making it possible to leverage external sources for improvements. In TF-DCon [320], data condensation strategies have also emerged, where LLMs generate compact yet informative behavioral samples, reducing the volume of training data required without sacrificing representativeness. In e-commerce, LLMs are fine-tuned for tasks such as query rewriting (KAG [321]), helping to bridge semantic gaps for less common or ambiguous behaviors.

2) LLMs for Encoding

In traditional deep learning methods, behavioral data are often formulated as the digitized features through techniques like one-hot encoding, which subsequently allows for an embedding layer to summarize and transform these features into dense vector representations. On the other hand, LLMs are built upon Transformer-based architectures, which excel at modeling contextual dependencies in textual data within behaviors. Such alignment between LLMs and embedding layers makes it intuitive to utilize LLMs as advanced feature encoders for more effective and context-aware representation generation.

Depending on the encoding strategy, existing methods that use LLMs for encoding can generally be categorized into two types: (a) **Representation Enhancement** methods, which utilize semantic information to further elevate representations; and (b) **Cross-modality Unification** methods, which exploit the generalization capabilities of LLMs to unify behavioral data from different modalities¹, thereby synthesizing more expressive and universal representations. The following sections will describe these two categories in detail.

a) Representation Enhancement

In this group of methods, LLMs are utilized as feature encoders for the available textual information within behavioral data, such as user profiles or item descriptions in recommendation systems [300], abstracts or introductions in citation networks [322], and financial documents in financial systems [323]. These approaches typically leverage LLM-based feature encoding in various applications, including news recommendation [324], document ranking [325], tweet search [326], and tag selection [327]. Additionally, beyond static textual features above, there also exist methods that utilize LLMs to capture dynamic and evolving semantic information, thereby modeling temporal dynamics and adapting to shifting preferences over time, especially in user-item

¹Different modalities refer to behavioral data with different distributions, such as series/graphs, different scenarios, different datasets, different systems, etc.

behavioral data among recommendations. For instance, U-BERT [328] enhances user representations by transforming review texts into sequences of dense vectors using BERT [329], and then applies tailored attention mechanisms to effectively model user interests. Similarly, LLM4ARec [98] leverages GPT-2 to extract personalized aspect terms and latent features from user profiles and reviews, thereby improving the quality of recommendations. Unlike the aforementioned methods that directly use the textual features as LLM input, some methods leverage encoded embedding tokens to represent behavioral semantics. For example, LC-Re [330] proposes to integrate collaborative semantic representation information to generate items from the entire item set for recommendation. Similarly, EAGER [331] introduces a two-stream behavior-collaborative architecture that extracts semantic information into embedding tokens, thereby enhancing overall performance in recommendation systems.

b) Cross-modality Unification

In addition to representation improvement methods, the generalization capability of LLMs offers a promising solution for transfer learning and cross-modality unification, where natural language serves as a bridge to align distribution differences across various modalities. For instance, OneLLM [332] proposes a multi-modal LLM that aligns eight types of behavioral data to language within a unified framework, utilizing a multi-modal encoder and a progressive multi-modal alignment pipeline. RUNSRec [333] comprehends universal user behavioral patterns across different domains but also captures their inherent preferences to make recommendations. SocialMind [334] employs human-like perception leveraging multi-modal sensors to extract both verbal and nonverbal behaviors, social factors, and implicit personas, incorporating these social behaviors into LLM reasoning for social suggestion generation. Brooks, et al. [335] present two models available from an API-based suite of emotional expression models that measure nuanced facial and vocal signals, providing rich, high-dimensional emotional expression estimates among human-computer interactions. Uni-CTR [336] utilizes a shared LLM to extract semantic representations at multiple layers, effectively capturing shared features across diverse domains. This approach enhances the model's ability to perform multi-domain recommendation by leveraging the underlying commonalities among different domains. Some works in recommendation systems [337], [338] leverage unified cross-domain textual embeddings from a fixed LLM (e.g., Sheared-LLaMA [296]) to tackle scenarios with cold-start users/items or low-frequency long-tail features. CROSS [339] further explores layerwise mixer to obtain better representations over textual features between the LLM-generated texts and behavior structures.

3) LLMs for Controlling

As the number of model parameters continues to grow, LLMs have demonstrated remarkable *emergent abilities* that smaller models cannot achieve, such as in-context learning, instruction following, self-reflection, and effective tool use. With these capabilities, LLMs can move beyond serving as components within a traditional pipeline. They can take on the role of pipeline controllers or decision-makers across modules

in behavioral data representation learning, thereby enhancing interactivity and interpretability of models.

In recommendation systems, conversational agents like Chat-REC [301] and LLMREC [340] leverage LLMs to interact with users in a dialogue format. These models can infer user preferences from purchase histories, determine when to invoke recommendation APIs, and dynamically filter or rerank candidate items, ultimately delivering more accurate and personalized recommendations. Rather than relying on LLMs to manage the entire recommendation workflow, a more granular approach involves breaking down the recommendation task into smaller, manageable components. RecMind [341] processes behavioral data using a self-inspiring prompting strategy and multi-step reasoning, incorporating tools like expert models, SQL databases, and search engines to enrich the analysis and interpretation of user behaviors.

In finance systems, FinCon [342] mimics an investment firm's structure with “Manager” and specialist “Analyst” agents collaborating via natural language. A Manager LLM consolidates insights from news, reports, and quantitative analysts, while dual-level risk-control agents monitor market risk and update investment beliefs through self-reflection. As for multi-modal behavioral data, FinAgent [343] integrates a GPT-based core with specialized tools: it processes news text, stock charts (vision), and APIs for pricing and analytics. Its pipeline has modules for market intelligence (news summarization), trading behavior generation (via Chain-of-Thought), and portfolio execution. These finance-domain systems [344]–[346] position LLMs not as stand-alone predictors but as decision-making cores in modular architectures. They integrate chain-of-thought reasoning, retrieval or vision tools, and multiple specialized agents to tackle complex tasks like trading, forecasting, QA, and fraud detection.

In social networks, LLMs are central agents or modules in simulations of human behavior. They exhibit multi-turn reasoning and context-awareness, utilize tools, and even self-reflect. For example, AgentSociety [347] integrates environmental data with LLM agents that plan activities, interact, and communicate, where LLMs function as autonomous social actors whose emergent behaviors shed light on community phenomena. ProSim [348] is a modular framework for simulating prosocial behavior, where LLM agents autonomously decide whether to cooperate, punish unfairness, etc. It shows agents naturally exhibit human-like prosocial actions and adjust to policy changes. This illustrates LLMs as modular social actors capable of nuanced, multi-step social reasoning. The research suggests that LLMs can mirror collective dynamics across modules through techniques such as cooperation, tool usage, or reasoning capabilities, making them powerful for controlling the model pipeline in behavioral data processing.

IV. APPLICATION SCENARIOS

Having reviewed the foundational principles and empirical characteristics of major behavioral data representation methods, we now shift focus to their practical applicability across real-world tasks. This section examines the application of various data representations—including tabular, event sequence,

TABLE V: LLM-related methods for behavioral data.

Methods	LLM Function	Modality Transformation	LLM Fine-tuning	Year
LLMs for Augmenting				
Carranza et al. [317]	Behavior augmentation	Language	No	2023
TF-DCon [320]	Behavior augmentation	Language	No	2023
CUP [307]	Feature augmentation	Language	No	2023
Precious2GPT [316]	Behavior augmentation	Multi-omics → Language	Yes	2024
TriSum [319]	Behavior augmentation	Sequence → Language	Yes	2024
Chen et al. [318]	Behavior augmentation	Sequence → Language	No	2024
KAR [305]	Feature augmentation	Language	No	2024
Ghanem et al. [312]	Feature augmentation	Graph → Language	Yes	2024
Ghanem et al. [313]	Feature augmentation	Graph → Language	No	2024
APLe [308]	Feature augmentation	Sequence (multi-modal) → Language	Yes	2024
KAG [321]	Behavior augmentation	Language	Yes	2025
SAGCN [306]	Feature augmentation	Sequence → Language	No	2025
Pandey et al. [310]	Feature augmentation	Sequence → Language	Yes	2025
DyLas [309]	Feature augmentation	Language	Yes	2025
LLMs for Encoding				
U-BERT [328]	Representation enhancement	Language	Yes	2021
Social-LLM [322]	Representation enhancement	Graph → Language	No	2023
Brooks et al. [335]	Cross-modality unification	Sequence → Language	No	2023
UFIN [337]	Cross-modality unification	Sequence → Language	No	2023
Uni_CTR [336]	Cross-modality unification	Sequence → Language	No	2023
GNR [324]	Representation enhancement	Language	Yes	2024
EAGER [331]	Representation enhancement	Sequence → Language	Yes	2024
LC-Re [330]	Representation enhancement	Language	Yes	2024
OneLLM [332]	Cross-modality unification	Graph/Sequence → Language	Yes	2024
LC-Re [330]	Cross-modality unification	Language	Yes	2024
Lu et al. [325]	Representation enhancement	Language	Yes	2025
Chavinda et al. [326]	Representation enhancement	Language	Yes	2025
Poison-RAG [327]	Representation enhancement	Sequence → Language	No	2025
RUNSRec [333]	Cross-modality unification	Sequence → Language	Yes	2025
Socialmind [334]	Cross-modality unification	Graph/Sequence → Language	No	2025
CROSS [339]	Cross-modality unification	Graph → Language	Yes	2025
Lin et al. [338]	Cross-modality unification	Sequence → Language	No	2025
LLMs for Controlling				
RecLLM [340]	Pipeline control	Language	No	2023
RecMind [341]	Pipeline control	Language	No	2023
FinCon [342]	Pipeline control	Language	Yes	2024
FinAgent [343]	Pipeline control	Language	Yes	2024
TradingAgents [344]	Pipeline control	Language	Yes	2025
TS-Reasoner [345]	Pipeline control	Language	Yes	2025
RiskLabs [346]	Pipeline control	Language	Yes	2025
AgentSociety [347]	Pipeline control	Language	Yes	2025
ProSim [348]	Pipeline control	Language	Yes	2025

dynamic graph, and textual—to key application scenarios across five prominent domains: **E-commerce**, **Healthcare**, **Social Media**, **Finance**, and **Gaming**.

A. E-commerce

In e-commerce, user interactions generate large-scale behavioral data, including clickstreams, search queries, and purchase records. Effectively learning representations from these data is crucial for powering intelligent downstream applications. We will now explore the application of behavioral data representations in two pivotal application scenarios in e-commerce: **(1) Personalized Search and Recommendation**, **(2) Enhanced User Understanding and Profiling**.

1) Personalized Search and Recommendation

As the core of e-commerce platforms, personalized search and recommendation aim to match users with the most relevant products by leveraging their behavioral data and contextual

signals. Tabular data has long served as the backbone in e-commerce. Early studies leveraged clickstream logs and transaction tables to build aggregate user profiles for personalization [349]. With the growth of large-scale e-commerce platforms, structured tabular features have become central to collaborative filtering (e.g., NCF [350]) and hybrid deep learning models for recommendation (e.g., DeepFM [26], xDeepFM [27], AutoInt [351]).

By modeling a user’s behavior as a time-stamped list of events, event sequence modeling has become a cornerstone of personalized recommendation. The foundation was laid by methods combining traditional matrix factorization with sequential signals, such as FPMC [352], which integrated first-order Markov chains to capture dependencies on the user’s last action. The paradigm shifted decisively with the introduction of deep learning. GRU4Rec [118] was a seminal work that first applied RNNs to model session-based clickstreams, demon-

strating the power of learning complex sequential patterns.

To overcome the limitations of RNNs in capturing long-range dependencies, the field then embraced attention mechanisms, leading to the rise of attention-based models. SAS-Rec [353], for instance, pioneered the use of a unidirectional Transformer, allowing the model to weigh the importance of all past items to make more context-aware predictions. Building on this architectural foundation, further research has explored innovations in other dimensions. One key direction is enhancing item representations. MARN [354], for example, disentangles item features into modality-specific and modality-invariant components to create a richer representation before feeding it into a sequential model. Similarly, Moreira et al. [355] combine tabular data with textual and image vectors for multi-modal feature learning within a Transformer ensemble. Another direction focuses on optimizing the sequence structure itself. SAGE [166], inspired by DNA expression, organizes user behavior into critical "snapshots" to preserve essential signals while reducing sequence length and computational overhead. Recently, the fusion of tabular data with event sequences has become a popular paradigm in modern e-commerce recommendation. For example, IU4Rec [356] leverages tabular product attributes to construct stable "Interest Units" (IUs) and jointly models fine-grained item sequences with higher-level IU sequences for improved recommendation.

However, representing user-item interactions purely as sequences can overlook latent structural connections that evolve over time. To address this, dynamic graph modeling has been introduced to capture both temporal and structural patterns. There are two key applications of dynamic graph modeling in personalized search and recommendation: First, edge prediction. DGSR [357] connects different user sequences through a dynamic graph structure, and converts the next-item prediction task in sequential recommendation into a link prediction between the user node and the item node. While existing Temporal Graph Neural Networks (T-GNNs) often suffer from scalability and efficiency challenges due to high computational overhead, to address this limitation, a lightweight framework is proposed, named EAGLE [358], which integrates short-term temporal recency and long-term global structural patterns. Second, recommendation and ranking. SRJGraph [359] jointly models search and recommendation with a unified graph neural network to improve ranking performance by alleviating user behavior sparsity.

Nowadays, representing behavioral data with LLMs has initiated a paradigm shift, reconceptualizing user-item interaction sequences as a semantic language of consumer intent. This approach enables a more nuanced, context-aware understanding of user journeys, moving beyond traditional ID-based methods. This paradigm has evolved significantly: Initial work, such as BERT4Rec [360], adapted the masked language model objective from BERT to behavioral data, demonstrating the power of deep bidirectional context. The modern approach, however, directly leverages generative LLMs. A seminal work, P5 [361], established a text-to-text framework using prompts to unify diverse recommendation tasks (e.g., sequential prediction, rating, explanation) within a single T5-based model. Following this, the field of LLM4Rec [362] has burgeoned, exploring the

use of powerful foundation models. Chat-REC [363], which demonstrates how models like ChatGPT can be effectively prompted and guided to act as high-quality conversational recommender agents with minimal fine-tuning. Furthermore, methods like M6-Rec [364] and V-RECS [365] integrate multi-modal information (textual descriptions, images) into the sequence, creating richer item representations that allow the LLM to capture complex semantic and visual preferences, thus mitigating cold-start issues and improving recommendation quality.

2) Enhanced User Understanding and Profiling

In e-commerce, enhancing user understanding through their data is key to personalization. Initially, this was achieved using tabular data processed through extensive feature engineering. This paradigm was dominant in critical industrial tasks such as Click-Through Rate (CTR) prediction and customer modeling, often employing models like logistic regression or factorization machines on massive, sparse feature sets [366]–[368]. For example, the Repeat Buyer Prediction model [369] perfectly illustrates this approach by constructing static profiles for users and merchants from transactional statistics to identify potential loyal customers.

To overcome the limitations of static profiles, later research began leveraging deep learning to model user behavior sequences as event sequences. DUPN [370] demonstrated that a universal user representation could be learned from these sequences and shared across multiple tasks like search and recommendation. Further advancing this direction, HUP [371] introduced a hierarchical profiling method, which captures users' multi-granularity interests by modeling not only item-level sequences but also fine-grained micro-behaviors.

In dynamic graph modeling, enhanced user understanding and profiling is commonly formulated as a node classification problem. For instance, by constructing a tripartite graph connecting users, products, and personas, TriPer [372] reformulates the multi-label persona identification task into a link prediction problem, allowing the model to generalize to new personas not seen during training.

Besides, LLMs excel at generating deep, qualitative insights into user behavior, creating rich, human-readable user profiles from raw interaction logs. This moves analysis from "what" a user will do next to "who" the user is. This paradigm is exemplified by architectures like the Language-based Factorization Model (LFM) [373], which employs an LLM encoder-decoder to distill a user's interaction history into a compact, natural-language summary, thereby enhancing interpretability and addressing cold-start challenges. To refine this process, the GPG [374] method demonstrates that adding a crucial intermediate summarization step is vital; this guides the LLM to effectively parse sparse data and extract salient personal features, significantly boosting personalization accuracy. Furthermore, this approach has evolved to capture the dynamic nature of user preferences. For instance, the framework from Sabouri et al. [375], explicitly models temporal dynamics by generating distinct profiles for a user's short-term versus long-term interests, offering enhanced explainability. Complementing this, the PURE [376] framework focuses on the continuous evolution of profiles by systematically extracting information

from new user-generated text, such as reviews, to maintain an up-to-date understanding of the user. Collectively, these innovations in generating, refining, and dynamically managing textual profiles are paving the way for e-commerce systems that are not only more accurate but also fundamentally more transparent and adaptive.

B. Healthcare

Hospitals and clinics record a continuous flow of patient information, such as admissions, diagnoses, lab results, and treatments. These clinical events, each characterized by a timestamp and metadata, form a patient's Electronic Health Record (EHR). Therefore, this section is constructed around two primary categories: **(1) Disease Prediction and Risk Stratification, (2) Personalized Treatment Recommendation.**

1) Disease Prediction and Risk Stratification

Disease prediction and risk stratification aim to forecast future health events—such as the onset of sepsis, ICU transfer, or hospital readmission—based on a patient's historical clinical behaviors. Behavioral representation learning, particularly from event sequences and tabular EHR data, has enabled significant advancements in this field. For example, models like RETAIN [377] and more recent attention-based frameworks (e.g., BEHRT [378]) learn temporal dependencies from sequences of visits, diagnoses, and medications to produce informative patient embeddings. ProtoEHR [379] further advances event sequence modeling by introducing a hierarchical prototype-based representation framework that captures relationships across medical codes, visits, and patients.

In addition, dynamic graph-based representations are increasingly used to model evolving patient states. MH-GRL [380] utilizes heterogeneous graphs to model patient data, integrating multi-modal attributes and structural relationships for enhanced clinical prediction tasks such as disease prediction and EHR clustering.

Recently, the application of LLMs has expanded this field beyond direct sequence modeling into more interactive and reasoning-driven frameworks. EHRAgent [381] is an LLM-powered agent designed for natural language interaction with EHR systems, capable of multi-tabular reasoning and autonomous code execution to retrieve complex patient information.

2) Personalized Treatment Recommendation

Treatment recommendation systems rely heavily on accurately modeling patient behaviors and responses to previous interventions.

Tabular EHR data, which typically include demographic profiles, laboratory test results, and vital signs, remain a key source for personalized treatment modeling. For example, DeepSurv [382] extends the classical Cox proportional hazards model with deep neural networks to capture nonlinear interactions between patient covariates and treatment effectiveness, thereby enabling individualized survival-based treatment recommendations.

Patient care histories are naturally represented as clinical event sequences, capturing diagnoses, medications, and pro-

cedures over time. To address the patient-specific variability inherent in such sequences, Lee and Hauskrecht [383] proposed adaptive event sequence prediction models that refine population-wide models into subpopulation- and patient-specific dynamics, enabling more accurate and individualized clinical predictions. Building on this line, SRL-RNN [384] introduces a supervised reinforcement learning framework with recurrent neural networks that leverages both clinician prescription signals and long-term survival rewards from EHR sequences, achieving more effective and safer dynamic treatment recommendations.

Besides, knowledge graphs have also become a critical component in enhancing treatment reasoning. KARE [385] integrates LLMs with community-level knowledge graph retrieval to provide reasoning-enhanced predictions in clinical settings. GAME [386] is a graph-augmented representation learning framework that harmonizes multi-institutional EHR data through knowledge graph integration and federated learning, enabling robust cross-site clinical studies while preserving data privacy.

Finally, LLMs are increasingly applied in an end-to-end fashion. Recent advancements in LLMs, such as Med-PaLM 2 [387], have significantly improved medical question answering, outperforming human physicians in certain clinical tasks and demonstrating potential for real-world medical applications.

C. Social Media

The social media domain contains multi-modal, high-velocity behavioral traces (i.e., posts, likes, shares, comments, follows, clicks). To make these signals actionable, researchers have applied diverse representation paradigms. Below we summarize how they are used across key application scenarios and categorize them into four classes: **(1) User Profiling and Interest Modeling, (2) Content Recommendation and Feed Ranking, (3) Misinformation and Rumor Detection, and (4) Toxicity and Safety Moderation.**

1) User Profiling and Interest Modeling

In social media platforms, user profiling and interest modeling aim to infer latent characteristics—such as personality traits, preferences, and evolving interests—from rich behavioral traces. Natural language-based approaches focus on extracting signals directly from user-generated text. For instance, Peters demonstrated that GPT-3.5/4 can predict Big Five personality traits from Facebook status updates in a zero-shot setting, achieving moderate correlations with self-reports [388]. More recently, Sabouri et al. [389] introduced LLM-TUP, which summarizes user histories into natural-language descriptions and encodes them with pre-trained language models, effectively capturing both short- and long-term preferences. Graph-based representations, by contrast, leverage the relational context of users. GREENER [390] modeled news outlets as nodes in a user–media bipartite graph, applying GNNs to predict factuality and bias, while HLMRFs [391] integrated textual content, profile images, and social ties through a structured relational model to jointly infer user demographics and traits. Sequential modeling provides

yet another perspective by emphasizing temporal dynamics of user behavior. Liu et al. [392] proposed SA-LSTM to jointly model user interaction sequences and the influence of friends, while Yang et al. [393] used Bi-LSTMs with a specialized optimizer to capture temporal dependencies in ad click streams for interest prediction.

2) Content Recommendation and Feed Ranking

In social platforms, content recommendation and feed ranking aim to surface the most relevant items from massive content streams by learning expressive user and item representations. Natural language-based methods leverage the semantic richness of text to enhance recommendation. For example, SPAR [394] uses pre-trained language models to encode user histories and candidate item text, introducing sparse poly-attention to effectively handle long engagement sequences and improve personalization. Sequential modeling approaches emphasize temporal dynamics in user behavior. Yang [395] augments user interaction sequences with latent item relations mined from language models, thereby enriching sequential dependencies beyond simple co-occurrence. Meanwhile, Mamba4Rec [396] replaces attention-heavy Transformers with selective state space models to capture long-term dependencies more efficiently, achieving both scalability and accuracy on large datasets. Graph-based approaches bring a structural perspective. ContextGNN [397] integrates pair-wise subgraph representations with global two-tower embeddings, addressing the pair-agnostic limitation of traditional architectures. Similarly, CGAT [398] exploits item knowledge graphs with attention over local and non-local neighbors, enhancing item embeddings and ultimately boosting recommendation accuracy.

3) Misinformation and Rumor Detection

Detecting misinformation and rumors on social media is a critical task, requiring models that can capture both the semantic content of claims and the structural and temporal dynamics of their spread. Natural language-based approaches exploit textual cues to directly judge veracity. Chen et al. [399] conducted a broad empirical study showing that large language models can match strong baselines in content-only classification, while Ruiz et al. [400] went further by integrating argumentation schemes and critical question answering to provide interpretable justifications for misinformation detection. Graph-based methods instead emphasize propagation and relational signals. For example, Liu et al. [401] jointly modeled user correlations and diffusion trees, while GSMA [402] improved GraphSAGE by weighting neighbors according to their structural positions in propagation graphs, achieving stronger accuracy on Weibo datasets. Sequential modeling brings temporal order into focus: Dong et al. [403] captured word- and sentence-level dependencies to improve detection with limited labels, and GETAE [404] combined Bi-RNN text encoders with propagation dynamics for robust ensemble predictions. Hybrid approaches such as MAGIC [405] further integrate multi-modal content and adaptive graph attention to capture both textual semantics and relational context.

4) Toxicity and Safety Moderation

Ensuring safe and respectful online communities requires robust moderation systems capable of detecting toxic or harm-

ful behavior across diverse contexts. Natural language-based approaches remain the backbone of this task: lightweight yet effective transformer models such as Tiny-toxic-detector [406] achieve competitive accuracy with minimal resources, while recent survey [407] synthesize progress across multilingual datasets, highlighting the challenges of implicit language and cultural variation. Graph-enhanced models enrich textual detection with structured knowledge or relational context. MetaTox [408] integrates a meta-toxic knowledge graph with LLMs to provide external toxic concept grounding, while ShieldVLM [409] extends moderation to multi-modal implicit toxicity, using deliberative reasoning across vision-language graphs to capture harmful meanings hidden in text-image combinations. Sequential methods bring temporal and contextual signals into play: Cheng et al. [410] formulates moderation as a sequential decision problem to reduce bias in detection over comment streams, and Oyetayo et al. [411] leverages dialogue context with RNN-based encoders to classify multiple types of toxicity simultaneously. Hybrid multi-modal frameworks further expand the scope, such as MHSDF [412], which fuses textual, visual, and audio/video modalities through attention mechanisms to detect complex hate speech embedded in memes or multi-modal content.

D. Finance

The data in the financial sector are inherently heterogeneous, temporally structured, and often high-stakes. Different behavioral data representations are prominent in specific financial applications. For risk management and fraud detection, event sequences, dynamic graphs, and LLMs are the primary modalities. In contrast, stock forecasting and portfolio optimization frequently rely on tabular data and LLMs. Consequently, we structure our review of these applications around two critical domains: (1) **Risk Management and Fraud Detection**, (2) **Stock Forecasting and Portfolio Optimization**.

1) Risk Management and Fraud Detection

In financial services, user behavioral event sequences have gained increasing attention in recent years for applications in risk management and fraud detection. These sequences effectively capture the distinctive digital behavioral signature of individual users, which is inherently more difficult to forge or manipulate compared to static profile data. By learning expressive representations from these event streams, models can capture subtle temporal patterns and behavioral deviations, thereby offering enhanced capabilities for proactive risk assessment and anomaly detection. Since SimpleRNN networks are prone to the vanishing gradient problem, Xavier et al. [413] find that GRU outperforms both SimpleRNN and LSTM models. Branco et al. [414] treats user behavior as a sequence composed of many interleaved, unbounded subsequences, and use GRUs to build fraud detection models. FinDeepBehaviorCluster [415] combines RNNs with intuitive features generated by risk experts to form a hybrid feature representation, which models the sequence of user clicks for fraud detection. UB-PTM [416] designs three agent tasks at different granularities, such as action, intention, and sequence levels, to detect fraud activities from unlabeled data. CLeAR [417]

devise an Intensity-Aware Transformer to capture dramatic changes in the intensity of transaction amounts and timing, while design two representation learning tasks that utilizes varying levels of contrastive learning to improve representation robustness.

Node representation learning on discrete dynamic graphs divides data into static snapshots at discrete time intervals, then each snapshot is processed with graph network models. Researchers use time-related models to get the user (node) representations. DyHDGE [418] jointly captures temporal transaction dependencies and dynamic heterogeneous graph structures to enable accurate real-time fraud detection in financial scenarios via integrated risk assessment and contrastive learning. EvolveGCN [419] adapts the GCN model over time without relying on node embeddings. Instead, it captures the evolving nature of graph sequences by using a RNN to update the GCN parameters dynamically. FiFrauD [420] turns these real-time transactions into a stream of graphs, utilizing density signals in graphs in unsupervised training paradigm. Unlike discrete dynamic graphs, continuous dynamic graphs capture changes in nodes and edges as they occur continuously over time, rather than at discrete intervals. GADY [421] proposes a continuous dynamic graph model to capture complex time information, and uses a message-passing framework integrated with positional features to generate edge embeddings, which are subsequently decoded to detect anomalies. EL²-DGAD [422] integrates continuous-time embeddings into an attention-based graph encoder model. SSH-T³ [423] then uses a self-supervised hierarchical two-tower Transformer framework that models long, multi-scenario payment behavior sequences via robust day-level pre-training and amount-aware scenario modeling to significantly improve financial risk assessment under scarce labels and complex real-world settings.

LLMs are fundamentally reshaping risk management and fraud detection, moving beyond static rules to analyze dynamic, context-rich data. They are notably applied to detect scams and fraud by analyzing textual and conversational data, where models like GPT-3.5 have proven effective at identifying phishing and other deceptive language [424], leading to higher precision and recall than traditional methods [425]. This capability is exemplified in advanced real-time systems, such as a RAG framework that analyzes phone calls to prevent impersonation and solicitation fraud as they occur [426]. Beyond immediate threats, LLMs enable more comprehensive financial risk assessment by integrating diverse, multi-modal data sources. The RiskLabs framework pioneers this approach by fusing textual and vocal data from earnings calls with market time-series data to forecast financial volatility and variance, demonstrating a more holistic risk prediction capability [427]. Collectively, these applications showcase a shift towards a more predictive and integrative approach to managing financial risk.

2) Stock Forecasting and Portfolio Optimization

In financial modeling, tabular data are advancing stock forecasting and portfolio optimization by leveraging heterogeneous numerical and categorical features to support accurate predictions and interpretable investment strategies.

For stock price forecasting, a common approach is to treat tabular data as a time series. Within this paradigm, the LSTM network has been widely adopted due to its ability to capture long-range dependencies and address the vanishing gradient problem [428]–[430]. Building on this, recent works have explored the Transformer model, which leverages the attention mechanism not only for long-range dependencies but also to explicitly model complex interactions across multiple assets [279]. In contrast to these sequence-focused methods, another line of research modeling the problem by leveraging the inherent attribute of tabular data. For instance, Zhang et al. [431] propose a hybrid model combining a deep neural network with TabNet, a model specifically designed for tabular data, which achieves instance-wise feature selection through its unique attention mechanism. In the context of portfolio optimization, the dominant paradigm is a two-stage, “predict-then-optimize” framework. In this approach, the primary challenge is framed as a time-series forecasting problem as mentioned above, these forecasts are fed into classical optimization algorithms, most commonly the Mean-Variance Optimization (MVO) model, to determine optimal asset allocation [428], [432], [433]. In addition, some end-to-end methods have also been proposed. SARL [434] employs a reinforcement learning agent to make end-to-end portfolio decisions, it explicitly addresses the challenge of information-rich environments by augmenting the agent’s state representation to tackle the challenges of data heterogeneity and market non-stationarity.

At the same time, LLMs are revolutionizing stock forecasting and portfolio optimization by synthesizing diverse data sources for sophisticated investment strategies. A key innovation is their ability to enhance predictive models by integrating unstructured text with traditional market data. For instance, the StockTime [435] architecture treats stock price series as a language sequence for prediction, while LED-GNN [436] uses an LLM to extract inter-stock relationships from news to build dynamic graphs, improving trading volume forecasts. Beyond pure prediction, LLMs are enabling more transparent and customized investment frameworks. The SEP [437] framework trains an LLM to generate human-readable explanations for its stock predictions through a self-reflective process, addressing the “black box” problem. Concurrently, research on Persona-Based Ensembles [438] leverages LLMs to simulate different institutional investor personas, creating adaptive strategies that outperform traditional methods in specific market conditions. These advancements collectively push towards more accurate, explainable, and personalized portfolio management.

E. Gaming

The gaming domain provides an exceptionally rich environment for behavioral data analysis due to the high volume, velocity, and complexity of player interactions. To decipher player intent, skill, and engagement from these intricate data streams, researchers have developed diverse representation learning approaches, which have been applied to a range of application scenarios. In this section, we review representative tasks and highlight how different representation paradigms—such as event-sequence modeling and natural

language-based approaches—are leveraged within them. We organize the discussion from three perspectives: (1) **Player Goal and Strategy Recognition**, (2) **Non-Player Character (NPC) Generation and Control**, and (3) **Toxicity Detection and Community Management**.

1) Player Goal and Strategy Recognition

Understanding player intent and strategy is a foundational task in game analytics. Sequence-based models have shown strong effectiveness here: early work used RNNs and LSTMs to frame goal recognition as sequence labeling, significantly improving prediction performance [439]. In the Real-Time Strategy (RTS) game domain, unsupervised replay embeddings learned from event sequences enabled automatic discovery and labeling of player strategies without manual annotation [440]. More recently, attention-based methods such as player2vec [441] have tokenized in-game events and used self-supervised learning to capture long-range dependencies, offering robust player representations for intent recognition. Moreover, innovations like Masking by Token Confidence (MTC) further refined pre-training by adaptively masking task-relevant actions, improving recognition accuracy in Massively Multiplayer Online Role-Playing Game (MMORPGs) [442].

2) NPC Generation and Control

A second key task is generating believable and adaptive NPCs. Sequence learning approaches demonstrated early promise by using LSTMs with attention mechanisms to generate human-like action sequences, enhancing NPC realism and engagement [443]. With the advent of LLMs, this direction has expanded substantially: LLMs now serve as the reasoning “brains” of game agents. For example, MineDojo provides an internet-scale knowledge base to support embodied agents trained via LLMs [444], and Voyager showcases lifelong skill acquisition through automatic curriculum generation [445]. Other frameworks integrate LLMs with reinforcement learning, where the LLM constructs a skill graph and plan that the RL agent executes to solve long-horizon tasks [446]. Beyond planning, code-tuned LLMs empower NPCs to not only generate free-form dialogue but also produce executable, game-state-altering code, enabling deeply interactive experiences [447].

3) Toxicity Detection and Community Management

Ensuring healthy in-game communities is another important application scenario. Early linguistic analyses characterized the communication patterns of toxic players [448], measured the prevalence of disruptive behaviors [449], and established links between toxicity and game outcomes [450]. Modern approaches extend this line by tackling real-time, multi-modal toxicity detection challenges [451]. Here, LLM-based language representations have been especially valuable in understanding complex chat dynamics. Furthermore, reinforcement learning techniques, such as contextual bandits, have been introduced to allocate monitoring resources efficiently toward sessions with the highest predicted toxicity, improving both detection efficiency and player safety [452].

V. KEY MODULES

Although foundational modeling approaches such as tabular modeling, sequential modeling, and dynamic graph modeling

have achieved significant progress in behavioral data analysis, the inherent complexity and diversity of behavioral data often render these basic methods insufficient for fully capturing critical factors such as temporal dependencies, structural patterns, and contextual information. To address these limitations, researchers have proposed a variety of enhanced modeling strategies designed to further improve the representational capacity and predictive performance of behavior models. These enhancement methods are not standalone modeling paradigms, but rather modular components or mechanisms that can be flexibly integrated into different types of models. They include, but are not limited to, time encoding, position encoding, context modeling, structure-aware mechanisms, contrastive learning, and pre-training. These strategies have demonstrated strong generalizability and effectiveness across a wide range of behavioral modeling tasks in recent years, becoming key drivers in the advancement of behavioral data modeling.

This section provides a systematic overview of these enhanced modeling methods, analyzing their underlying design principles, applicable scenarios, and integration approaches with primary modeling techniques.

A. Time Encoding

In behavioral data modeling, temporal information often contains critical dynamic patterns such as periodicity, time-delay dependencies, and triggering sequences. Whether in user behavior, medical events, or financial transactions, the timing or intervals of actions play a crucial role in understanding their underlying evolution. Early studies typically relied on handcrafted temporal features specifically designed for downstream tasks [453]–[455]. These methods involved decomposing timestamps into components (e.g., year, month, day), assigning embeddings to each component, and summing them to form the final time representation. While effective in certain scenarios, such approaches were labor-intensive, domain-specific, and limited in their ability to capture only predefined temporal patterns [456].

With the rapid development of attention mechanisms, researchers began exploring more flexible ways to model temporal information, particularly in capturing long-term dependencies and dynamic changes. This led to the emergence of Functional Time Encoding (FTE) methods, which aimed to provide time representations compatible with self-attention architectures. Representative works include Functional Time Representation [287] and Time2Vec [456]. These methods apply multiple linear transformations to the raw time input, followed by pre-defined nonlinear functions (e.g., trigonometric functions) to generate multi-dimensional time embeddings. Their ability to effectively encode periodic patterns made them widely adopted in dynamic graph representation learning and time series modeling [457], [458].

Despite the success of functional time encoding in capturing periodic behaviors, these methods inherently rely on strong inductive biases rooted in predefined periodic assumptions [287], [456], [459]. As a result, they often struggle to model more complex temporal patterns such as non-periodic and mixed behaviors. To address these limitations, recent studies proposed more expressive approaches like LeTE [458], which

aim to capture a broader range of temporal dynamics. LeTE introduces neural transformations that allow the model to automatically learn meaningful temporal features directly from raw time values, without relying on handcrafted components or fixed periodic functions. This enables the model to better adapt to the diverse and complex nature of real-world temporal behaviors. In contrast, Chung et al. propose using simpler linear time encodings and show that self-attention can extract temporal patterns from these features while reducing model complexity [460].

The evolution of time encoding techniques has significantly advanced the field of behavioral data analysis. In many real-world scenarios, such as user interactions, clinical events, or financial activities, behaviors are not only determined by what happens, but also by when and in what order they occur. Accurate temporal representations help uncover latent behavioral patterns such as periodicity, recency effects, burstiness, or long-term dependencies. Modern time encoding methods, especially those based on neural transformations, enable models to better capture these complex temporal dynamics without relying on rigid assumptions. This has led to more faithful representations of behavioral sequences, improved interpretability of behavioral trends, and enhanced generalization across diverse behavioral contexts. As research continues, there is a growing need for time encoding techniques that are not only expressive but also tailored to the unique challenges of behavioral data, such as irregular sampling, multi-scale temporal patterns, and context-dependent timing effects.

B. Position Encoding

In early sequence modeling, positional information was either implicitly captured through the use of sequential architectures such as RNNs, or explicitly introduced using integer-based position indices. These approaches allowed models to process tokens in a temporal order, thereby encoding positional dependencies through recurrence or positional tags. However, the emergence of attention-based models—particularly the Transformer—marked a paradigm shift, as these models lack an inherent notion of sequence order. This limitation sparked a surge of research interest in position encoding, aiming to explicitly inject positional information into attention mechanisms. Among the proposed solutions, two main categories have emerged: absolute position encoding, which assigns unique representations to fixed positions, and relative position encoding, which focuses on modeling the distance or offset between tokens. The development of more effective position encoding schemes has since become a central theme in advancing the performance and generalization of attention-based models.

1) Absolute Position Encoding

Along with the attention mechanism, the sinusoidal position encoding is proposed [2], a form of absolute position encoding that maps each position to a unique, analytically defined vector. While this encoding is theoretically extrapolatable to arbitrarily long sequences, subsequent studies [461], [462] revealed practical limitations. Specifically, since the model is trained on sequences of limited length, the attention weights

may fail to generalize to longer sequences during inference, leading to unstable outputs. This highlights a fundamental limitation of absolute position encoding in terms of extrapolation capacity in real-world applications.

In response to the extrapolation challenges of absolute position encoding, researchers have proposed various enhancement strategies. Ideally, position encodings should possess specific properties; notably, shift invariance—the property that a function’s output remains unchanged when its input is uniformly shifted—was emphasized in [463] as a key factor for achieving robust extrapolation. A method involving random positional shifts during training was introduced by [464], where the original position index pos with $pos + k$, where k is sampled from a discrete uniform distribution $u(0, K)$. This approach discourages the model from relying on absolute positions and instead promotes learning of relative positional patterns. Building on this idea, additional perturbations, including global shifts, local jitters, and global scaling, were proposed in [465] to model position encodings as continuous signals with controlled variability to improve generalization. Additionally, representing word embeddings as continuous functions of position has been suggested [463], allowing word representations to evolve smoothly with increasing position indices. Finally, a dynamic systems perspective was explored by [466], directly modeling the temporal dynamics between positional representations. Collectively, these efforts represent significant progress in enhancing absolute position encoding, aiming to improve its generalization and robustness in long-sequence modeling tasks.

2) Relative Position Encoding

While absolute position encodings provide a straightforward way to inject position information into Transformer models, they are inherently limited in their ability to generalize to sequences of unseen lengths or shifted patterns. More importantly, in many sequence modeling tasks, it is not the absolute position of a token that matters most, but rather its relative position with respect to other tokens. For instance, syntactic dependencies or temporal correlations are often defined by how far apart two elements are, rather than their fixed positions in the sequence [467], [468].

Motivated by this observation, relative position encoding was first formalized in [469] by explicitly incorporating pairwise positional relationships into the self-attention mechanism. This approach enhanced the model’s ability to capture local structures and long-range dependencies.

Building upon this foundation, subsequent works extended relative position encoding in various directions. For instance, Transformer-XL [461] introduced a segment-level recurrence mechanism to enable longer context modeling during inference. Similarly, a simplified form of relative position encoding was adopted in T5 [470], where each relative position is mapped to a learnable scalar that is directly added to the attention logits, improving computational efficiency. The concept was further advanced by DeBERTa [471], which decoupled content and positional information and applied the embeddings in a Transformer-XL-style architecture. Additionally, a log-bucket relative position bucketing strategy has been utilized [472]; this method discriminates nearby positions

TABLE VI: Behavioral datasets for tabular representation learning

Dataset	# Numeric	# Categories	# Samples	# Tasks
California Housing (CA)	8	0	20,640	Regression
Adult (AD)	6	8	48,842	Classification
Helena (HE)	27	0	65,196	Classification
Jannis (JA)	54	0	83,733	Classification
Higgs (HI)	28	0	1,000,000	Classification
ALOI(AL)	128	0	108,000	Classification
Epsilon (EP)	2,000	0	500,000	Classification
Year (YE)	90	0	515,345	Regression
Covtype(CO)	54	0	518,012	Classification
Bank(BK)	7	9	45,211	Classification
Blastchar (BC)	3	17	7,043	Classification
Shoppers (SH)	4	14	12,330	Classification
Volkert (VO)	180	1	58,310	Classification
Income (IC)	6	8	32,561	Classification
Yahoo (YA)	699	0	709,877	Regression
Microsoft (MI)	136	0	1,200,192	Regression

more finely while coarsely representing distant ones. Such a logarithmic approach reduces the number of parameters and improves extrapolation capacity.

To further enhance length generalization, strategies such as adding bias terms to the attention scores [473] and incorporating actual distance information [474] have been explored. Notably, ALiBi [475] emerged as the first relative position encoding method specifically designed for length extrapolation. It combines simplicity with strong performance by applying fixed, position-dependent biases to the attention weights.

Inspired by sinusoidal encodings, Rotary Position Embedding (RoPE) is introduced [476], which encodes relative position by rotating the query and key vectors in a position-dependent manner, rather than adding explicit position vectors. This continuous and extrapolatable encoding has been widely adopted in large-scale language models such as LLaMA and ChatGLM, making it one of the most prominent relative position encoding techniques today.

In conclusion, enhancing temporal and position encoding is essential for effective behavioral data analysis. Behavioral sequences often involve complex temporal patterns where the relative timing and order of actions are more informative than their absolute positions. Advanced encoding strategies—such as relative position embeddings and time-aware attention—enable models to better capture dependencies, generalize across varying sequence lengths, and improve both performance and interpretability. These improvements highlight the critical role of encoding-aware designs in modeling real-world behavioral data.

VI. DATASETS AND BENCHMARKS

High-quality datasets play a crucial role in advancing research on behavioral data representation learning, as they provide a solid foundation for model development and evaluation. Corresponding benchmarks further enable fair and consistent comparisons across different methods. In this section, we discuss commonly used behavioral data datasets along with benchmarks, and present relevant statistical information for each.

A. Tabular Data

a) Datasets

To support research on behavior modeling in tabular data formats, both academia and industry have developed a wide range of representative datasets that are widely used for evaluating and comparing model performance. Among them, datasets such as California Housing (CA), Adult (AD), Helena (HE), and Jannis (JA) have been repeatedly adopted in numerous studies [47], [56], [58], [59], [63], and have gradually become standard benchmarks for tabular behavior modeling tasks. These datasets span a variety of prediction objectives—from regression to multi-class classification—and cover diverse application domains, including financial risk assessment, socioeconomic analysis, and local lifestyle services.

Some of these datasets originate from public data science competitions, such as those hosted on the UCI Machine Learning Repository and Kaggle, ensuring a high degree of standardization and broad generality. This contributes to the reproducibility of research and facilitates fair comparison across methods. Moreover, behaviors in these datasets are typically recorded in a structured form—such as samples composed of user attributes and behavioral outcomes—naturally framing them as classification or regression problems, and providing a solid foundation for building more complex behavior modeling frameworks. We summarize the most commonly used datasets in Tab. VI.

b) Benchmarks

To systematically evaluate the performance of various behavior modeling approaches, researchers have proposed a range of benchmark settings to standardize experimental protocols and enable fair comparisons across models. In the context of behavior modeling with tabular data, most datasets are sourced from the open platform OpenML, and many well-established benchmarks in this domain have been built upon it. Depending on their specific evaluation goals, researchers selectively construct benchmark suites by sampling from the OpenML repository.

One of the most widely adopted benchmarks is OpenML-CC18 [477], which was constructed using eight filtering criteria and includes 72 high-quality classification datasets. This benchmark has become a standard reference for evaluating

TABLE VII: Behavioral benchmarks for tabular representation learning.

Benchmark	Paper	Repository	Year
OpenML-CC18	[477]	OpenML	2017
WellTunedSimpleNets	[478]	GitHub	2021
TabularBench	[85]	GitHub	2022
TabZilla	[479]	GitHub	2023
OpenTabs	[480]	GitHub	2024
TALENT	[481]	GitHub	2024
TDBench	[66]	GitHub	2025

TABLE VIII: Sequential behavioral datasets

Dataset	# Users	# Items/Businesses	# Samples	# Domain
Amazon Product Reviews 2023	54.51M	48.19M	571.54M	E-commerce
Amazon Q&A	–	191K	1.48M Q, 4.02M A	E-commerce
ModCloth Marketing Bias	44.78K	1.02K	99.89K	E-commerce
Google Local Reviews 2021	113.64M	4.96M	666.32M	Local services
Google Restaurants	1.01M	65K	1.77M	Local services
Twitch	15.5M	465K	124M	Media content
Food.com Recipe & Review	226.57K	231.64K	1,13M	Media content
EndoMondo Fitness Tracking Data	1.10K	–	253.02K	Healthcare
Behance Community Art Data	63.50K	178.79K	1M	Art
Taobao UserBehavior	987.99K	4.16M	100.15M	E-commerce
MovieLens 32M	200.95K	87.59K	32.00M Ratings	Media content
Steam Video Game and Bundle Data	2.57M	15.47K Items, 615 Bundles	7.79M	Gaming

machine learning models on tabular data. In addition, the influential work [85] selected 45 representative datasets from OpenML based on criteria such as heterogeneity, feature richness, data realism, and task difficulty, creating a challenging and insightful benchmark for comparing deep learning and traditional models.

Building upon this, TaZilla [479] conducted a comprehensive analysis of 176 classification datasets from OpenML. It introduced additional filtering criteria that consider not only task diversity but also practical constraints such as runtime. Furthermore, it ensured broad coverage of datasets commonly used in popular studies, contributing to a more representative and reproducible benchmark foundation. Alongside TD-ColER [66], the authors also propose TDBench, a tabular distillation benchmark covering 23 datasets, 7 model classes, and 11 distillation strategies. We summarize the collected benchmarks in Tab. VII.

B. Event Sequences

a) Datasets

The datasets utilized in event sequence modeling encompass a broad spectrum of domains, such as E-commerce, Local Services, Media Content, and specialized verticals including Healthcare, Art, and Gaming. In these datasets, an event is typically defined as a timestamped user interaction, such as a purchase, rating, review, view, or content engagement.

A significant portion of research focuses on the E-commerce domain. Prominent examples include the Amazon Product Reviews 2023 [482] and Amazon Q&A [483], [484], which capture rich user feedback and interaction histories. In the fashion and retail sector, ModCloth Marketing Bias [485] and the widely used Taobao UserBehavior [486]–[488] are frequently employed to benchmark models on purchasing and browsing sequences.

In the realm of Local Services, event sequences are extensively studied to understand user preferences for physical establishments. Datasets such as Google Local Reviews 2021 [489], [490] and Google Restaurants [490] offer valuable insights into location-based user activities and service interactions. For Media Content, large-scale datasets are used to evaluate recommendation and sequence prediction capabilities across different media types. This category includes Twitch [491] for live streaming interactions, the massive MovieLens 32M [492] for movie ratings, and Food.com Recipe & Review [493] for culinary content exploration.

Furthermore, several datasets target specific vertical domains. These include EndoMondo Fitness Tracking Data [494] for Healthcare, Behance Community Art Data [495] for Art appreciation, and Steam Video Game and Bundle Data [280], [496], [497] for Gaming behavior analysis. We summarize the statistics and characteristics of these event sequence datasets in Tab. VIII.

b) Benchmarks

To evaluate the performance of various approaches, a set of classical and widely effective sequential modeling methods have been selected as benchmarks. These methods are representative works within the categories of time series modeling discussed previously. They are general-purpose and applicable to diverse downstream domains, such as item recommendation, financial forecasting, and healthcare prediction. While domain-specific methods exist, they often build upon these general paradigms (e.g., GRU4Rec in recommendation is based on GRU, BEHRT in healthcare on BERT, and Player2Vec in gaming on the Transformer architecture). These general benchmark methods are summarized in Tab. IX.

TABLE IX: Behavioral sequential representation learning benchmarks.

Benchmark	Paper	Repository	Year
GRU	[118]	GitHub	2015
BERT	[90]	GitHub	2017
CPC	[498]	GitHub	2018
Transformer	[2]	GitHub	2017
PrimeNet	[144]	GitHub	2023
RMTPP	[119]	GitHub	2016

TABLE X: Behavioral datasets for dynamic graph representation learning.

	Dataset	# Nodes	# Edges	# Snapshots / Unique Steps	Domain	Time Granularity
DTDG	BSI-ZK	1,744,561	56,194,191	257	Finance	Days
	BSI-SVT	89,564	190,133	49	Finance	Weeks
	Bitcoin-OTC	5,881	35,592	279	Finance	Weeks
	Bitcoin-Alpha	3,783	24,186	274	Finance	Weeks
	Reddit-Title	54,075	571,927	178	Interaction	Weeks
	Reddit-Body	35,776	286,561	178	Interaction	Weeks
	AS-733	7,716	11,965,533	733	Traffic	Days
	UCI-Message	1,899	59,835	49	Social	Weeks
	Flights	13,169	1,927,145	122	Transport	Days
	Can. Parl.	734	74,478	14	Politics	Years
	US Legis.	225	60,396	12	Politics	Congresses
	UN Trade	255	507,497	32	Finance	
	UN Vote	201	1,035,742	72	Politics	
	Contact	692	2,426,279	8,064	Proximity	5 Minutes
CTDG	Wikipedia	9,227	157,474	152,757	Social	Unix timestamps
	Reddit	10,984	672,447	669,065	Social	Unix timestamps
	MOOC	7,144	411,749	345,600	Interaction	Unix timestamps
	LastFM	1,980	1,293,103	1,283,614	Interaction	Unix timestamps
	Enron	184	125,235	22,632	Social	Unix timestamps
	Social Evo.	74	2,099,519	565,932	Proximity	Unix timestamps
	ML25M	221,588	25,000,095	20,115,267	Interactions	Unix timestamps
	DGraphFin	4,889,537	4,300,999	821	Finance	Unix timestamps
	Taobao	5,149,747	100,135,088	815,859	E-commerce	Unix timestamps
	tgb1-review-v2	352,637	4,873,540	6,865	E-commerce	Unix timestamps
	tgb1-coin	638,486	22,809,486	1,295,720	Finance	Unix timestamps
	tgb1-comment	994,790	44,314,507	30,998,030	Interaction	Unix timestamps
	tgb1-flight	18,143	67,169,570	1,385	Transport	Unix timestamps

C. Dynamic Graphs

a) Datasets

The datasets used in dynamic graph representation learning span a wide range of domains, including finance, social media, and e-commerce, providing researchers with diverse resources to study behavioral data across different application scenarios. Based on the intrinsic characteristics of the datasets, we categorize them into DTDG and CTDG datasets. As discussed in Sec. III-C2, CTDG representation learning models can also handle DTDG data. For example, in DyGFormer [267], the authors evaluate their model using several DTDG datasets, such as UCI-Message, Flights, and Can.Parl.

Commonly used DTDG datasets include BSI-ZK, BSI-SVT, Bitcoin-OTC, Bitcoin-Alpha, Reddit-Title, Reddit-Body, and AS-733 [198]. For CTDG datasets, Wikipedia, Reddit, MOOC, and LastFM are among the most widely adopted. These four datasets are originally introduced in the JODIE paper [17] and have since been extensively used in subsequent research. The authors of JODIE continue to maintain and update the Stanford Network Analysis Project (SNAP)², which hosts many other dynamic graph datasets.

²<https://snap.stanford.edu/index.html>

Since the publication of DyGFormer, broader evaluations on CTDG representation learning models have emerged. For instance, in addition to several DTDG datasets, DyGFormer also uses Enron and Social Evo. datasets for evaluation. Large-scale datasets such as DGraphFin [499], ML25M [500], and Taobao [501] are often used to benchmark scalable training frameworks. Recently, the release of the Temporal Graph Benchmark (TGB) [502], [503] has introduced a series of larger-scale datasets, including tgb1-review-v2, tgb1-coin, tgb1-comment and tgb1-flight, which have further enriched the evaluation landscape. We summarize the commonly used dynamic graph datasets in Tab. X.

b) Benchmarks

The early stage of dynamic graph benchmark development, exemplified by PyTorch Geometric Temporal (PyG-Temporal) [504], focuses primarily on DTDGs. This early framework provides essential support for snapshot-based modeling but remains limited in scalability and task diversity.

With the increasing interest in modeling fine-grained temporal dynamics, the next phase sees the emergence of CTDG benchmarks. TGL [288] and SPEED [290] emphasize large-scale learning and efficient training on billion-edge graphs, marking a shift toward practical deployment scenarios and

TABLE XI: Benchmarks for dynamic graph representation learning.

Benchmark	Paper	Repository	Specialize	Year
PyG-Temporal	[504]	GitHub	DTDG	2021
TGL	[288]	GitHub	Large-scale CTDG	2022
SPEED	[290]	GitHub	Large-scale CTDG	2023
DYGL	[508]	GitHub	DTDG and CTDG	2023
DyGLib	[267]	GitHub	CTDG	2024
TGB	[502], [503]	GitHub	CTDG	2024
BenchTemp	[505]	GitHub	CTDG	2024
DGB	[509]	GitHub	DTDG and CTDG	2024
BenchTGNN	[506]	GitHub	CTDG	2024
TGX	[507]	GitHub	CTDG	2024
DGNN	[213]	GitHub	DTDG and CTDG	2024
UTG	[283]	GitHub	DTDG and CTDG	2024

TABLE XII: Datasets for textual behavioral data.

Datasets	# Entity	# Behavior	# Behavior Category	# Timestamp	Domain
Enron	42,711	797,907	10	1,006	E-mail
GDELT	6,786	1,339,245	237	2,591	Knowledge graph
ICEWS1819	31,796	1,100,071	266	730	Knowledge graph
Stack elec	397,702	1,262,225	2	5,224	Multi-round dialogue
Stack ubuntu	674,248	1,497,006	2	4,972	Multi-round dialogue
Googlenet_CCT	111,168	1,380,623	5	55,521	E-commerce
Amazon movies	293,566	3,217,324	5	7,287	E-commerce
Yelp	2,138,242	6,990,189	5	6,036	E-commerce

industrial applicability. Recent years have witnessed rapid expansion of the ecosystem. The Dynamic Graph Library (DyGLib) [267], released alongside DyGFormer, as well as the Temporal Graph Benchmark (TGB) [502] and its extended version TGB 2.0 [503], set new standards by offering large-scale, domain-diverse CTDG datasets alongside automated evaluation pipelines and leaderboards, significantly improving reproducibility and comparability. Parallel efforts—such as BenchTemp [505], BenchTGNN [506], Temporal Graph Analysis with TGX (TGX) [507], DGNN [213] have further enriched the landscape with modular toolkits that cover an even wider spectrum of models.

In addition to CTDG-focused efforts, several benchmark libraries aim to bridge the DTDG and CTDG paradigms. DYGL-library (DYGL) [508] and the Dynamic Graph Benchmark (DGB) [509] support both DTDG and CTDG models and tasks under a unified framework. Similarly, Unifying Temporal Graph (UTG) [283] attempts to provide a general infrastructure that accommodates both DTDG and CTDG representation learning methods, further promoting methodological consistency across different temporal settings. We summarize representative benchmarks in Tab. XI.

D. Textual Data

a) Datasets

Finally, we introduce the *language-based behavioral* datasets. These datasets typically consist of entities and the behavioral interactions occurring among them. Such interactions are often timestamped, and both entities and interactions are associated with rich textual attributes. We summarize these datasets in Tab. XII, which collectively span a broad spectrum of domains, ranging from *E-mail communication* and *Knowledge graphs* to *Multi-round dialogues* and large-scale *E-commerce platforms*. For example, the Enron dataset models

organizational email exchanges: entities correspond to individual users annotated with descriptive metadata such as email addresses, while interactions represent message transmissions that are further accompanied by the full textual content of the emails. These datasets not only mirror real-world communication and decision-making processes, but also provide fertile ground for advancing research at the nexus of natural language processing and behavioral modeling. They offer a robust empirical foundation for harnessing LLMs to capture fine-grained behavioral regularities and to drive performance gains across a diverse set of downstream applications.

b) Benchmarks

The benchmarking practice of applying LLMs to behavioral data is still in its early stages. For instance, DTGB [510] has recently introduced a benchmark that models user behaviors through dynamic text-attributed graphs, thereby providing one of the first systematic attempts to bridge behavioral dynamics with textual semantics in an LLM-friendly manner. Nevertheless, beyond this effort, we find relatively few benchmarks or frameworks that explicitly focus on modeling language-based behavioral data with LLMs. One possible reason is the inherent complexity of such data: interactions are not only temporally ordered but also linguistically rich, which poses challenges in aligning heterogeneous modalities, designing scalable representations, and ensuring efficient training. Moreover, the absence of standardized datasets and evaluation protocols further hinders the community from conducting consistent and reproducible studies. This gap highlights a promising research direction for future work. Establishing comprehensive benchmarks, unified evaluation metrics, and diverse datasets would significantly accelerate progress, allowing LLMs to better capture the interplay between textual semantics and behavioral regularities.

VII. CONCLUSION AND FUTURE DIRECTIONS

In conclusion, behavioral data representation learning plays a critical role in extracting meaningful patterns from complex, dynamic behaviors, supporting diverse real-world applications. This survey systematically categorizes existing methods, clarifies their functionalities and application scenarios, and provides an insightful framework to guide future developments.

Looking forward, several critical areas demand further research: (1) development of integrated multi-modal models for richer and more comprehensive behavioral representations; (2) scaling representation learning techniques to handle real-time data streams and large-scale datasets; (3) enhancing interpretability and explainability to facilitate user trust and regulatory compliance; (4) leveraging the potential of foundational models such as large language models and pre-trained graph neural networks for advanced transfer learning and domain adaptation; and (5) ensuring fairness, transparency, and ethical considerations in behavioral modeling practices. Addressing these areas will substantially advance the practical utility and theoretical robustness of behavioral data representation methodologies.

REFERENCES

- [1] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.
- [3] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.
- [4] A. Graves, “Long short-term memory,” in *Supervised Sequence Labelling with Recurrent Neural Networks*, 2012, vol. 385, pp. 37–45.
- [5] E. Rossi, B. P. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. M. Bronstein, “Temporal graph networks for deep learning on dynamic graphs,” *arXiv preprint arXiv:2006.10637*, 2020.
- [6] OpenAI, “GPT-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [7] A. H. Huang, H. Wang, and Y. Yang, “Finbert: A large language model for extracting information from financial text,” *Contemporary Accounting Research*, vol. 40, no. 2, pp. 806–841, 2023.
- [8] A. Gomez-Marin, J. J. Paton, A. R. Kampff, R. M. Costa, and Z. F. Mainen, “Big behavioral data: psychology, ethology and the foundations of neuroscience,” *Nature Neuroscience*, vol. 17, no. 11, pp. 1455–1462, 2014.
- [9] L. von Ziegler, O. Sturman, and J. Bohacek, “Big behavior: Challenges and opportunities in a new era of deep behavior profiling,” *Neuropsychopharmacology*, vol. 46, no. 1, pp. 33–44, 2021.
- [10] R. C. Wilson and A. G. E. Collins, “Ten simple rules for the computational modeling of behavioral data,” *eLife*, vol. 8, p. e49547, 2019.
- [11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [12] Y. Bengio, “Deep learning of representations for unsupervised and transfer learning,” in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, vol. 27, 2012, pp. 17–36.
- [13] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [14] Y. Bengio, Y. LeCun, and G. Hinton, “Deep learning for AI,” *Communications of the ACM*, vol. 64, no. 7, pp. 58–65, 2021.
- [15] S. R. Eddy, “Hidden Markov models,” *Current Opinion in Structural Biology*, vol. 6, no. 3, pp. 361–365, 1996.
- [16] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, “BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1441–1450.
- [17] S. Kumar, X. Zhang, and J. Leskovec, “Predicting dynamic embedding trajectory in temporal interaction networks,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 1269–1278.
- [18] S. Feuerriegel, A. Maarouf, D. Bär, D. Geissler, J. Schweisthal, N. Pröllochs, C. E. Robertson, S. Rathje, J. Hartmann, S. M. Mohammad, O. Netzer, A. A. Siegel, B. Plank, and J. J. Van Bavel, “Using natural language processing to analyse text data in behavioural science,” *Nature Reviews Psychology*, vol. 4, no. 2, pp. 96–111, 2025.
- [19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI Blog*, 2019, accessed January 4, 2026.
- [20] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.
- [21] L. Xiong, H. Wang, X. Chen, L. Sheng, Y. Xiong, J. Liu, Y. Xiao, H. Chen, Q.-L. Han, and Y. Tang, “Deepseek: Paradigm shifts and technical evolution in large AI models,” *IEEE/CAA Journal of Automatica Sinica*, vol. 12, no. 5, pp. 841–858, 2025.
- [22] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
- [23] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “LightGBM: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 3146–3154.
- [24] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “CatBoost: unbiased boosting with categorical features,” in *Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 6639–6649.
- [25] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, “Wide & deep learning for recommender systems,” in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 7–10.
- [26] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, “DeepFM: A factorization-machine based neural network for CTR prediction,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 1725–1731.
- [27] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, “xDeepFM: Combining explicit and implicit feature interactions for recommender systems,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1754–1763.
- [28] G. Ke, J. Zhang, Z. Xu, J. Bian, and T.-Y. Liu, “TabNN: A universal neural network solution for tabular data,” *ICLR 2019 Conference Blind Submission, OpenReview*, 2018.
- [29] I. Shavitt and E. Segal, “Regularization learning networks: Deep learning for tabular datasets,” in *Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 1386–1396.
- [30] S. Popov, S. Morozov, and A. Babenko, “Neural oblivious decision ensembles for deep learning on tabular data,” in *International Conference on Learning Representations*, 2020.
- [31] B. Sun, L. Yang, W. Zhang, M. Lin, P. Dong, C. Young, and J. Dong, “Supertml: Two-dimensional word embedding for the precognition on structured tabular data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 2973–2981.
- [32] S. Ö. Arik and T. Pfister, “TabNet: Attentive interpretable tabular learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6679–6687.
- [33] G. Ke, Z. Xu, J. Zhang, J. Bian, and T.-Y. Liu, “Deep-GBM: A deep learning framework distilled by GBDT for online prediction tasks,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 384–394.
- [34] Y. Luo, H. Zhou, W.-W. Tu, Y. Chen, W. Dai, and Q. Yang, “Network on network for tabular data classification in real-world applications,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2317–2326.

- [35] A. Abutbul, G. Elidan, L. Katzir, and R. El-Yaniv, “DNF-net: A neural architecture for tabular data,” *arXiv preprint arXiv:2006.06465*, 2020.
- [36] J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar, “VIME: Extending the success of self- and semi-supervised learning to tabular domain,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 11 033–11 043.
- [37] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, “Tabtransformer: Tabular data modeling using contextual embeddings,” *arXiv preprint arXiv:2012.06678*, 2020.
- [38] S. Cai, K. Zheng, G. Chen, H. V. Jagadish, B. C. Ooi, and M. Zhang, “ARM-net: Adaptive relation modeling network for structured data,” in *Proceedings of “ARM-Net: Adaptive Relation Modeling Network for Structured Data” - dblp[https://dblp.org/rec/conf/sigmod/CaiZ0JOZ21] the 2021 International Conference on Management of Data*, 2021, pp. 207–220.
- [39] J. Kossen, N. Band, C. Lyle, A. N. Gomez, T. Rainforth, and Y. Gal, “Self-attention between datapoints: Going beyond individual input-output pairs in deep learning,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 28 742–28 756.
- [40] A. Kadra, M. Lindauer, F. Hutter, and J. Grabocka, “Well-tuned simple nets excel on tabular datasets,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 23 928–23 941.
- [41] S. Ivanov and L. Prokhorenkova, “Boost then convolve: Gradient boosting meets graph neural networks,” in *Proceedings of the 9th International Conference on Learning Representations*, 2021.
- [42] Z. Liu, Q. Liu, H. Zhang, and Y. Chen, “DNN2LR: Interpretation-inspired feature crossing for real-world tabular data,” *arXiv preprint arXiv:2008.09775*, 2020.
- [43] Y. Zhu, T. S. Brettin, F. Xia, A. Partin, M. Shukla, H. Yoo, Y. A. Evrard, J. H. Doroshow, and R. L. Stevens, “Converting tabular data into images for deep learning with convolutional neural networks,” *Scientific Reports*, vol. 11, no. 1, p. 11325, 2021.
- [44] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, “Revisiting deep learning models for tabular data,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 18 932–18 943.
- [45] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein, “SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training,” *arXiv preprint arXiv:2106.01342*, 2021.
- [46] D. Bahri, H. Jiang, Y. Tay, and D. Metzler, “SCARF: Self-supervised contrastive learning using random feature corruption,” in *Proceedings of the Tenth International Conference on Learning Representations*, 2022, pp. 1–14.
- [47] M. Joseph and H. Raj, “GANDALF: Gated adaptive network for deep automated learning of features,” *arXiv preprint arXiv:2207.08548*, 2022.
- [48] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko, “TabDDPM: Modelling tabular data with diffusion models,” in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, 2023, pp. 17 564–17 579.
- [49] G. Liu, J. Yang, and L. Wu, “Ptab: Using the pre-trained language model for modeling tabular data,” *arXiv preprint arXiv:2209.08060*, 2022.
- [50] M. E. Zarlenga, Z. Shams, M. E. Nelson, B. Kim, and M. Jamnik, “Tabcbm: Concept-based interpretable neural networks for tabular data,” *Transactions on Machine Learning Research*, 2023.
- [51] K.-Y. Chen, P.-H. Chiang, H.-R. Chou, T.-W. Chen, and T.-H. Chang, “Trompt: Towards a better deep neural network for tabular data,” in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, 2023, pp. 4392–4434.
- [52] P. Chen, S. Sarkar, L. Lausen, B. Srinivasan, S. Zha, R. Huang, and G. Karypis, “HyTreI: Hypergraph-enhanced tabular data representation learning,” in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 32 173–32 193.
- [53] S. Chen, J. Wu, N. Hovakimyan, and H. Yao, “ReConTab: Regularized contrastive representation learning for tabular data,” in *Table Representation Learning Workshop, Advances in Neural Information Processing Systems 2023*, 2023.
- [54] B. Zhu, X. Shi, N. Erickson, M. Li, G. Karypis, and M. Shoaran, “Xtab: Cross-table pretraining for tabular transformers,” in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, 2023, pp. 43 181–43 204.
- [55] A. F. Thielmann, A. Reuter, T. Kneib, D. Rügamer, and B. Säfken, “Interpretable additive tabular transformer networks,” *Transactions on Machine Learning Research*, 2024.
- [56] M. A. Ahamed and Q. S. Cheng, “MambaTab: A plug-and-play model for learning tabular data,” in *Proceedings of the 2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval*, 2024, pp. 369–375.
- [57] A. Margelou, N. Simidjievski, P. Lio, and M. Jamnik, “Gcondnet: A novel method for improving neural networks on small high-dimensional tabular data,” *Transactions on Machine Learning Research*, 2024.
- [58] C. Xu, Y.-C. Huang, J. Y.-C. Hu, W. Li, A. Gilani, H.-S. Goan, and H. Liu, “Bishop: Bi-directional cellular learning for tabular data with generalized sparse modern hopfield model,” in *Proceedings of the 41st International Conference on Machine Learning*, vol. 235, 2024, pp. 55 048–55 075.
- [59] K. Na, J.-H. Lee, and E. Kim, “LF-transformer: Latent factorizer transformer for tabular learning,” *IEEE Access*, vol. 12, pp. 10 690–10 698, 2024.
- [60] Y. Gorishniy, I. Rubachev, N. Kartashev, D. Shlenskii, A. Kotelnikov, and A. Babenko, “TabR: Tabular deep learning meets nearest neighbors,” in *The Twelfth International Conference on Learning Representations*, 2024, pp. 1–14.
- [61] J. Yan, B. Zheng, H. Xu, Y. Zhu, D. Z. Chen, J. Sun, J. Wu, and J. Chen, “Making pre-trained language models great on tabular prediction,” in *International Conference on Learning Representations*, 2024, pp. 1–14.
- [62] M. J. Kim, L. Grinsztajn, and G. Varoquaux, “CARTE: Pretraining and transfer for tabular learning,” in *Proceedings of the 41st International Conference on Machine Learning*, vol. 235, 2024, pp. 23 843–23 866.
- [63] J. Wu, S. Chen, Q. Zhao, R. Sergazinov, C. Li, S. Liu, C. Zhao, T. Xie, H. Guo, C. Ji, D. Cociorva, and H. Brunzell, “SwitchTab: Switched autoencoders are effective tabular learners,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 14, 2024, pp. 15 924–15 933.
- [64] T. Afonja, H.-P. Wang, R. Kerouche, and M. Fritz, “Dp-2stage: Adapting language models as differentially private tabular data generators,” *Transactions on Machine Learning Research*, 2025.
- [65] M. J. Kim, F. Lefebvre, G. Brison, A. Perez-Lebel, and G. Varoquaux, “Table foundation models: on knowledge pre-training for tabular learning,” *Transactions on Machine Learning Research*, 2025.
- [66] I. Kang, P. Ram, Y. Zhou, H. Samulowitz, and O. Seneviratne, “On learning representations for tabular data distillation,” *Transactions on Machine Learning Research*, 2025.
- [67] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. Eisenschlos, “TaPas: Weakly supervised table parsing via pre-training,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4320–4333.
- [68] Q. Liu, B. Chen, J. Guo, M. Ziyadi, Z. Lin, W. Chen, and J.-G. Lou, “[TAPEX: Table [find results for ‘download bibtex’](https://www.microsoft.com/en-us/research/publication/tapex-table-pre-training-via-learning-a-neural-sql-executor/)]pre-training via learning a neural SQL executor,” in *International Conference on Learning Representations*, 2022.
- [69] S. Hegselmann, A. Buendia, H. Lang, M. Agrawal, X. Jiang, and D. Sontag, “TabLLM: Few-shot classification of tabular data with large language models,” in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, vol. 206, 2023, pp. 5549–5581.
- [70] A. S. Sundar and L. Heck, “cTBLS: Augmenting large language models with conversational tables,” in *Proceedings of the 5th Workshop on NLP for Conversational AI*, 2023, pp. 59–70.
- [71] S. Han, S. Lee, M. Cha, S. O. Arik, and J. Yoon, “Llm-guided self-supervised tabular learning with task-specific pre-text tasks,” *Transactions on Machine Learning Research*, 2025.
- [72] S. Cromp, S. S. S. N. GNVV, M. Alkhudhayri, C. Cao, S. Guo, N. Roberts, and F. Sala, “Tabby: A language model architecture for tabular and structured data synthesis,” *Transactions on Machine Learning Research*, 2026.
- [73] R. Schwartz-Ziv and A. Armon, “Tabular data: Deep learning is not all you need,” *Information Fusion*, vol. 81, pp. 84–90, 2022.
- [74] S. Somvanshi, S. Das, S. A. Javed, G. Antarksa, and A. Hossain, “A survey on deep tabular learning,” *arXiv preprint arXiv:2410.12034*, 2024.
- [75] T. Ucar, E. Hajiramezanali, and L. Edwards, “Subtab: Subsetting features of tabular data for self-supervised representation learning,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 18 853–18 865.
- [76] Y. Zhu, T. Brettin, F. Xia, A. Partin, M. Shukla, H. Yoo, Y. A. Evrard, J. H. Doroshow, and R. L. Stevens, “Converting tabular data into images for deep learning with convolutional neural networks,” *Scientific Reports*, vol. 11, no. 1, p. 11325, 2021.
- [77] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark,

- "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, p. 160035, 2016.
- [78] D. Ulmer, L. Meijerink, and G. Cinà, "Trust issues: Uncertainty estimation does not enable reliable OOD detection on medical tabular data," in *Proceedings of the Machine Learning for Health NeurIPS Workshop*, vol. 136, 2020, pp. 341–354.
- [79] A. Kumar, I. Garg, and S. Kaur, "Loan approval prediction based on machine learning approach," *IOSR Journal of Computer Engineering*, vol. 18, no. 3, pp. 79–81, 2016.
- [80] J. M. Clements, D. Xu, N. Yousefi, and D. Efimov, "Sequential deep learning for credit risk monitoring with tabular financial data," *arXiv preprint arXiv:2012.15330*, 2020.
- [81] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 7–10.
- [82] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 1725–1731.
- [83] S. Popov, S. Morozov, and A. Babenko, "Neural oblivious decision ensembles for deep learning on tabular data," in *International Conference on Learning Representations*, 2020.
- [84] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [85] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 507–520.
- [86] V. Borisov, K. Seßler, T. Leemann, M. Pawelczyk, and G. Kasneci, "Language models are realistic tabular data generators," in *International Conference on Learning Representations*, 2023.
- [87] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 18932–18943.
- [88] C. McCarter, "Unmasking trees for tabular data," *Transactions on Machine Learning Research*, 2025.
- [89] A. Larionov, N. M. Adams, and K. N. Webster, "Investigating the impact of missing value handling on boosted trees and deep learning for tabular data: A claim-reserving case study," *Transactions on Machine Learning Research*, 2025.
- [90] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.
- [91] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *Proceedings of the Conference on Language Modeling*, 2024.
- [92] H. Iida, D. Thai, V. Manjunatha, and M. Iyyer, "TABBIE: Pretrained representations of tabular data," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 3446–3456.
- [93] Y. Fu, H. Peng, and T. Khot, "How does GPT obtain its ability? tracing emergent abilities of language models to their sources," 2022, accessed: 2026-01-04.
- [94] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022.
- [95] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 39:1–39:45, 2024.
- [96] B. Zhao, C. Ji, Y. Zhang, W. He, Y. Wang, Q. Wang, R. Feng, and X. Zhang, "Large language models are complex table parsers," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 14786–14802.
- [97] Y. Sui, M. Zhou, S. Han, and D. Zhang, "Table meets llm: Can large language models understand structured table data? a benchmark and empirical study," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 645–654.
- [98] P. Kenfack, S. E. Kahou, and U. Aïvodji, "Towards fair in-context learning with tabular foundation models," *Transactions on Machine Learning Research*, 2026.
- [99] A. Singha, J. Cambronero, S. Gulwani, V. Le, and C. Parnin, "Tabular representation, noisy operators, and impacts on table structure understanding tasks in LLMs," *arXiv preprint arXiv:2310.10358*, 2023.
- [100] A. Narayan, I. Chami, L. J. Orr, S. Arora, and C. Ré, "Can foundation models wrangle your data?" *arXiv preprint arXiv:2205.09911*, 2022.
- [101] B. Yu, C. Fu, H. Yu, F. Huang, and Y. Li, "Unified language representation for question answering over text, tables, and images," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 4756–4765.
- [102] K. R. Bhandari, S. Xing, S. Dan, and J. Gao, "Exploring the robustness of language models for tabular question answering via attention analysis," *Transactions on Machine Learning Research*, 2025.
- [103] Y. Sui, M. Zhou, M. Zhou, S. Han, and D. Zhang, "Table meets LLM: Can large language models understand structured table data? a benchmark and empirical study," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 645–654.
- [104] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.
- [105] Y. Sui, J. Zou, M. Zhou, X. He, L. Du, S. Han, and D. Zhang, "TAP4LLM: Table provider on sampling, augmenting, and packing semi-structured data for large language model reasoning," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 10306–10323.
- [106] R. Aly, Z. Guo, M. S. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, and A. Mittal, "FEVEROUS: Fact extraction and verification over unstructured and structured information," in *Advances in Neural Information Processing Systems*, 2021.
- [107] M. Ggalwango, H. Nakayiza, D. Jjingo, and J. Nakatumba-Nabende, "Prompt engineering in large language models," in *Proceedings of the International Conference on Data Intelligence and Cognitive Informatics*, 2024, pp. 387–402.
- [108] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24824–24837.
- [109] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," *Transactions on Machine Learning Research*, 2023.
- [110] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *International Conference on Learning Representations*, 2023.
- [111] H. Zhang, P. Chang, and Z. Ji, "Bridging the gap: Deciphering tabular data using large language model," *arXiv preprint arXiv:2308.11891*, 2023.
- [112] S.-C. Liu, S. Wang, T. Chang, W. Lin, C.-W. Hsiung, Y.-C. Hsieh, Y.-P. Cheng, S.-H. Luo, and J. Zhang, "Jarvix: A LLM no code platform for tabular data analysis and optimization," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2023, pp. 622–630.
- [113] W. Zhang, Y. Wang, Y. Song, V. J. Wei, Y. Tian, Y. Qi, J. H. Chan, R. C.-W. Wong, and H. Yang, "Natural language interfaces for tabular data querying and visualization: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6699–6718, 2024.
- [114] X. Fang, W. Xu, F. A. Tan, Z. Hu, J. Zhang, Y. Qi, S. H. Sengamedu, and C. Faloutsos, "Large language models (llms) on tabular data: Prediction, generation, and understanding - a survey," *Transactions on Machine Learning Research*, 2024.
- [115] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proceedings of the 19th International World Wide Web Conference*, 2010, pp. 811–820.
- [116] G.-E. Yap, X.-L. Li, and P. S. Yu, "Effective next-items recommendation via personalized sequential pattern mining," in *Database Systems for Advanced Applications: 17th International Conference, Busan, South Korea, April 15–18, 2012, Proceedings, Part II*, vol. 7239, 2012, pp. 48–64.
- [117] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI recommendation," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 2069–2075.

- [118] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, “Session-based recommendations with recurrent neural networks,” in *Proceedings of the 4th International Conference on Learning Representations*, 2016.
- [119] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, “Recurrent marked temporal point processes: Embedding event history to vector,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1555–1564.
- [120] H. Mei and J. M. Eisner, “The neural hawkes process: A neurally self-modulating multivariate point process,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 6754–6764.
- [121] S. Hong, M. Wu, H. Li, and Z. Wu, “Event2vec: Learning representations of events on temporal sequences,” in *Web and Big Data: First International Joint Conference, APWeb-WAIM 2017, Beijing, China, July 7–9, 2017, Proceedings, Part II*, vol. 10367, 2017, pp. 33–47.
- [122] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, “IRGAN: A minimax game for unifying generative and discriminative information retrieval models,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 515–524.
- [123] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, “Personalizing session-based recommendations with hierarchical recurrent neural networks,” in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 130–137.
- [124] J. Tang and K. Wang, “Personalized top-N sequential recommendation via convolutional sequence embedding,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 565–573.
- [125] S. Zhang, Y. Tay, L. Yao, and A. Sun, “Next item recommendation with self-attention,” *arXiv preprint arXiv:1808.06414*, 2018.
- [126] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, “Sequential recommendation with user memory networks,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 108–116.
- [127] H. Bharadhwaj, H. Park, and B. Y. Lim, “RecGAN: Recurrent generative adversarial networks for recommendation systems,” in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 372–376.
- [128] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, “Learning time series associated event sequences with recurrent point process networks,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 10, pp. 3124–3136, 2019.
- [129] F. Zhu, C. Chen, Y. Wang, G. Liu, and X. Zheng, “DTCDR: A framework for dual-target cross-domain recommendation,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1533–1542.
- [130] T. Zhang, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, D. Wang, G. Liu, and X. Zhou, “Feature-level deeper self-attention network for sequential recommendation,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, pp. 4320–4326.
- [131] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, “A simple convolutional generative network for next item recommendation,” in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 582–590.
- [132] Q. Zhang, A. Lipani, O. Kirnap, and E. Yilmaz, “Self-attentive Hawkes process,” in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 2020, pp. 11 183–11 193.
- [133] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha, “Transformer hawkes process,” in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 2020, pp. 11 692–11 702.
- [134] Y. Li, S. Rao, J. R. A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, and G. Salimi-Khorshidi, “BEHRT: Transformer for electronic health records,” *Scientific Reports*, vol. 10, no. 1, p. 7155, 2020.
- [135] J. Li, Y. Wang, and J. J. McAuley, “Time interval aware self-attention for sequential recommendation,” in *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining*, 2020, pp. 322–330.
- [136] H. Ren, J. Wang, W. X. Zhao, and N. Wu, “RAPT: Pre-training of time-aware transformer for learning robust healthcare representation,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3503–3511.
- [137] Z. Liu, Y. Chen, J. Li, P. S. Yu, J. McAuley, and C. Xiong, “Contrastive self-supervised sequential recommendation with robust augmentation,” *arXiv preprint arXiv:2108.06479*, 2021.
- [138] Z. Liu, Z. Fan, Y. Wang, and P. S. Yu, “Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 1608–1612.
- [139] Y. Hou, S. Mu, W. X. Zhao, Y. Li, B. Ding, and J.-R. Wen, “Towards universal sequence representation learning for recommender systems,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 585–593.
- [140] C. Li, M. Zhao, H. Zhang, C. Yu, L. Cheng, G. Shu, B. Kong, and D. Niu, “RecGURU: Adversarial learning of generalized user representations for cross-domain recommendation,” in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 571–581.
- [141] S. Xue, Y. Wang, Z. Chu, X. Shi, C. Jiang, H. Hao, G. Jiang, X. Feng, J. Y. Zhang, and J. Zhou, “Prompt-augmented temporal point process for streaming event sequence,” in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 18 885–18 905.
- [142] W. Bae, M. O. Ahmed, F. Tung, and G. L. Oliveira, “Meta temporal point processes,” in *International Conference on Learning Representations*, 2023, pp. 1–14.
- [143] S. Hu, Z. Zhang, B. Luo, S. Lu, B. He, and L. Liu, “BERT4ETH: A pre-trained transformer for ethereum fraud detection,” in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2189–2197.
- [144] R. R. Chowdhury, J. Li, X. Zhang, D. Hong, R. K. Gupta, and J. Shang, “PrimeNet: Pre-training for irregular multivariate time series,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7184–7192.
- [145] S. Ni, W. Zhou, J. Wen, L. Hu, and S. Qiao, “Enhancing sequential recommendation with contrastive generative adversarial network,” *Information Processing & Management*, vol. 60, no. 3, p. 10331, 2023.
- [146] F. M. Abushaqra, H. Xue, Y. Ren, and F. D. Salim, “Seqlink: A robust neural-ode architecture for modelling partially observed time series,” *Transactions on Machine Learning Research*, 2024.
- [147] A. F. Ansari, L. Stella, A. C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, J. Zschieger, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and B. Wang, “Chronos: Learning the language of time series,” *Transactions on Machine Learning Research*, 2024.
- [148] T. Wang, M. Honari-Jahromi, S. Katsarou, O. Mikheeva, T. Panagiotakopoulos, S. Asadi, and O. Smirnov, “player2vec: A language modeling approach to understand player behavior in games,” *arXiv preprint arXiv:2404.04234*, 2024.
- [149] S. J. Talukder, Y. Yue, and G. Gkioxari, “Totem: Tokenized time series embeddings for general time series analysis,” *Transactions on Machine Learning Research*, 2024.
- [150] P. Mohapatra, L. Wang, and Q. Zhu, “Phase-driven generalizable representation learning for nonstationary time series classification,” *Transactions on Machine Learning Research*, 2025.
- [151] R. Yuan and G. Fang, “Residual TPP: A unified lightweight approach for event stream data analysis,” in *Proceedings of the 42nd International Conference on Machine Learning*, vol. 267, 2025, pp. 73 455–73 477.
- [152] W. Wang, J. Ma, Y. Zhang, K. Zhang, J. Jiang, Y. Yang, Y. Zhou, and Z. Zhang, “Intent oriented contrastive learning for sequential recommendation,” in *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, 2025, pp. 12 748–12 756.
- [153] S. Hu, W. Wu, Z. Tang, Z. Huan, L. Wang, X. Zhang, J. Zhou, L. Zou, and C. Li, “HORAE: Temporal multi-interest pre-training for sequential recommendation,” *ACM Transactions on Information Systems*, vol. 43, no. 4, pp. 88:1–88:29, 2025.
- [154] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: MIT Press, 1998.
- [155] S. Wang and L. Cao, “Inferring implicit rules by learning explicit and hidden item dependency,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 3, pp. 935–946, 2020.
- [156] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. A. Orgun, “Sequential recommender systems: Challenges, progress and prospects,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, pp. 6332–6338.
- [157] C. Cheng, H. Yang, M. R. Lyu, and I. King, “Where you like to go next: Successive point-of-interest recommendation,” in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2013, pp. 2605–2611.
- [158] J.-D. Zhang, C.-Y. Chow, and Y. Li, “LORE: Exploiting sequential influence for location recommendations,” in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014, pp. 103–112.

- [159] J. L. Moore, S. Chen, D. R. Turnbull, and T. Joachims, "Taste over time: The temporal dynamics of user preferences," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 2013, pp. 401–406.
- [160] K. W. Church, "Word2vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.
- [161] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based object classification for neuromorphic vision sensing," in *2019 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019.
- [162] Z. Wu, X. Wang, H. Chang, H. Chen, L. Sun, and W. Zhu, "Aligning large multimodal model with sequential recommendation via content-behavior guidance," in *Proceedings of the 2025 International Conference on Multimedia Retrieval*, 2025, pp. 1507–1516.
- [163] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734.
- [164] L. Butera, G. D. Felice, A. Cini, and C. Alippi, "On the regularization of learnable embeddings for time series forecasting," *Transactions on Machine Learning Research*, 2025.
- [165] P. Skalski, D. Sutton, S. Burrell, I. Perez, and J. Wong, "Towards a foundation purchasing model: Pretrained generative autoregression on transaction sequences," in *Proceedings of the 4th ACM International Conference on AI in Finance*, 2023, pp. 141–149.
- [166] Z. Wang, Q. Wu, B. Zheng, J. Wang, K. Huang, and Y. Shi, "Sequence as genes: An user behavior modeling framework for fraud transaction detection in e-commerce," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 5194–5203.
- [167] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 2020, pp. 1893–1902.
- [168] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," in *Proceedings of the 2022 IEEE 38th International Conference on Data Engineering*, 2022, pp. 1259–1273.
- [169] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 813–823.
- [170] X. Chen, Y. Xiong, S. Zhang, J. Zhang, Y. Zhang, S. Zhou, X. Wu, M. Zhang, T. Liu, and W. Wang, "DTFormer: A transformer-based method for discrete-time dynamic graph representation learning," *arXiv preprint arXiv:2407.18523*, 2024.
- [171] T. Postuvan, C. Grohfeldt, M. Russo, and G. Lovisotto, "Learning-based link anomaly detection in continuous-time dynamic graphs," *Transactions on Machine Learning Research*, 2024.
- [172] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 554–560.
- [173] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, "On evolutionary spectral clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 4, pp. 1–30, 2009.
- [174] M.-S. Kim and J. Han, "A particle-and-density based evolutionary clustering method for dynamic networks," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 622–633, 2009.
- [175] M. Gupta, C. C. Aggarwal, J. Han, and Y. Sun, "Evolutionary clustering and analysis of bibliographic networks," in *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, 2011, pp. 63–70.
- [176] L. Yao, L. Wang, L. Pan, and K. Yao, "Link prediction based on common-neighbors for dynamic social network," *Procedia Computer Science*, vol. 83, pp. 82–89, 2016.
- [177] L. Zhou, Y. Yang, X. Ren, F. Wu, and Y. Zhuang, "Dynamic network embedding by modeling triadic closure process," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
- [178] R. Hisano, "Semi-supervised graph embedding approach to dynamic link prediction," in *Complex Networks IX: Proceedings of the 9th Conference on Complex Networks*, 2018, pp. 109–121.
- [179] U. Sharan and J. Neville, "Temporal-relational classifiers for prediction in evolving domains," in *Proceedings of the Eighth IEEE International Conference on Data Mining*, 2008, pp. 540–549.
- [180] N. M. A. Ibrahim and L. Chen, "Link prediction in dynamic social networks by integrating different types of information," *Applied Intelligence*, vol. 42, no. 4, pp. 738–750, 2015.
- [181] N. M. Ahmed, L. Chen, Y. Wang, B. Li, Y. Li, and W. Liu, "Sampling-based algorithm for link prediction in temporal networks," *Information Sciences*, vol. 374, pp. 1–14, 2016.
- [182] U. Singer, I. Guy, and K. Radinsky, "Node embedding over temporal graphs," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, pp. 4605–4612.
- [183] P. Goyal, N. Kamra, X. He, and Y. Liu, "DynGEM: Deep embedding method for dynamic graphs," *arXiv preprint arXiv:1805.11273*, 2018.
- [184] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Neural Information Processing - 25th International Conference, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part I*, vol. 11301, 2018, pp. [“Structured Sequence Modeling with Graph Convolutional Recurrent ... – dblp](https://dblp.org/rec/journals/corr/SeoDVB18)362–373.
- [185] A. Narayan and P. H. O’N Roe, "Learning graph dynamics using deep neural networks," in *9th Vienna International Conference on Mathematical Modelling*, vol. 51, no. 2, 2018, pp. 433–438.
- [186] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
- [187] S. Bonner, A. Atapour-Abarghouei, P. T. Jackson, J. Brennan, I. Kureshi, G. Theodoropoulos, A. S. McGough, and B. Obara, "Temporal neighbourhood aggregation: Predicting future links in temporal graphs via recurrent variational graph convolutions," in *Proceedings of the 2019 IEEE International Conference on Big Data*, 2019, pp. 5336–5345.
- [188] E. Hajiramezanali, A. Hasanzadeh, K. Narayanan, N. Duffield, M. Zhou, and X. Qian, "Variational graph recurrent neural networks," *Advances in Neural Information Processing Systems*, vol. 32, pp. 10 700–10 710, 2019.
- [189] J. Li, Z. Han, H. Cheng, J. Su, P. Wang, J. Zhang, and L. Pan, "Predicting path failure in time-evolving graphs," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 1279–1289.
- [190] J. Chen, J. Zhang, X. Xu, C. Fu, D. Zhang, Q. Zhang, and Q. Xuan, "E-LSTM-D: A deep learning framework for dynamic network link prediction," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3699–3712, 2021.
- [191] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson, "Evolvengn: Evolving graph convolutional networks for dynamic graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5363–5370.
- [192] P. Goyal, S. R. Chhetri, and A. Canedo, "dyngraph2vec: Capturing network dynamics using dynamic graph representation learning," *Knowledge-Based Systems*, vol. 187, p. 104816, 2020.
- [193] J. Wu, M. Cao, J. C. K. Cheung, and W. L. Hamilton, "TeMP: Temporal message passing for temporal knowledge graph completion," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 5730–5746.
- [194] F. Manessi, A. Rozza, and M. Manzo, "Dynamic graph convolutional networks," *Pattern Recognition*, vol. 97, p. 107000, 2020.
- [195] F. Zhou, X. Xu, C. Li, G. Trajcevski, T. Zhong, and K. Zhang, "A heterogeneous dynamical graph neural networks approach to quantify scientific impact," *arXiv preprint arXiv:2003.12042*, 2020.
- [196] M. Yang, M. Zhou, M. Kalander, Z. Huang, and I. King, "Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2021, pp. 1975–1985.
- [197] J. Chen, X. Wang, and X. Xu, "GC-LSTM: graph convolution embedded LSTM for dynamic network link prediction," *Applied Intelligence*, vol. 52, no. 7, pp. 7513–7528, 2022.
- [198] J. You, T. Du, and J. Leskovec, "ROLAND: Graph learning framework for dynamic graphs," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2358–2366.
- [199] K. Liang, L. Meng, M. Liu, Y. Liu, W. Tu, S. Wang, S. Zhou, and X. Liu, "Learn from relational correlations and periodic events for temporal knowledge graph reasoning," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 1559–1568.

- [200] X. Qin, N. Sheikh, C. Lei, B. Reinwald, and G. Domeniconi, “Seign: A simple and efficient graph neural network for large dynamic graphs,” in *Proceedings of the 2023 IEEE 39th International Conference on Data Engineering*, 2023, pp. 2850–2863.
- [201] K. Liu, F. Zhao, G. Xu, X. Wang, and H. Jin, “RETIA: Relation-entity twin-interact aggregation for temporal knowledge graph extrapolation,” in *Proceedings of the 39th IEEE International Conference on Data Engineering*, 2023, pp. 1761–1774.
- [202] R. Jiang, Z. Wang, J. Yong, P. Jeph, Q. Chen, Y. Kobayashi, X. Song, S. Fukushima, and T. Suzumura, “Spatio-temporal meta-graph learning for traffic forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8078–8086.
- [203] A. Bastos, A. Nadgeri, K. Singh, T. Suzumura, and M. Singh, “Learnable spectral wavelets on dynamic graphs to capture global interactions,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 6779–6787.
- [204] G. Jin, L. Liu, F. Li, and J. Huang, “Spatio-temporal graph neural point process for traffic congestion event prediction,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 12, 2023, pp. 14 268–14 276.
- [205] Y. Zhu, F. Cong, D. Zhang, W. Gong, Q. Lin, W. Feng, Y. Dong, and J. Tang, “WinGNN: Dynamic graph neural networks with random gradient aggregation window,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 3650–3662.
- [206] J. Li, Z. Yu, Z. Zhu, L. Chen, Q. Yu, Z. Zheng, S. Tian, R. Wu, and C. Meng, “Scaling up dynamic graph representation learning via spiking neural networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8588–8596.
- [207] H. Li, Z. Zhang, D. Liang, and Y. Jiang, “K-truss based temporal graph convolutional network for dynamic graphs,” in *Proceedings of the 15th Asian Conference on Machine Learning*, vol. 222, 2024, pp. 739–754.
- [208] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, “DySAT: Deep neural representation learning on dynamic graphs via self-attention networks,” in *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining*, 2020, pp. 519–527.
- [209] Y. Wang, P. Li, C. Bai, and J. Leskovec, “TEDIC: Neural modeling of behavioral patterns in dynamic social interaction networks,” in *Proceedings of the Web Conference 2021*, 2021, pp. 693–705.
- [210] H. Xue, L. Yang, W. Jiang, Y. Wei, Y. Hu, and Y. Lin, “Modeling dynamic heterogeneous network for link prediction using hierarchical attention with temporal RNN,” in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, vol. 12457, 2021, pp. 282–298.
- [211] S. Zheng, H. Yin, T. Chen, Q. V. H. Nguyen, W. Chen, and L. Zhao, “DREAM: Adaptive reinforcement learning based on attention mechanism for temporal knowledge graph reasoning,” *arXiv preprint arXiv:2304.03984*, 2023.
- [212] J. Hu, Y. Liang, Z. Fan, H. Chen, Y. Zheng, and R. Zimmermann, “Graph neural processes for spatio-temporal extrapolation,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 752–763.
- [213] Z. Feng, R. Wang, T. Wang, M. Song, S. Wu, and S. He, “A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments and challenges,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 38, no. 1, pp. 26–46, 2026.
- [214] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, “Representation learning for dynamic graphs: A survey,” *Journal of Machine Learning Research*, vol. 21, no. 70, pp. 1–73, 2020.
- [215] D. Liben-Nowell and J. Kleinberg, “The link prediction problem for social networks,” in *Proceedings of the 12th ACM International Conference on Information and Knowledge Management*, 2003, pp. 556–559.
- [216] N. M. A. Ibrahim and L. Chen, “An efficient algorithm for link prediction in temporal uncertain social networks,” *Information Sciences*, vol. 331, pp. 120–136, 2016.
- [217] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.
- [218] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 855–864.
- [219] L. Yao, C. Mao, and Y. Luo, “Graph convolutional networks for text classification,” in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, 2019, pp. 7370–7377.
- [220] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, “Graph convolutional networks: a comprehensive review,” *Computational Social Networks*, vol. 6, no. 11, pp. 1–23, 2019.
- [221] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [222] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 1024–1034.
- [223] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,” *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.
- [224] S. Zhang, Y. Xiong, Y. Zhang, X. Wu, Y. Sun, and J. Zhang, “iLoRE: Dynamic graph representation with instant long-term modeling and re-occurrence preservation,” in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3216–3225.
- [225] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, “Measuring and relieving the over-smoothing problem for graph neural networks from the topological view,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3438–3445.
- [226] C. Yang, R. Wang, S. Yao, S. Liu, and T. F. Abdelzaher, “Revisiting over-smoothing in deep GCNs,” *arXiv preprint arXiv:2003.13663*, 2020.
- [227] H. Dai, Y. Wang, R. Trivedi, and L. Song, “Recurrent coevolutionary latent feature processes for continuous-time recommendation,” in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 29–34.
- [228] R. Trivedi, H. Dai, Y. Wang, and L. Song, “Know-evolve: Deep temporal reasoning for dynamic knowledge graphs,” in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 3462–3471.
- [229] W. Jin, M. Qu, X. Jin, and X. Ren, “Recurrent event network: Autoregressive structure inference over temporal knowledge graphs,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 6669–6683.
- [230] J. You, Y. Wang, A. Pal, P. Eksombatchai, C. Rosenberg, and J. Leskovec, “Hierarchical temporal convolutional networks for dynamic recommender systems,” in *Proceedings of the 2019 World Wide Web Conference*, 2019, pp. 2236–2246.
- [231] A. Micheli and D. Tortorella, “Discrete-time dynamic graph echo state networks,” *Neurocomputing*, vol. 496, pp. 85–95, 2022.
- [232] Y. Ma, Z. Guo, Z. Ren, J. Tang, and D. Yin, “Streaming graph neural networks,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 719–728.
- [233] L. Fang, K. Feng, J. Gui, S. Feng, and A. Hu, “Anonymous edge representation for inductive anomaly detection in dynamic bipartite graphs,” *Proceedings of the VLDB Endowment*, vol. 16, no. 5, pp. 1154–1167, 2023.
- [234] Y. Chen, A. Zeng, Q. Yu, K. Zhang, Y. Cao, K. Wu, G. Huzhang, H. Yu, and Z. Zhou, “Recurrent temporal revision graph networks,” in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 69 348–69 360.
- [235] Y. Luo and P. Li, “Neighborhood-aware scalable temporal network representation learning,” in *Proceedings of the First Learning on Graphs Conference*, vol. 198, 2022, pp. 1:1–1:18.
- [236] L. Han, R. Zhang, L. Sun, B. Du, Y. Fu, and T. Zhu, “Generic and dynamic graph representation learning for crowd flow modeling,” in *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4293–4301.
- [237] Z. Zhang, Z. Zhang, X. Wang, Y. Qin, Z. Qin, and W. Zhu, “Dynamic heterogeneous graph attention neural architecture search,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 11 307–11 315.
- [238] Y. Zhang, Y. Xiong, Y. Liao, Y. Sun, Y. Jin, X. Zheng, and Y. Zhu, “TIGER: Temporal interaction graph embedding with restarts,” in *Proceedings of the 32nd ACM World Wide Web Conference*, 2023, pp. 478–488.
- [239] S. Zhang, Y. Xiong, Y. Zhang, Y. Sun, X. Chen, Y. Jiao, and Y. Zhu, “RDGSL: Dynamic graph representation learning with structure learning,” in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3174–3183.

- [240] J. Su, D. Zou, and C. Wu, "PRES: Toward scalable memory-based dynamic graph neural networks," in *The Twelfth International Conference on Learning Representations*, 2024, pp. 1–14.
- [241] H. Li, C. Li, K. Feng, Y. Yuan, G. Wang, and H. Zha, "Robust knowledge adaptation for dynamic graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6920–6933, 2024.
- [242] S. Zhang, X. Chen, Y. Xiong, X. Wu, Y. Zhang, Y. Fu, Y. Zhao, and J. Zhang, "Towards adaptive neighborhood for advancing temporal interaction graph modeling," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4290–4301.
- [243] S. Ji, M. Liu, L. Sun, C. Liu, and T. Zhu, "MemMap: An adaptive and latent memory structure for dynamic graph learning," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 1257–1268.
- [244] G. Sheng, J. Su, C. Huang, and C. Wu, "MSPipe: Efficient temporal GNN training via staleness-aware pipeline," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 2651–2662.
- [245] Y. Zuo, G. Liu, H. Lin, J. Guo, X. Hu, and J. Wu, "Embedding temporal network via neighborhood formation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2857–2866.
- [246] Y. Lu, X. Wang, C. Shi, P. S. Yu, and Y. Ye, "Temporal network embedding with micro- and macro-dynamics," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 469–478.
- [247] Z. Han, Y. Ma, Y. Wang, S. Günnemann, and V. Tresp, "Graph hawkes neural network for forecasting on temporal knowledge graphs," in *Automated Knowledge Base Construction*, 2020.
- [248] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "DyRep: Learning representations over dynamic graphs," in *International Conference on Learning Representations*, 2019, pp. 1–14.
- [249] B. Knyazev, C. Augusta, and G. W. Taylor, "Learning temporal attention in dynamic graphs with bilinear interactions," *PLOS ONE*, vol. 16, no. 3, p. e0247936, 2021.
- [250] Z. Wen and Y. Fang, "TREND: TempoRal event and node dynamics for graph representation learning," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1159–1169.
- [251] Z. Zhao, X. Zhu, T. Xu, A. Lizhiyu, Y. Yu, X. Li, Z. Yin, and E. Chen, "Time-interval aware share recommendation via bi-directional continuous time dynamic graphs," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 822–831.
- [252] C. Chen, H. Geng, N. Yang, X. Yang, and J. Yan, "EasyDGL: Encode, train and interpret for continuous-time dynamic graph learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [253] G. H. Nguyen, J. B. Lee, R. A. Rossi, N. K. Ahmed, E. Koh, and S. Kim, "Dynamic network embeddings: From random walks to temporal random walks," in *Proceedings of the 2018 IEEE International Conference on Dynamic Network Embeddings: From Random Walks to Temporal ... - dblp*(<https://dblp.org/rec/conf/bigdataconf/NguyenLRAKK18>)Conference on Big Data, 2018, pp. 1085–1092.
- [254] Z. Qiu, W. Hu, J. Wu, W. Liu, B. Du, and X. Jia, "Temporal network embedding with high-order nonlinear information," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, 2020, pp. 5436–5443.
- [255] Y. Wang, Y.-Y. Chang, Y. Liu, J. Leskovec, and P. Li, "Inductive representation learning in temporal networks via causal anonymous walks," in *International Conference on Learning Representations*, 2021, pp. 1–14.
- [256] M. Jin, Y.-F. Li, and S. Pan, "Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 19 874–19 886.
- [257] A. Souza, D. Mesquita, S. Kaski, and V. Garg, "Provably expressive temporal graph networks," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 32 257–32 269.
- [258] X. Lu, L. Sun, T. Zhu, and W. Lv, "Improving temporal link prediction via temporal walk matrix projection," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 141 153–141 182.
- [259] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Acham, "Inductive representation learning on temporal graphs," in *Proceedings of the 8th International Conference on Learning Representations*, 2020, pp. 1–14.
- [260] L. Wang, X. Chang, S. Li, Y. Chu, H. Li, W. Zhang, X. He, L. Song, J. Zhou, and H. Yang, "TCL: Transformer-based dynamic graph modelling via contrastive learning," *arXiv preprint arXiv:2105.07944*, 2021.
- [261] K. Feng, C. Li, X. Zhang, and J. Zhou, "Towards open temporal graph neural networks," in *The Eleventh International Conference on Learning Representations*, 2023, pp. 1–14.
- [262] S. Suresh, M. Shrivastava, A. Mukherjee, J. Neville, and P. Li, "Expressive and efficient representation learning for ranking links in temporal graphs," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 567–577.
- [263] S. Ji, X. Lu, M. Liu, L. Sun, C. Liu, B. Du, and H. Xiong, "Community-based dynamic graph learning for popularity prediction," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 930–940.
- [264] M. Alomrani, M. Biparva, Y. Zhang, and M. Coates, "Dyg2vec: Efficient representation learning for dynamic graphs," *Transactions on Machine Learning Research*, 2023.
- [265] Y. Wu, Y. Fang, and L. Liao, "On the feasibility of simple transformer for dynamic graph modeling," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 1–11.
- [266] M. Biparva, R. Karimi, F. Faez, and Y. Zhang, "Todyformer: Towards holistic dynamic graph transformers with structure-aware tokenization," *Transactions on Machine Learning Research*, 2024.
- [267] L. Yu, L. Sun, B. Du, and W. Lv, "Towards better dynamic graph learning: New architecture and unified library," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 67 686–67 700.
- [268] X. Wang, D. Lyu, M. Li, Y. Xia, Q. Yang, X. Wang, X. Wang, P. Cui, Y. Yang, B. Sun, and Z. Guo, "APAN: Asynchronous propagation attention network for real-time temporal graph embedding," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 2628–2638.
- [269] L. Qu, H. Zhu, Q. Duan, and Y. Shi, "Continuous-time link prediction via temporal dependent graph neural network," in *Proceedings of The Web Conference 2020*, 2020, pp. 3026–3032.
- [270] H. Tang, S. Wu, G. Xu, and Q. Li, "Dynamic graph evolution learning for recommendation," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 1589–1598.
- [271] C. Wu, C. Wang, J. Xu, Z. Fang, T. Gu, C. Wang, Y. Song, K. Zheng, X. Wang, and G. Zhou, "Instant representation learning for recommendation over large dynamic graphs," in *Proceedings of the 2023 IEEE 39th International Conference on Data Engineering*, 2023, pp. 82–95.
- [272] Y. Tian, Y. Qi, and F. Guo, "FreeDyG: Frequency enhanced continuous-time dynamic graph model for link prediction," in *The Twelfth International Conference on Learning Representations*, 2024, pp. 1–14.
- [273] K. Cheng, L. Peng, J. Ye, L. Sun, and B. Du, "Co-neighbor encoding schema: A light-cost structure encoding method for dynamic link prediction," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 421–432.
- [274] S. Zhang, X. Chen, Y. Xiong, X. Wu, Y. Jiao, Y. Zhang, M. Zhang, T. Liu, W. Wang, and J. Zhang, "Interactions exhibit clustering rhythm: A prevalent observation for advancing temporal link prediction," in *International Conference on Learning Representations*, 2025.
- [275] Z. Ding, Y. Li, Y. He, A. Norelli, J. Wu, V. Tresp, M. M. Bronstein, and Y. Ma, "Dygnamba: Efficiently modeling long-term temporal dependency on continuous-time dynamic graphs with state space models," *Transactions on Machine Learning Research*, 2025.
- [276] W. Cong, S. Zhang, J. Kang, B. Yuan, H. Wu, X. Zhou, H. Tong, and M. Mahdavi, "Do we really need complicated model architectures for temporal networks?" in *Proceedings of the Eleventh International Conference on Learning Representations*, 2023, pp. 1–14.
- [277] T. Zou, Y. Mao, J. Ye, and B. Du, "Repeat-aware neighbor sampling for dynamic graph learning," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4722–4733.
- [278] Y. Li, Y. Xu, X. Lin, W. Zhang, and Y. Zhang, "Ranking on dynamic graphs: An effective and robust band-pass disentangled approach," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 3918–3929.
- [279] E. Lezmi and J. Xu, "Time series forecasting with transformer models and application to asset management," *Amundi Research Center Working Paper*, no. 139, pp. 1–44, 2023.
- [280] W.-C. Kang and J. J. McAuley, "Self-attentive sequential recommendation," in *Proceedings of the 2018 IEEE International Conference on Data Mining*, 2018, pp. 197–206.

- [281] J. Skarding, B. Gabrys, and K. Musial, "Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey," *IEEE Access*, vol. 9, pp. 79 143–79 168, 2021.
- [282] P. Jiao, H. Chen, H. Tang, Q. Bao, L. Zhang, Z. Zhao, and H. Wu, "Contrastive representation learning on dynamic networks," *Neural Networks*, vol. 174, p. 106240, 2024.
- [283] S. Huang, F. Pourafaie, R. Rabbany, G. Rabusseau, and E. Rossi, "UTG: Towards a unified view of snapshot and event based models for temporal graphs," in *Proceedings of the Third Learning on Graphs Conference*, vol. 269, 2025, pp. 28:1–28:16.
- [284] P. Jiao, H. Chen, X. Guo, Z. Zhao, D. He, and D. Jin, "A survey on temporal interaction graph representation [find results for 'bibtext'](https://www.ijcai.org/proceedings/2025/1166)learning: Progress, challenges, and opportunities," in *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025, pp. 10 499–10 507.
- [285] S. ENNADIR, G. Z. Gandler, F. Cornell, L. Cao, O. Smirnov, T. Wang, L. Zólyomi, B. Brinne, and S. Asadi, "Expressivity of representation learning on continuous-time dynamic graphs: An information-flow centric review," *Transactions on Machine Learning Research*, 2025.
- [286] X. Chen, S. Zhang, Y. Xiong, X. Wu, J. Zhang, X. Sun, Y. Zhang, F. Zhao, and Y. Kang, "Prompt learning on temporal interaction graphs," *arXiv preprint arXiv:2402.06326*, 2024.
- [287] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Acham, "Self-attention with functional time representation learning," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 15 915–15 925.
- [288] H. Zhou, D. Zheng, I. Nisa, V. Ioannidis, X. Song, and G. Karypis, "TGL: A general framework for temporal GNN training on billion-scale graphs," *Proceedings of the VLDB Endowment*, vol. 15, no. 8, pp. 1572–1580, 2022.
- [289] H. Zhou, D. Zheng, X. Song, G. Karypis, and V. Prasanna, "DistTGL: Distributed memory-based temporal graph neural network training," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, 2023, pp. 1–11.
- [290] X. Chen, Y. Liao, Y. Xiong, Y. Zhang, S. Zhang, J. Zhang, and Y. Sun, "SPEED: Streaming partition and parallel acceleration for temporal interaction graph embedding," *arXiv preprint arXiv:2308.14129*, 2023.
- [291] Y. Zhong, G. Sheng, T. Qin, M. Wang, Q. Gan, and C. Wu, "GNNflow: A distributed framework for continuous temporal GNN learning on dynamic graphs," *arXiv preprint arXiv:2311.17410*, 2023.
- [292] A. N. Eddin, J. Bono, D. O. Aparicio, H. Ferreira, P. M. P. Ribeiro, and P. Bizarro, "Deep-graph-sprints: Accelerated representation learning in continuous-time dynamic graphs," *Transactions on Machine Learning Research*, 2024.
- [293] OpenAI, "GPT-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.
- [294] Gemini Team, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.
- [295] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "PaLM: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.
- [296] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambré, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.
- [297] Q. Li, L. Fu, W. Zhang, X. Chen, J. Yu, W. Xia, W. Zhang, R. Tang, and Y. Yu, "Adapting large language models for education: Foundational capabilities, potentials, and challenges," *arXiv preprint arXiv:2401.08664*, 2024.
- [298] O. Nov, N. Singh, and D. M. Mann, "Putting ChatGPT's medical advice to the (turing) test: Survey study," *JMIR Medical Education*, vol. 9, p. e46939, 2023.
- [299] S. Wu, O. Irsoy, S. Lu, V. Dabrowski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "BloombergGPT: A large language model for finance," *arXiv preprint arXiv:2303.17564*, 2023.
- [300] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, and X. He, "TALLRec: An effective and efficient tuning framework to align large language model with recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1007–1014.
- [301] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, "Chat-rec: Towards interactive and explainable LLMs-augmented recommender system," *arXiv preprint arXiv:2303.14524*, 2023.
- [302] J. Chen, "A survey on large language models for personalized and explainable recommendations," *arXiv preprint arXiv:2311.12338*, 2023.
- [303] M. Marripudugala, "Real-time IoT data analytics using advanced large language model techniques," in *Proceedings of the 2024 Global Conference on Communications and Information Technologies*, 2024, pp. 1–6.
- [304] N. Gao, Z. Yu, Y. Xu, C. Yu, Y. Wang, F. D. Salim, and Y. Shi, "Leveraging large language models for generating mobile sensing strategies in human behavior modeling," in *Companion of the 2024 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2024, pp. 729–735.
- [305] Y. Xi, W. Liu, J. Lin, X. Cai, H. Zhu, J. Zhu, B. Chen, R. Tang, W. Zhang, and Y. Yu, "Towards open-world recommendation with knowledge augmentation from large language models," in *Proceedings of the 18th ACM Conference on Recommender Systems*, 2024, pp. 12–22.
- [306] F. Liu, Y. Liu, H. Chen, Z. Cheng, L. Nie, and M. S. Kankanhalli, "Understanding before recommendation: Semantic aspect-aware review exploitation via large language models," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–26, 2025.
- [307] G. H. Torbati, A. Tiginova, A. Yates, and G. Weikum, "Recommendations by concise user profiles from review text," *arXiv preprint arXiv:2311.01314*, 2023.
- [308] G. Cao, K. Shi, H. Fu, H. Zhang, and G. Xu, "APLe: Token-wise adaptive for multi-modal prompt learning," *arXiv preprint arXiv:2401.06827*, 2024.
- [309] L. Ren, Y. Liu, C. Ouyang, Y. Yu, S. Zhou, Y. He, and Y. Wan, "DyLas: A dynamic label alignment strategy for large-scale multi-label text classification," *Information Fusion*, vol. 120, p. 103081, 2025.
- [310] P. Pandey and J. P. Singh, "Generating product reviews from aspect-based ratings using large language models," *Journal of Retailing and Consumer Services*, vol. 84, p. 104244, 2025.
- [311] C. Qin, L. Zhang, Y. Cheng, R. Zha, D. Shen, Q. Zhang, X. Chen, Y. Sun, C. Zhu, H. Zhu, and H. Xiong, "A comprehensive survey of artificial intelligence techniques for talent analytics," *Proceedings of the IEEE*, vol. 113, no. 2, pp. 125–171, 2025.
- [312] H. Ghanem and C. Cruz, "Fine-tuning vs. prompting: Evaluating the knowledge graph construction with LLMs," in *Proceedings of the 3rd International Workshop on Knowledge Graph Generation from Text, co-located with the Extended Semantic Web Conference*, vol. 3747, 2024.
- [313] ———, "Enhancing knowledge graph construction: Evaluating with emphasis on hallucination, omission, and graph similarity metrics," in *Knowledge Graphs and Semantic Web: 6th International Conference, KGSWC 2024, Paris, France, December 11–13, 2024, Proceedings*, vol. 15459, 2025, pp. 32–46.
- [314] A. G. Regino, V. Hochgreb, and J. C. dos Reis, "Humanizing answers for compatibility questions in e-commerce using large language models," in *Proceedings of the 39th Brazilian Symposium on Databases*, 2024, pp. 300–312.
- [315] Z. Gao, L. Li, S. Ma, Q. Wang, L. Hemphill, and R. Xu, "Examining the potential of ChatGPT on biomedical information retrieval: Fact-checking drug-disease associations," *Annals of Biomedical Engineering*, vol. 52, no. 8, pp. 1919–1927, 2024.
- [316] D. Sidorenko, S. Pushkov, A. Sakip, G. H. D. Leung, S. W. Y. Lok, A. Urban, D. Zagirova, A. Veviorksiy, N. Tihonova, A. Kalashnikov, E. Kozlova, V. Naumov, F. W. Pun, A. Aliper, F. Ren, and A. Zhavoronkov, "Precious2GPT[Precious2GPT: the combination of multiomics pretrained ... - Europe PMC](https://europepmc.org/article/MED/39117678): the combination of multiomics pretrained transformer and conditional diffusion for artificial multi-omics multi-species multi-tissue sample generation," *npg Aging*, vol. 10, no. 1, p. 37, 2024.
- [317] A. G. Carranza, R. Farahani, N. Ponomareva, A. Kurakin, M. Jagielski, and M. Nasr, "Synthetic query generation for privacy-preserving deep retrieval systems using differentially private language models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 3920–3930.

- [318] Q. Chen, W. Shuai, J. Zhang, Z. Sun, and N. Cao, “Beyond numbers: Creating analogies to enhance data comprehension and communication with generative AI,” in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–14.
- [319] P. Jiang, C. Xiao, Z. Wang, P. Bhatia, J. Sun, and J. Han, “TriSum: Learning summarization ability from large language models with structured rationale,” in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 2805–2819.
- [320] J. Wu, Q. Liu, H. Hu, W. Fan, S. Liu, Q. Li, X.-M. Wu, and K. Tang, “Leveraging ChatGPT to empower training-free dataset condensation for content-based recommendation,” in *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 1402–1406.
- [321] L. Liang, M. Sun, Z. Gui, Z. Zhu, Z. Jiang, L. Zhong, Y. Qu, P. Zhao, Z. Bo, J. Yang, H. Xiong, L. Yuan, J. Xu, Z. Wang, Z. Zhang, W. Zhang, H. Chen, W. Chen, and J. Zhou, “KAG: Boosting LLMs in professional domains via knowledge augmented generation,” in *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 334–343.
- [322] J. Jiang and E. Ferrara, “Social-LLM: Modeling user behavior at scale using language models and social network data,” *Sci*, vol. 7, no. 4, p. 138, 2025.
- [323] P. Srivastava, M. Malik, V. Gupta, T. Ganu, and D. Roth, “Evaluating LLMs’ mathematical reasoning in financial document question answering,” in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 3853–3878.
- [324] S. Gao, J. Fang, Q. Tu, Z. Yao, Z. Chen, P. Ren, and Z. Ren, “Generative news recommendation,” in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 3444–3453.
- [325] T. Lu, Z. Zhou, J. Wang, and Y. Wang, “A large language model-based approach for personalized search results re-ranking in professional domains,” *Academia Nexus Journal*, vol. 4, no. 1, 2025.
- [326] K. Chavinda and U. Thayasiyam, “A dual contrastive learning framework for enhanced hate speech detection in low-resource languages,” in *Proceedings of the First Workshop on Challenges in Processing South Asian Languages*, 2025, pp. 115–123.
- [327] F. Nazary, Y. Deldjoo, and T. Di Noia, “Poison-RAG: Adversarial data poisoning attacks on retrieval-augmented generation in recommender systems,” in *Advances in Information Retrieval: 47th European Conference on Information Retrieval, Lucca, Italy, April 6–10, 2025, Proceedings, Part IV*, vol. 15575, 2025, pp. 239–251.
- [328] Z. Qiu, X. Wu, J. Gao, and W. Fan, “U-BERT: Pre-training user representations for improved recommendation,” in *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4320–4327.
- [329] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [330] B. Zheng, Y. Hou, H. Lu, Y. Chen, W. X. Zhao, M. Chen, and J.-R. Wen, “Adapting large language models by integrating collaborative semantics for recommendation,” in *Proceedings of the 2024 IEEE 40th International Conference on Data Engineering*, 2024, pp. 1435–1448.
- [331] Y. Wang, J. Xun, M. Hong, J. Zhu, T. Jin, W. Lin, H. Li, L. Li, Y. Xia, Z. Zhao, and Z. Dong, “EAGER: Two-stream generative recommender with behavior-semantic collaboration,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 3245–3254.
- [332] J. Han, K. Gong, Y. Zhang, J. Wang, K. Zhang, D. Lin, Y. Qiao, P. Gao, and X. Yue, “OneLLM: One framework to align all modalities with language,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26584–26595.
- [333] J. Zhang, W. Sun, Y. Hou, W. X. Zhao, and J.-R. Wen, “Review-enhanced universal sequence representation learning for recommender systems,” *ACM Transactions on Information Systems*, vol. 43, pp. 1–31, 2025.
- [334] B. Yang, Y. Guo, L. Xu, Z. Yan, H. Chen, G. Xing, and X. F. Jiang, “SocialMind: LLM-based proactive AR social assistive system with human-like perception for in-situ live interactions,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 9, no. 1, pp. 1–30, 2025.
- [335] J. A. Brooks, V. Tiruvadi, A. Baird, P. Tzirakis, H. Li, C. Gagne, M. Oh, and A. Cowen, “Emotion expression estimates to measure and improve multimodal social-affective interactions,” in *Companion Publication of the 25th International Conference on Multimodal Interaction*, 2023, pp. 353–358.
- [336] Z. Fu, X. Li, C. Wu, Y. Wang, K. Dong, X. Zhao, M. Zhao, H. Guo, and R. Tang, “A unified framework for multi-domain CTR prediction via large language models,” *ACM Transactions on Information Systems*, vol. 43, no. 5, pp. 117:1–117:33, 2025.
- [337] Z. Tian, C. Zhang, W. X. Zhao, X. Zhao, J.-R. Wen, and Z. Cao, “UFIN: Universal feature interaction network for multi-domain click-through rate prediction,” in *Database Systems for Advanced Applications: 30th International Conference, Singapore, May 26–29, 2025, Proceedings, Part II*, vol. 15987, 2026, pp. 3–18.
- [338] J. Lin, X. Dai, R. Shan, B. Chen, R. Tang, Y. Yu, and W. Zhang, “Large language models make sample-efficient recommender systems,” *Frontiers of Computer Science*, vol. 19, no. 4, p. 194328, 2025.
- [339] S. Zhang, Y. Xiong, Y. Tang, J. Xu, X. Chen, Z. Gu, X. Hao, Z. Jia, and J. Zhang, “Unifying text semantics and graph structures for temporal text-attributed graphs with large language models,” in *Advances in Neural Information Processing Systems*, 2025.
- [340] L. Friedman, S. Ahuja, D. Allen, Z. Tan, H. Sidahmed, C. Long, J. Xie, G. Schubiner, A. Patel, H. Lara, B. Chu, Z. Chen, and M. Tiwari, “Leveraging large language models in conversational recommender systems,” *arXiv preprint arXiv:2305.07961*, 2023.
- [341] Y. Wang, Z. Jiang, Z. Chen, F. Yang, Y. Zhou, E. Cho, X. Fan, Y. Lu, X. Huang, and Y. Yang, “Recmind: Large language model powered agent for recommendation,” in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 4351–4364.
- [342] Y. Yu, Z. Yao, H. Li, Z. Deng, Y. Jiang, Y. Cao, Z. Chen, J. W. Suchow, Z. Cui, R. Liu, Z. Xu, D. Zhang, K. Subbalakshmi, G. Xiong, Y. He, J. Huang, D. Li, and Q. Xie, “FinCon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making,” in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 137010–137045.
- [343] W. Zhang, L. Zhao, H. Xia, S. Sun, J. Sun, M. Qin, X. Li, Y. Zhao, Y. Zhao, X. Cai, L. Zheng, X. Wang, and B. An, “A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4314–4325.
- [344] Y. Xiao, E. Sun, D. Luo, and W. Wang, “Tradingagents: Multi-agents LLM financial trading framework,” *arXiv preprint arXiv:2412.20138*, 2024.
- [345] W. Ye, W. Yang, D. Cao, Y. Zhang, L. Tang, J. Cai, and Y. Liu, “Domain-oriented time series inference agents for reasoning and automated analysis,” *arXiv preprint arXiv:2410.04047*, 2024.
- [346] Y. Cao, Z. Chen, P. Kumar, Q. Pei, Y. Yu, H. Li, F. Dimino, L. Ausiello, K. P. Subbalakshmi, and P. M. Ndiaye, “RiskLabs: Predicting financial risk using large language model based on multimodal and multi-sources data,” *arXiv preprint arXiv:2404.07452*, 2024.
- [347] J. Piao, Y. Yan, J. Zhang, N. Li, J. Yan, X. Lan, Z. Lu, Z. Zheng, J. Y. Wang, D. Zhou, C. Gao, F. Xu, F. Zhang, K. Rong, J. Su, and Y. Li, “Agentsociety: Large-scale simulation of LLM-driven generative agents advances understanding of human behaviors and society,” *arXiv preprint arXiv:2502.08691*, 2025.
- [348] Y. Zhou, H. Wang, Q. Ai, Z. Wu, and Y. Liu, “Investigating prosocial behavior theory in LLM agents under policy-induced inequities,” *arXiv preprint arXiv:2505.15857*, 2025.
- [349] B. Mobasher, R. Cooley, and J. Srivastava, “Automatic personalization based on web usage mining,” *Communications of the ACM*, vol. 43, no. 8, pp. 142–151, 2000.
- [350] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural collaborative filtering,” in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182.
- [351] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, “AutoInt: Automatic feature interaction learning via self-attentive neural networks,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1161–1170.
- [352] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized Markov chains for next-basket recommendation,” in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 811–820.
- [353] W.-C. Kang and J. J. McAuley, “Self-attentive sequential recommendation,” in *Proceedings of the 2018 IEEE International Conference on Data Mining*, 2018, pp. 197–206.
- [354] X. Li, C. Wang, J. Tan, X. Zeng, D. Ou, and B. Zheng, “Adversarial multimodal representation learning for click-through rate prediction,” in *Proceedings of The Web Conference 2020*, 2020, pp. 827–836.
- [355] G. d. S. P. Moreira, S. Rabhi, R. Ak, M. Y. Kabir, and E. Oldridge, “Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation,” *arXiv preprint arXiv:2107.05124*, 2021.

- [356] W. Wu, X. Li, L. Wang, J. Zhou, D. Wu, Q. Xie, Q. Zhang, Y. Zhang, S. Han, F. Huang, and J. Chen, “IU4Rec: Interest unit-based product organization and recommendation for e-commerce platform,” in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, vol. 2, 2025, pp. 5039–5048.
- [357] M. Zhang, S. Wu, X. Yu, Q. Liu, and L. Wang, “Dynamic graph neural networks for sequential recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4741–4753, 2023.
- [358] H. Li, Y. Xu, Y. Li, H. Liu, D. Li, C. J. Zhang, L. Chen, and Q. Li, “When speed meets accuracy: An efficient and effective graph model for temporal link prediction,” *Proceedings of the VLDB Endowment*, vol. 18, no. 10, pp. 3396–3405, 2025.
- [359] K. Zhao, Y. Zheng, T. Zhuang, X. Li, and X. Zeng, “Joint learning of e-commerce search and recommendation with a unified graph neural network,” in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1461–1469.
- [360] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, “BERT4rec: Sequential recommendation with bidirectional encoder representations from transformer,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 2019, pp. 1441–1450.
- [361] S. Geng, S. Liu, Z. Fu, Y. Ge, and Y. Zhang, “Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5),” in *Proceedings of the Sixteenth ACM Conference on Recommender Systems*, 2022.
- [362] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu, H. Xiong, and E. Chen, “A survey on large language models for recommendation,” *arXiv preprint arXiv:2305.19860*, 2023.
- [363] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, “Chat-rec: Towards interactive and explainable LLMs-augmented recommender system,” *arXiv preprint arXiv:2303.14524*, 2023.
- [364] Z. Cui, J. Ma, C. Zhou, J. Zhou, and H. Yang, “M6-Rec: Generative pretrained language models are open-ended recommender systems,” *arXiv preprint arXiv:2205.08084*, 2022.
- [365] L. Podo, M. Angelini, and P. Velardi, “V-RECS, a low-cost LLM4VIS recommender with explanations, captioning and suggestions,” *arXiv preprint arXiv:2406.15259*, 2024.
- [366] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnelsson, T. Boulos, and J. Kubica, “Ad click prediction: a view from the trenches,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 1222–1230.
- [367] S. Rendle, “Factorization machines,” in *Proceedings of the 2010 IEEE International Conference on Data Mining*, 2010, pp. 995–1000.
- [368] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Quiñonero Candela, “Practical lessons from predicting clicks on ads at Facebook,” in *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, 2014, pp. 1–9.
- [369] G. Liu, T. T. Nguyen, G. Zhao, W. Zha, J. Yang, J. Cao, M. Wu, P. Zhao, and W. Chen, “Repeat buyer prediction for e-commerce,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 155–164.
- [370] Y. Ni, D. Ou, S. Liu, X. Li, W. Ou, A. Zeng, and L. Si, “Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 596–605.
- [371] Y. Gu, Z. Ding, S. Wang, and D. Yin, “Hierarchical user profiling for E-commerce recommender systems,” in *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining*, 2020, pp. 223–231.
- [372] A. Gupta, P. Kumar, A. Mishra, A. Singh, S. Kumar, M. Chelliah, A. Chakraborty, and S. Ranu, “Persona identification in E-Commerce with scarce labels and in-context graph learning,” in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, vol. 2, 2025, pp. 778–789.
- [373] J. Zhou, Y. Dai, and T. Joachims, “Language-based user profiles for recommendation,” *arXiv preprint arXiv:2402.15623*, 2024.
- [374] J. Zhang, “Guided profile generation improves personalization with large language models,” in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 4005–4016.
- [375] M. Sabouri, M. Mansoury, K. Lin, and B. Mobasher, “Towards explainable temporal user profiling with LLMs,” in *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, 2025, pp. 219–227.
- [376] S. Bang and H. Song, “LLM-based user profile management for recommender system,” *arXiv preprint arXiv:2502.14541*, 2025.
- [377] E. Choi, M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun, “RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism,” in *Advances in Neural Information Processing Systems*, vol. 29, 2016, pp. 3512–3520.
- [378] Y. Li, S. Rao, J. R. A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, and G. Salimi-Khorshidi, “BEHRT: Transformer for electronic health records,” *Scientific Reports*, vol. 10, no. 1, p. 7155, 2020.
- [379] Z. Cai, Y. Liu, Z. Luo, and T. Zhu, “ProtoEHR: Hierarchical prototype learning for EHR-based healthcare predictions,” *arXiv preprint arXiv:2508.18313*, 2025.
- [380] F. Liu, L. Li, X. Wang, F. Luo, C. Liu, J. Su, and Y. Qian, “MH-GRL: An effective representation learning model for electronic health records,” in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, 2024, pp. 11 272–11 282.
- [381] W. Shi, R. Xu, Y. Zhuang, Y. Yu, J. Zhang, H. Wu, Y. Zhu, J. C. Ho, C. Yang, and M. D. Wang, “EHRAgent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 22 315–22 339.
- [382] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, “DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network,” *BMC Medical Research Methodology*, vol. 18, p. 24, 2018.
- [383] J. M. Lee and M. Hauskrecht, “Personalized event prediction for Electronic Health Records,” *Artificial Intelligence in Medicine*, vol. 143, p. 102620, 2023.
- [384] L. Wang, W. Zhang, X. He, and H. Zha, “Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2447–2456.
- [385] P. Jiang, C. Xiao, M. Jiang, P. Bhatia, T. Kass-Hout, J. Sun, and J. Han, “Reasoning-enhanced healthcare predictions with knowledge graph community retrieval,” in *Proceedings of the 13th International Conference on Learning Representations*, 2025, pp. 1–46.
- [386] D. Zhou, H. Tong, L. Wang, S. Liu, X. Xiong, Z. Gan, R. Griffier, B. P. Hejblum, Y.-C. Liu, C. Hong, C.-L. Bonzel, T. Cai, K. Pan, Y.-L. Ho, L. Costa, V. A. Panickan, J. M. Gaziano, K. D. Mandl, V. Jouhet, R. Thiebaut, Z. Xia, K. Cho, K. P. Liao, and T. Cai, “Representation learning to advance multi-institutional studies with electronic health record data,” *arXiv preprint arXiv:2502.08547*, 2025.
- [387] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, M. Amin, L. Hou, K. Clark, S. R. Pfohl, H. Cole-Lewis, D. Neal, Q. M. Rashid, M. Schaekermann, A. Wang, D. Dash, J. Chen, N. H. Shah, S. Lachgar, P. Mansfield, S. Prakash, B. Green, E. Dominowska, B. Agüera y Arcas, N. Tomasev, Y. Liu, R. Wong, C. Semturs, S. S. Mahdavi, J. K. Barral, D. Webster, G. S. Corrado, Y. Matias, S. Azizi, A. Karthikesalingam, and V. Natarajan, “Toward expert-level medical question answering with large language models,” *Nature Medicine*, vol. 31, no. 3, pp. 943–950, 2025.
- [388] H. Peters and S. C. Matz, “Large language models can infer psychological dispositions of social media users,” *PNAS Nexus*, vol. 3, no. 6, p. pgae231, 2024.
- [389] M. Sabouri, M. Mansoury, K. Lin, and B. Mobasher, “Towards explainable temporal user profiling with LLMs,” in *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*. Association for Computing Machinery, 2025, pp. 219–227.
- [390] P. Panayotov, U. Shukla, H. T. Sencar, M. Nabeel, and P. Nakov, “GREENER: Graph neural networks for news media profiling,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 7470–7480.
- [391] G. Farnadi, L. Getoor, M.-F. Moens, and M. De Cock, “User profiling using hinge-loss markov random fields,” *arXiv preprint arXiv:2001.01177*, 2020.
- [392] C. H. Liu, J. Xu, J. Tang, and J. Crowcroft, “Social-aware sequential modeling of user interests: A deep learning approach,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 11, pp. 2200–2212, 2019.
- [393] Y. Yang, L. Zhang, and J. Liu, “Temporal user interest modeling for online advertising using Bi-LSTM network improved by an updated

- version of parrot optimizer,” *Scientific Reports*, vol. 15, no. 1, p. 18858, 2025.
- [394] C. Zhang, Y. Sun, J. Chen, J. Lei, M. Abdul-Mageed, S. Wang, R. Jin, S. Park, N. Yao, and B. Long, “SPAR: Personalized content-based recommendation via long engagement attention,” *arXiv preprint arXiv:2402.10555*, 2024.
- [395] S. Yang, W. Ma, P. Sun, Q. Ai, Y. Liu, M. Cai, and M. Zhang, “Sequential recommendation with latent relations based on large language model,” in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 335–344.
- [396] C. Liu, J. Lin, J. Wang, H. Liu, and J. Caverlee, “Mamba4Rec: Towards efficient sequential recommendation with selective state space models,” *arXiv preprint arXiv:2403.03900*, 2024.
- [397] Y. Yuan, Z. Zhang, X. He, A. Nitta, W. Hu, M. Shah, B. Stojanović, S. Huang, J. E. Lenssen, J. Leskovec, and M. Fey, “ContextGNN: Beyond two-tower recommendation systems,” in *Proceedings of the International Conference on Learning Representations*, 2025, pp. 1–14.
- [398] S. Yang, Y. Liu, Y. Xu, C. Miao, M. Wu, and J. Zhang, “Contextualized graph attention network for recommendation with item knowledge graph,” *arXiv preprint arXiv:2004.11529*, 2020.
- [399] M. Chen, L. Wei, H. Cao, W. Zhou, and S. Hu, “Explore the potential of LLMs in misinformation detection: An empirical study,” *arXiv preprint arXiv:2311.12699*, 2024.
- [400] R. Ruiz-Dolz and J. Lawrence, “An explainable framework for misinformation identification via critical question answering,” *arXiv preprint arXiv:2503.14626*, 2025.
- [401] T. Liu, Q. Cai, C. Xu, B. Hong, F. Ni, Y. Qiao, and T. Yang, “Rumor detection with a novel graph neural network approach,” *arXiv preprint arXiv:2403.16206*, 2024.
- [402] M. Ma, C. Zhang, Y. Li, J. Chen, and X. Wang, “Rumor detection model with weighted GraphSAGE focusing on node location,” *Scientific Reports*, vol. 14, no. 1, pp. 1–13, 2024.
- [403] X. Dong and L. Qian, “Semi-supervised bidirectional RNN for misinformation detection,” *Machine Learning with Applications*, vol. 10, p. 100428, 2022.
- [404] C.-O. Truică, E.-S. Apostol, M. Marogel, and A. Paschke, “GETAE: Graph information Enhanced deep neural NeTwork ensemble ArchitecturE for fake news detection,” *arXiv preprint arXiv:2412.01825*, 2024.
- [405] J. Xu, “A multimodal adaptive graph-based intelligent classification model for fake news,” *arXiv preprint arXiv:2411.06097*, 2024.
- [406] M. Kamphuis, “Tiny-toxic-detector: A compact transformer-based model for toxic content detection,” *arXiv preprint arXiv:2409.02114*, 2024.
- [407] G. K. Shahi and T. A. Majchrzak, “Defining, understanding, and detecting online toxicity: Challenges and machine learning approaches,” *arXiv preprint arXiv:2509.14264*, 2025.
- [408] Y. Zhao, J. Zhu, C. Xu, Y. Liu, and X. Li, “Enhancing LLM-based hatred and toxicity detection with meta-toxic knowledge graph,” in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 24747–24760.
- [409] S. Cui, Q. Zhang, X. Ouyang, R. Chen, Z. Zhang, Y. Lu, H. Wang, H. Qiu, and M. Huang, “ShieldVLM: Safeguarding the multimodal implicit toxicity via deliberative reasoning with LVLMs,” *arXiv preprint arXiv:2505.14035*, 2025.
- [410] L. Cheng, A. Mosallanezhad, Y. N. Silva, D. L. Hall, and H. Liu, “Bias mitigation for toxicity detection via sequential decisions,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1750–1760.
- [411] T. T. Oyetayo, J. O. Adeniran, S. Adebanji, T. A. Adekunle, and O. Y. Adebayo, “A neural multi-label model for enhanced toxicity detection in online conversations,” *Adeleke University Journal of Science*, vol. 3, no. 1, pp. 9–15, 2024.
- [412] R. Prabhu and V. Seethalakshmi, “A comprehensive framework for multi-modal hate speech detection in social media using deep learning,” *Scientific Reports*, vol. 15, no. 1, pp. 1–21, 2025.
- [413] I. Xavier, E. Veiga, A. M. N. Ribeiro, R. de Paula Monteiro, M. Oliveira, R. Amaro, and D. Pinheiro, “Towards the identification of money laundering in bank transactions using recurrent neural networks,” in *Proceedings of the 2024 IEEE Latin American Conference on Computational Intelligence*, 2024, pp. 1–5.
- [414] B. Branco, P. Abreu, A. S. Gomes, M. S. C. Almeida, J. T. Ascensão, and P. Bizarro, “Interleaved sequence RNNs for fraud detection,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 3101–3109.
- [415] W. Min, W. Liang, H. Yin, Z. Wang, M. Li, and A. Lal, “Explainable deep behavioral sequence clustering for transaction fraud detection,” *arXiv preprint arXiv:2101.04285*, 2021.
- [416] C. Liu, Y. Gao, L. Sun, J. Feng, H. Yang, and X. Ao, “User behavior pre-training for online fraud detection,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 3357–3365.
- [417] S. Huang, Y. Xiong, Y. Xie, T. Qiu, and G. Wang, “Robust sequence-based self-supervised representation learning for anti-money laundering,” in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 4571–4578.
- [418] X. Wang, J. Guo, X. Luo, and H. Yu, “DyHDGE: Dynamic heterogeneous transaction graph embedding for safety-centric fraud detection in financial scenarios,” *Journal of Safety Science and Resilience*, vol. 5, no. 4, pp. 486–497, 2024.
- [419] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson, “EvolveGCN: Evolving graph convolutional networks for dynamic graphs,” in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5363–5370.
- [420] S. Khodabandehlou and A. H. Golpayegani, “FiFrauD: unsupervised financial fraud detection in dynamic graph streams,” *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 5, pp. 1–29, 2024.
- [421] S. Lou, Q. Zhang, S. Yang, Y. Tian, Z. Tan, and M. Luo, “GADY: Unsupervised anomaly detection on dynamic graphs,” *arXiv preprint arXiv:2310.16376*, 2023.
- [422] J. Chen, S. Fu, Z. Ma, M. Feng, T. S. Wirjanto, and Q. Peng, “Semi-supervised anomaly detection with extremely limited labels in dynamic graphs,” in *Proceedings of the 30th International Conference on Database Systems for Advanced Applications*, vol. 15987, 2025, pp. 373–382.
- [423] Z. Gu, Y. Tang, J. Xu, S. Zhang, X. Zheng, X. Chen, and Y. Xiong, “SSH-T3: A hierarchical pre-training framework for multi-scenario financial risk assessment,” in *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 2025, pp. 5674–5682.
- [424] L. Jiang, “Detecting scams using large language models,” *arXiv preprint arXiv:2402.03147*, 2024.
- [425] S. Korkanti, “Enhancing financial fraud detection using LLMs and advanced data analytics,” in *Proceedings of the 2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems*, 2024, pp. 1328–1334.
- [426] G. Singh, P. Singh, and M. Singh, “Advanced real-time fraud detection using RAG-based LLMs,” *arXiv preprint arXiv:2501.15290*, 2025.
- [427] Y. Cao, Z. Chen, P. Kumar, Q. Pei, Y. Yu, H. Li, F. Dimino, L. Ausiello, K. P. Subbalakshmi, and P. M. Ndiaye, “RiskLabs: Predicting financial risk using large language model based on multimodal and multi-sources data,” *arXiv preprint arXiv:2404.07452*, 2024.
- [428] V.-D. Ta, C.-M. Liu, and D. A. Tadesse, “Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading,” *Applied Sciences*, vol. 10, no. 2, p. 437, 2020.
- [429] C. Han and X. Fu, “Challenge and opportunity: Deep learning-based stock price prediction by using bi-directional LSTM model,” *Frontiers in Business, Economics and Management*, vol. 8, no. 2, pp. 51–54, 2023.
- [430] C.-H. Wang, J. Yuan, Y. Zeng, and S. Lin, “A deep learning integrated framework for predicting stock index price and fluctuation via singular spectrum analysis and particle swarm optimization,” *Applied Intelligence*, vol. 54, no. 2, pp. 1770–1797, 2024.
- [431] T. Zhang, M. D. Huo, Z. Ma, J. Hu, Q. Liang, and H. Chen, “Prediction model of stock return on investment based on hybrid DNN and TabNet model,” *PeerJ Computer Science*, vol. 10, p. e2057, 2024.
- [432] M. Wysocki and P. Sakowski, “Investment portfolio optimization based on modern portfolio theory and deep learning models,” Faculty of Economic Sciences, University of Warsaw, Tech. Rep. Working Papers No. 12/2022 (388), 2022.
- [433] P. Singh, M. Jha, M. Sharaf, M. A. El-Meligy, and T. R. Gadekallu, “Harnessing a hybrid CNN-LSTM model for portfolio performance: A case study on stock selection and optimization,” *IEEE Access*, vol. 11, pp. 104 000–104 015, 2023.
- [434] Y. Ye, H. Pei, B. Wang, P.-Y. Chen, Y. Zhu, J. Xiao, and B. Li, “Reinforcement-learning-based portfolio management with augmented asset movement prediction states,” in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1112–1119.
- [435] S. Wang, T. Ji, L. Wang, Y. Sun, S.-C. Liu, A. Kumar, and C.-T. Lu, “StockTime: A time series specialized large language model archi-

- ture for stock price prediction,” *arXiv preprint arXiv:2409.08281*, 2024.
- [436] Z. Xu, Y. Liu, Y. Wang, R. Bao, K. Harimoto, and X. Sun, “Modeling interactions between stocks using LLM-enhanced graphs for volume prediction,” in *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing, the 6th Financial Narrative Processing, and the 1st Workshop on Large Language Models for Finance and Legal*, 2025, pp. 153–163.
- [437] K. J. Koa, Y. Ma, R. Ng, and T.-S. Chua, “Learning to generate explainable stock predictions using self-reflective large language models,” in *Proceedings of the ACM Web Conference 2024*. Association for Computing Machinery, 2024, pp. 4304–4315.
- [438] Y. Abe, S. Matsuo, R. Kondo, and R. Hisano, “Leveraging large language models for institutional portfolio management: Persona-based ensembles,” *arXiv preprint arXiv:2411.19515*, 2024.
- [439] W. Min, B. Mott, J. Rowe, B. Liu, and J. Lester, “Player goal recognition in open-world digital games with long short-term memory networks,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 2590–2596.
- [440] P. Kantharaju and S. Ontañón, “Discovering meaningful labelings for RTS game replays via replay embeddings,” in *Proceedings of the 2020 IEEE Conference on Games*, 2020, pp. 160–167.
- [441] T. Wang, M. Honari-Jahromi, S. Katsarou, O. Mikheeva, T. Panagiotopoulos, S. Asadi, and O. Smirnov, “player2vec: A language modeling approach to understand player behavior in games,” *arXiv preprint arXiv:2404.04234*, 2024.
- [442] T. Wang, M. Honarijahromi, S. Katsarou, O. Mikheeva, T. Panagiotopoulos, O. Smirnov, L. Cao, and S. Asadi, “Understanding players as if they are talking to the game in a customized language: A pilot study,” in *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual*, 2024, pp. 47–52.
- [443] S. Zhao, Y. Xu, Z. Luo, J. Tao, S. Li, C. Fan, and G. Pan, “Player behavior modeling for enhancing role-playing game engagement,” *IEEE Transactions on Computational Social Systems*, vol. 8, no. 2, pp. 464–474, 2021.
- [444] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar, “Minedojo: Building open-ended embodied agents with Internet-scale knowledge,” *arXiv preprint arXiv:2206.08853*, 2022.
- [445] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, “Voyager: An open-ended embodied agent with large language models,” *Transactions on Machine Learning Research*, 2024.
- [446] H. Yuan, C. Zhang, H. Wang, F. Xie, P. Cai, H. Dong, and Z. Lu, “Skill reinforcement learning and planning for open-world long-horizon tasks,” *arXiv preprint arXiv:2303.16563*, 2023.
- [447] R. Volum, S. Rao, M. Xu, G. DesGrennes, C. Brockett, B. Van Durme, O. Deng, A. Malhotra, and B. Dolan, “Craft an iron sword: Dynamically generating interactive game characters by prompting large language models tuned on code,” in *Proceedings of the 3rd Wordplay: When Language Meets Games Workshop*, 2022, pp. 25–43.
- [448] H. Kwak and J. Blackburn, “Linguistic analysis of toxic behavior in an online video game,” in *Social Informatics: SocInfo 2014 International Workshops, Barcelona, Spain, November 11, 2014, Revised Selected Papers*, vol. 8852, 2015, pp. 209–217.
- [449] J. C. Aguerri, M. Santisteban, and F. Miró-Llinares, “The enemy hates best? toxicity in League of Legends and its content moderation implications,” *European Journal on Criminal Policy and Research*, vol. 29, no. 3, pp. 437–456, 2023.
- [450] M. Mårtens, S. Shen, A. Iosup, and F. A. Kuipers, “Toxicity detection in multiplayer online games,” in *Proceedings of the 2015 International Workshop on Network and Systems Support for Games*, 2015, pp. 1–6.
- [451] L. H. X. Ng, A. X. W. Lim, and M. M. Yoder, “Challenges for real-time toxicity detection in online games,” *arXiv preprint arXiv:2407.04383*, 2024.
- [452] J. Morrier, R. Kocielnik, and R. M. Alvarez, “Reinforcement learning for efficient toxicity detection in competitive online video games,” *arXiv preprint arXiv:2503.20968*, 2025.
- [453] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, “Doctor AI: Predicting clinical events via recurrent neural networks,” in *Proceedings of the 1st Machine Learning for Healthcare Conference*, vol. 56, 2016, pp. 301–318.
- [454] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, “Patient subtyping via time-aware LSTM networks,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 65–74.
- [455] B. C. Kwon, M.-J. Choi, J. T. Kim, E. Choi, Y. B. Kim, S. Kwon, J. Sun, and J. Choo, “Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 299–309, 2019.
- [456] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker, “Time2Vec: Learning a vector representation of time,” *arXiv preprint arXiv:1907.05321*, 2019.
- [457] L. Yu, L. Sun, B. Du, and W. Lv, “Towards better dynamic graph learning: New architecture and unified library,” in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 67 686–67 700.
- [458] X. Chen, Y. Tang, J. Xu, J. Zhang, S. Zhang, S. Peng, X. Zheng, and Y. Xiong, “Rethinking time encoding via learnable transformation functions,” in *Proceedings of the 42nd International Conference on Machine Learning*, vol. 267, 2025, pp. 9073–9104.
- [459] Y. Li, N. Du, and S. Bengio, “Time-dependent representation for neural event sequence prediction,” in *6th International Conference on Learning Representations, Workshop Track*, 2018.
- [460] H.-H. Chung, S. S. Chaudhari, X. Han, Y. Wald, S. Sarria, and J. Ghosh, “Between linear and sinusoidal: Rethinking the time encoder in dynamic graph learning,” *Transactions on Machine Learning Research*, 2025.
- [461] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, “Transformer-XL: Attentive language models beyond a fixed-length context,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2978–2988.
- [462] M. Neishi and N. Yoshinaga, “On the relation between position information and sentence length in neural machine translation,” in *Proceedings of the 23rd Conference on Computational Natural Language Learning*, 2019, pp. 328–338.
- [463] B. Wang, D. Zhao, C. Lioma, Q. Li, P. Zhang, and J. G. Simonsen, “Encoding word order in complex embeddings,” in *International Conference on Learning Representations*, 2020, pp. 1–14.
- [464] S. Kiyono, S. Kobayashi, J. Suzuki, and K. Inui, “SHAPE: Shifted absolute position embedding for transformers,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 3309–3321.
- [465] T. Likhomanenko, Q. Xu, G. Synnaeve, R. Collobert, and A. Rogozhnikov, “CAPE: Encoding relative positions with continuous augmented positional embeddings,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 16 079–16 092.
- [466] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 34 892–34 916.
- [467] Z. Huang, D. Liang, P. Xu, and B. Xiang, “Improve transformer models with better relative position embeddings,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 3327–3335.
- [468] K. Sinha, A. Kazemnejad, S. Reddy, J. Pineau, D. Hupkes, and A. Williams, “The curious case of absolute position embeddings,” in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 4449–4472.
- [469] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 464–468.
- [470] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.
- [471] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with disentangled attention,” in *International Conference on Learning Representations*, 2021, pp. 1–20.
- [472] T.-C. Chi, T.-H. Fan, P. J. Ramadge, and A. I. Rudnicky, “KERPLE: Kernelized relative positional embedding for length extrapolation,” in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 8386–8399.
- [473] U. Wennberg and G. E. Henter, “The case for translation-invariant self-attention in transformer-based language models,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021, pp. 130–140.
- [474] C. Wu, F. Wu, and Y. Huang, “DA-transformer: Distance-aware transformer,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 2059–2068.

- [475] O. Press, N. A. Smith, and M. Lewis, "Train short, test long: Attention with linear biases enables input length extrapolation," in *Proceedings of the Tenth International Conference on Learning Representations*, 2022, pp. 1–14.
- [476] J. Su, M. Ahmed, Y. Lu, S. Pan, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.
- [477] B. Bischl, G. Casalicchio, M. Feurer, P. Gijsbers, F. Hutter, M. Lang, R. G. Mantovani, J. N. van Rijn, and J. Vanschoren, "OpenML benchmarking suites," in *Advances in Neural Information Processing Systems*, 2021.
- [478] A. Kadra, M. Lindauer, F. Hutter, and J. Grabocka, "Well-tuned simple nets excel on tabular datasets," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 23 928–23 941.
- [479] D. McElfresh, S. Khandagale, J. Valverde, V. C. Prasad, B. Feuer, C. Hegde, G. Ramakrishnan, M. Goldblum, and C. White, "When do neural nets outperform boosted trees on tabular data?" in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 76 336–76 369.
- [480] C. Ye, G. Lu, H. Wang, L. Li, S. Wu, G. Chen, and J. Zhao, "Towards cross-table masked pretraining for web data mining," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 4449–4459.
- [481] H.-J. Ye, S.-Y. Liu, H.-R. Cai, Q.-L. Zhou, and D.-C. Zhan, "A closer look at deep learning methods on tabular datasets," *arXiv preprint arXiv:2407.00956*, 2024.
- [482] Y. Hou, J. Li, Z. He, A. Yan, X. Chen, and J. McAuley, "Bridging language and items for retrieval and recommendation," *arXiv preprint arXiv:2403.03952*, 2024.
- [483] M. Wan and J. J. McAuley, "Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems," in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining*, 2016, pp. 489–498.
- [484] J. J. McAuley and A. Yang, "Addressing complex and subjective product-related queries with customer reviews," in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 625–635.
- [485] M. Wan, J. Ni, R. Misra, and J. McAuley, "Addressing marketing bias in product recommendations," in *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining*, 2020, pp. 618–626.
- [486] H. Zhu, X. Li, P. Zhang, G. Li, J. He, H. Li, and K. Gai, "Learning tree-based deep model for recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 1079–1088.
- [487] H. Zhu, D. Chang, Z. Xu, P. Zhang, X. Li, J. He, H. Li, J. Xu, and K. Gai, "Joint optimization of tree-based index and deep model for recommender systems," in *Advances in Neural Information [dblp: Joint Optimization of Tree-based Index and Deep Model for ...](https://dblp.org/rec/conf/nips/ZhuCXZLHLXG19)Processing Systems*, vol. 32, 2019, pp. 3973–3982.
- [488] J. Zhuo, Z. Xu, W. Dai, H. Zhu, H. Li, J. Xu, and K. Gai, "Learning optimal tree models under beam search," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 2020, pp. 11 650–11 659.
- [489] J. Li, J. Shang, and J. McAuley, "UCTopic: Unsupervised contrastive learning for phrase representations and topic mining," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 6159–6169.
- [490] A. Yan, Z. He, J. Li, T. Zhang, and J. J. McAuley, "Personalized showcases: Generating multi-modal explanations for recommendations," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 2251–2255.
- [491] J. Rappaz, J. McAuley, and K. Aberer, "Recommendation on live-streaming platforms: Dynamic availability and repeat consumption," in *Proceedings of the 15th ACM Conference on Recommender Systems*, 2021, pp. 390–399.
- [492] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 19:1–19:19, 2015.
- [493] B. P. Majumder, S. Li, J. Ni, and J. McAuley, "Generating personalized recipes from historical user preferences," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 5976–5982.
- [494] J. Ni, L. Muhlstein, and J. McAuley, "Modeling heart rate and activity data for personalized fitness recommendation," in *Proceedings of the 2019 World Wide Web Conference*, 2019, pp. 1343–1353.
- [495] R. He, C. Fang, Z. Wang, and J. J. McAuley, "Vista: A visually, socially, and temporally-aware model for artistic recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 309–316.
- [496] M. Wan and J. J. McAuley, "Item recommendation on monotonic behavior chains," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 86–94.
- [497] A. Pathak, K. Gupta, and J. McAuley, "Generating and personalizing bundle recommendations on Steam," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 1073–1076.
- [498] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
- [499] X. Huang, Y. Yang, Y. Wang, C. Wang, Z. Zhang, J. Xu, L. Chen, and M. Vazirgiannis, "DGraph: A large-scale financial dataset for graph anomaly detection," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.
- [500] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 19:1–19:19, 2015.
- [501] J. Zhuo, Z. Xu, W. Dai, H. Zhu, H. Li, J. Xu, and K. Gai, "Learning optimal tree models under beam search," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 2020, pp. 11 650–11 659.
- [502] S. Huang, F. Pourafaei, J. Danovitch, M. Fey, W. Hu, E. Rossi, J. Leskovec, M. Bronstein, G. Rabusseau, and R. Rabbany, "Temporal graph benchmark for machine learning on temporal graphs," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 2056–2073.
- [503] J. Gastinger, S. Huang, M. Galkin, E. Loghmani, A. Parviz, F. Pourafaei, J. Danovitch, E. Rossi, I. Koutis, H. Stuckenschmidt, R. Rabbany, and G. Rabusseau, "TGB 2.0: A benchmark for learning on temporal knowledge graphs and heterogeneous graphs," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 140 199–140 229.
- [504] B. Rozemberczki, P. Scherer, Y. He, G. Panagopoulos, A. Riedel, M. Astefanoaei, O. Kiss, F. Beres, G. López, N. Collignon, and R. Sarkar, "PyTorch geometric temporal: Spatiotemporal signal processing with neural machine learning models," in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, 2021, pp. 4564–4573.
- [505] Q. Huang, X. Wang, S. X. Rao, Z. Han, Z. Zhang, Y. He, Q. Xu, Y. Zhao, Z. Zheng, and J. Jiang, "Benchtemp: A general benchmark for evaluating temporal graph neural networks," in *Proceedings of the 2024 IEEE 40th International Conference on Data Engineering*, 2024, pp. 4044–4057.
- [506] Y. Yang, H. Zhou, R. Kannan, and V. K. Prasanna, "Towards ideal temporal graph neural networks: Evaluations and conclusions after 10,000 GPU hours," *Proceedings of the VLDB Endowment*, vol. 18, no. 4, pp. 956–969, 2024.
- [507] R. Shirzadkhani, S. Huang, E. Kooshafar, R. Rabbany, and F. Pourafaei, "Temporal graph analysis with TGX," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 1086–1089.
- [508] T. Ma, B. Shi, Y. Xu, Z. Zhao, S. Liang, and B. Dong, "DYGL: A unified benchmark and library for dynamic graph," in *Web and Big Data – 7th International Joint Conference, APWeb-WAIM 2023, Wuhan, China, October 6–8, 2023, Proceedings, Part III*, vol. 14333, 2024, pp. 389–401.
- [509] A. Gravina and D. Bacciu, "Deep learning for dynamic graphs: Models and benchmarks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2024.
- [510] J. Zhang, J. Chen, M. Yang, A. Feng, S. Liang, J. Shao, and R. Ying, "DTGB: A comprehensive benchmark for dynamic text-attributed graphs," in *Advances in Neural Information Processing Systems*, vol. 37, 2024.