

A Survey on Foundation Models for Structured Data: Tabular, Time Series, and Graphs

Qingyun Sun, *Member, IEEE*, Haonan Yuan, Yi Huang, Ziwei Zhang, Xingcheng Fu, Ruijie Wang, Haoyi Zhou, Jia Wu, *Senior Member, IEEE*, Jianxin Li, *Senior Member, IEEE* and Philip S. Yu, *Life Fellow, IEEE*

Abstract—Foundation models have emerged as a dominant paradigm in machine learning, enabling broad generalization and efficient adaptation across diverse tasks and domains. While this paradigm has achieved remarkable success in language and vision data, its extension to structured data remains far less understood. Foundation models for structured data are an emerging yet highly impactful research area with a rapidly growing body of literature. In this survey, we provide a systematic analysis of foundation models for structured data, focusing on tabular, time series, and graph data, covering over 150 representative methods. We analyze the intrinsic properties and inductive biases of structured data, clarify the core concepts of foundation models, and conduct an in-depth analysis of the key challenges that hinder the development of foundation models for structured data. Building on these insights, we organize existing approaches into a coherent taxonomy based on tokenization, architectures, pre-training objectives, and adaptation strategies. Finally, we discuss emerging research directions and open problems, aiming to provide guidance toward more principled and scalable foundation models for structured data.

Index Terms—Foundation Models, Structured Data, Tabular Data, Time Series, Graphs.

I. INTRODUCTION

LARGE language models (LLMs) and vision-language models (VLMs) have achieved remarkable breakthroughs in generalization and task breadth, and are widely seen as a practical path toward more general intelligence [1, 2]. They exemplify foundation models: large models pre-trained on broad, heterogeneous text or vision data and adapted to new tasks via fine-tuning, prompting, or tool use. Their advantages include strong transferability, data efficiency in low-label regimes, unified task interfaces, and scalable maintenance. Structured data underpins high-stakes decision-making across industry and science, powering risk modeling, forecasting, and relational reasoning at scale. Motivated by these trends, foundation models for structured data have emerged as a major research focus in recent years [3].

Q. Sun, H. Yuan, Y. Huang, Z. Zhang, R. Wang, and J. Li are with School of Computer Science and Engineering, Beihang University, Beijing 100191, China. E-mail: {sunqy, yuanhn, yihuang, zwzhang, rjwang, lijx}@buaa.edu.cn.

Z. Zhou is with School of Software, Beihang University, Beijing 100191, China. E-mail: {haoyi}@buaa.edu.cn.

X. Fu is with the Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, China. E-mail: fuxc@gxnu.edu.cn.

J. Wu is with the Department of Computing, Macquarie University, Sydney, NSW 2109, Australia. E-mail: jia.wu@mq.edu.au.

P.S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago 60607, USA. E-mail: psyu@uic.edu.

Manuscript received April 16, 2026.

Structured data broadly refers to data modalities whose semantics are tightly coupled with explicit structural dependencies or constraints. In this survey, we focus on three major types: *tabular*, *time series*, and *graphs*. Foundation models for structured data are inherently more challenging to develop than those for language or vision data. Structured data has distinct structural properties that require task-specific inductive biases: tabular data is row-permutation invariant but column-semantic, time series data requires temporal order awareness and shift invariance, and graph data demands permutation-equivariant relational reasoning. Moreover, structured data is often heterogeneous and constrained, making unified representation and generalization more difficult. As a result, building foundation models for structured data requires explicit modeling of both unified representations and appropriate inductive biases, rather than relying solely on increasing model size and data volume.

Foundation models for structured data remain an emerging yet highly impactful research area. Such foundation models for structured data have already shown strong performance across a wide range of applications. For tabular data, representative works such as TabPFNv2 [4] and LimiX [5] show superior performance by pre-training on large-scale synthetic data or real-world data. For time series data, representative lines of approaches, including Transformer-based models such as TimesFM [6] and LLM-based models such as Chronos [7], can substantially improve performance in forecasting and related tasks across multiple domains. For graph data, recent models such as GCOPE [8] and UniGraph [9] demonstrate strong generalization on molecular, citation, and social graphs.

Differences with existing surveys. Most existing surveys are organized around a single modality, such as tabular [10, 11], time series [12], or graphs [13, 14]. However, this modality-specific perspective makes it hard to transfer insights across domains and to compare design choices in a consistent way, even when the underlying questions are similar, such as how to transform the heterogeneous data into a unified input space, which inductive biases are necessary, what pre-training signals are effective, and how to generalize beyond the training distribution. This survey addresses these gaps by placing tabular data, time series data, and graph data within a unified framework. We first clarify modality-specific characteristics and shared challenges, and identify a set of common principles, then systematically review model families for each modality using consistent dimensions: data tokenization, model architecture, pre-training datasets and objectives, domain transferability, adaptation strategies, and supporting downstream tasks.

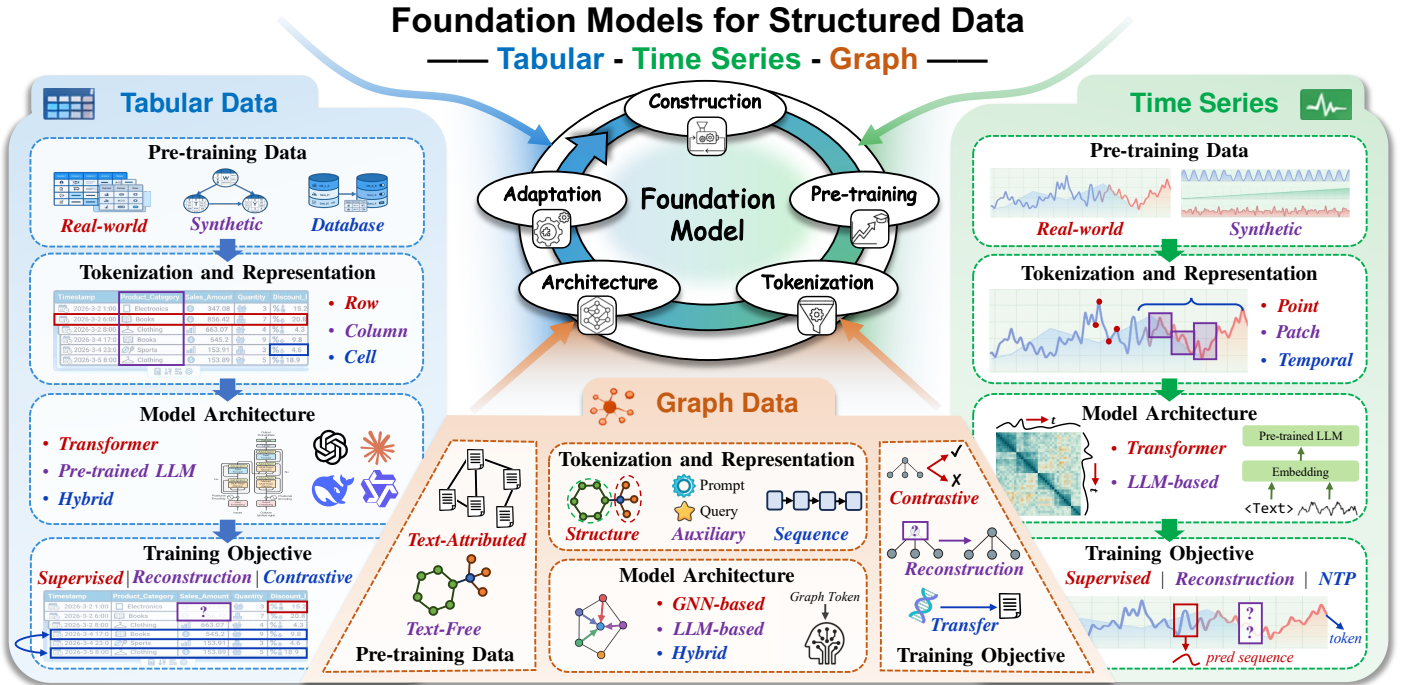


Fig. 1. Overview of the taxonomy for foundation models on structured data. At the top level, we identify a set of common principles underlying foundation models for structured data, including data tokenization, model architecture, pre-training objectives, and adaptation strategies. Building upon these shared dimensions, the framework further organizes existing approaches across three major structured data modalities (tabular, time series, and graph data), highlighting both their common design patterns and modality-specific considerations.

Contributions. This survey provides a systematic and in-depth review of foundation models for structured data, with a particular focus on tabular, time series, and graph data. We identify and analyze the core challenges that hinder the development of foundation models for structured data. We organize existing approaches into a coherent taxonomy based on **tokenization, architecture design, pre-training objectives, and adaptation strategies**. This taxonomy highlights both common design patterns across different structured modalities and modality-specific solutions. We further discuss emerging trends and open challenges, aiming to guide future research toward more principled and scalable foundation models. To facilitate further exploration, we maintain an up-to-date awesome list¹ of relevant papers and resources.

Organization. The content of this paper is organized as follows. In Section II, we introduce the three types of structured data, summarize the characteristics of foundation models, and analyze the challenges of constructing foundation models specifically for structured data. Then we provide the common principles for designing foundation models for structured data in Section III. Subsequently, we systematically introduce and analyze tabular foundation models (Section IV), time series foundation models (Section V), and graph foundation models (Section VI). Finally, we summarize the main challenges and future directions of advancing the foundation model for structured data in Section VII while concluding the survey in Section VIII. The overall organization is shown in Fig. 1.

II. BACKGROUND

In this section, we provide the background for understanding foundation models over structured data. We first analyze the intrinsic properties of structured data and then discuss the core concepts and characteristics of foundation models. Building on these perspectives, we then examine the key challenges that arise when adapting foundation model paradigms to structured data. An overview of these aspects is illustrated in Fig.2.

A. Structured Data: Types, Properties, and Inductive Biases

Although all tabular, time series and graph data are “structured”, they differ in how information is organized and which invariances should be preserved. These differences define the intrinsic properties and key inductive biases that foundation models must encode to learn and generalize.

1) **Tabular data:** Tabular data consists of instances represented as rows and features represented as columns, where each column may correspond to a distinct data type (e.g., numerical, categorical, or ordinal). Tabular data is inherently schema-driven, with intrinsic structure primarily residing at the schema level through feature definitions, data types, and domain constraints. Dominant inductive biases in tabular data include *feature-wise heterogeneity* under a fixed schema, *functional dependencies* between features, and *permutation invariance across instances*. Effective models must capture complex inter-instance relationships while respecting feature heterogeneity.

2) **Time series data:** Time series consist of sequences of observations indexed by time, potentially across multiple

¹<https://github.com/RingBDStack/awesome-structured-data-FM>

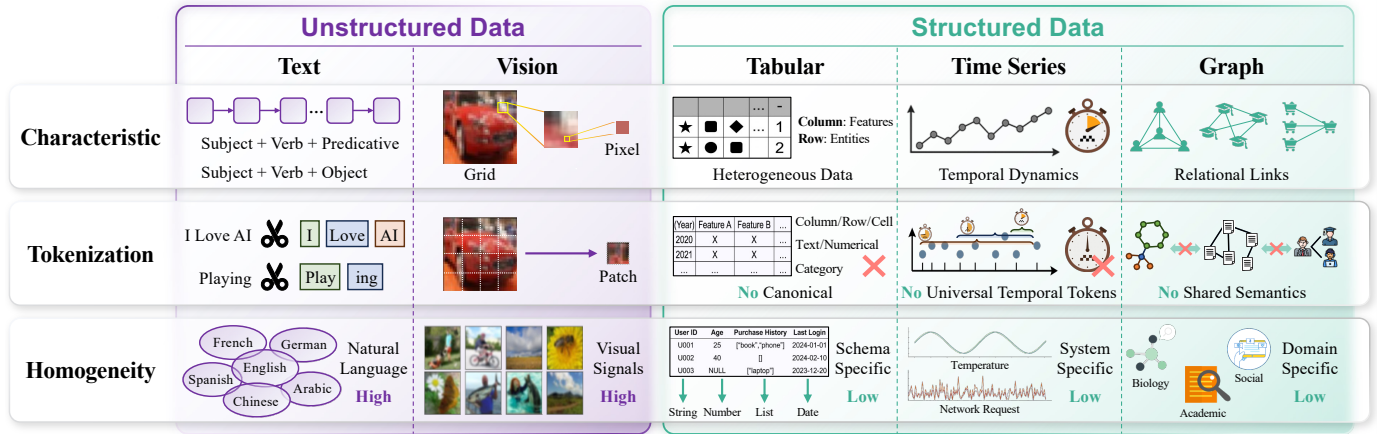


Fig. 2. Unstructured vs. structured data for foundation modeling. The figure compares text and images with tabular, time series, and graph data in terms of data properties, tokenization, and homogeneity. While unstructured data typically has homogeneous representations and canonical tokenization, structured data is heterogeneous, lacks a unified tokenization scheme, and requires modality-specific inductive biases.

variables. Time series data exhibit strong temporal ordering, dynamic dependencies, and often multi-scale patterns, including trends, seasonality, and abrupt changes. The core inductive bias of time series data include *temporal dependency*, *directionality*, *local continuity*, and *causality awareness*, typically assuming that nearby past time points are more strongly correlated with and can influence future ones. Unlike text sequences, temporal order in time series data often reflects underlying physical or causal processes, which complicates the transfer of sequence modeling from language models.

3) **Graph data:** Graph data represents entities as nodes and relationships as edges, forming explicit non-Euclidean structures. The semantics of graph data emerge from structural patterns, including multi-hop connectivity, substructures, and hierarchical and compositional organizations such as communities. The primary inductive biases in graph data include *permutation invariance* over nodes, *relational locality* within neighborhoods, and *higher-order dependencies* among substructures. Effective representations must encode high-order relational patterns and substructure semantics in a permutation-equivariant manner.

B. Foundation Models: Core Concepts and Characteristics

Foundation models are intended to act as a shared base model that can support a wide spectrum of downstream tasks or domains, which have a set of widely recognized characteristics that emphasize generality, reusability, and adaptability.

1) **General-purpose representations:** A central consensus is that foundation models learn reusable and transferable representations. These models are pretrained on large-scale and diverse datasets within a domain or across domains. These representations are typically task-agnostic and capture underlying patterns that generalize beyond narrowly defined tasks. The value of a foundation model lies in the breadth of tasks it enables, rather than achieving the best performance on any single benchmark.

2) **Efficient and robust adaptability:** Foundation models should be adapted to new tasks or data distributions at a

relatively low cost than retraining from scratch, serving as a stable “foundation” that downstream models can build upon. They are expected to exhibit robustness and retain their utility when transferred to new tasks or domains, reflecting the stability and generality of learned representations.

C. Why Structured Data Breaks Conventional Foundation Model Assumptions

The foundation model paradigm has been largely shaped by successes in language and vision, where assumptions such as *data homogeneity*, *canonical data tokenization*, and *universal grounded inductive biases generally hold*. In contrast, structured data systematically violates these assumptions.

1) **Data heterogeneity and inductive biases:** Structured data encompasses a wide range of data characteristics, each associated with distinct and often incompatible inductive biases. Unlike language or images, where relatively stable inductive biases enable the use of generic architectures across datasets, structured data exhibits substantial heterogeneity, with fragmented, dataset-specific biases arising from human-designed schemas, temporal processes, and relational semantics. This variability challenges the assumption that a single architectural paradigm or pre-training strategy can produce broadly transferable representations, limiting the effectiveness of task-agnostic pre-training.

2) **Lack of canonical tokenization:** A key enabler of foundation models in language and vision is that they have well-defined discrete tokens and regular shape, such as words or image patches, which allow data to be mapped into a uniform input space. Structured data, by contrast, lacks such canonical tokenization. Features, time steps, and nodes do not correspond to interchangeable semantic units, and their meanings are often context-dependent. As a result, defining a consistent tokenization scheme that supports scalable pre-training and transfer across datasets remains challenging.

3) **Coupling between structural semantics and tasks:** In structured data, the relevance of features, temporal patterns, or relational substructures is highly task-dependent, and the same

structural pattern may carry different meanings across tasks or domains. This contrasts with unstructured data, where pre-training objectives can often rely on largely task-independent signals. The tight coupling between structural semantics and downstream tasks complicates the design of universal pre-training objectives. As a result, a single pretrained representation is less likely to generalize across diverse tasks without substantial adaptation.

III. COMMON PRINCIPLES

Before diving into foundation models for tabular, time series, and graph data, it is useful to first identify a set of common principles that underpin their design. Although these modalities differ in surface form, they share a fundamental challenge: how to model structured, heterogeneous data while enabling scalable pre-training and transferable adaptation. To this end, we outline five key principles that will serve as a unifying lens throughout the rest of this survey.

A. Structured, Heterogeneous, and Semantically Constrained Data

Despite their differences, tabular, time series, and graph data can all be viewed under a unified paradigm of structured data. In all three cases, semantics are tightly coupled with explicit organization patterns rather than isolated tokens. Tabular data is governed by schema and feature dependencies, time series by temporal order and dynamics, and graphs by relational topology and higher-order connectivity. This shared property implies that foundation models in these domains must go beyond homogeneous token spaces, and instead reason over structured, heterogeneous, and semantically constrained inputs. This perspective motivates many of the design choices discussed in the following sections.

B. Tokenization as Structure-Preserving Abstraction

Unlike text or images, structured data are not naturally in the form of tokens. As a result, tokenization becomes a fundamental modeling step rather than mere preprocessing. Across modalities, the core idea is to transform raw structured inputs into transferable units that can support scalable pre-training. For example, tabular foundation models [15, 16] serialize cells or attribute-value pairs, time series foundation models [6, 17] operate on points or patches, and graph foundation models [18, 19] construct tokens from nodes, subgraphs, or structural components. The underlying principle remains the same: tokenization serves as an abstraction layer that preserves essential structure while making the data compatible with foundation-model-style learning.

C. Data-native vs. LLM-based Architecture Paradigms

Architectural design in structured data foundation models reflects a recurring trade-off between preserving modality-specific inductive biases and leveraging scalable, general-purpose backbones. This leads to three broad design paradigms: (1) data-native encoders tailored to specific structures [4, 20, 21], (2) language-model-based formulations that

recast structured data into sequences [16, 22, 23], and (3) hybrid approaches that combine both [24, 25, 26]. These paradigms highlight a shared design logic: architecture is not chosen solely for representational power, but also for how effectively it balances expressiveness, scalability, and transferability.

D. Pre-training by Learning Generalizable Structural Priors

Another common principle is that pre-training is used to learn reusable priors rather than task-specific heuristics. Across the three modalities, the dominant objectives repeatedly fall into a small number of paradigms, including supervised predictive learning [4, 27], masked or denoising reconstruction [17, 20], and autoregressive learning [7, 28] or self-supervised objectives [8, 9]. Although their implementations differ, these objectives all aim to encode stable regularities that can generalize across datasets, domains, and downstream tasks. This emphasis on transferable structural priors is central to the success of foundation models on structured data.

E. Efficient and Structure-Preserving Adaptation

Finally, structured data foundation models increasingly treat downstream tasks as efficient adaptations of a shared pretrained model, rather than training from scratch. This gives rise to common adaptation strategies such as fine-tuning [29, 6, 30], prompting [31, 32], and in-context learning [4, 33]. Across different modalities, these approaches follow the same high-level principle: adapt pretrained knowledge to new tasks with minimal modification, while preserving as much of the learned structural prior as possible.

IV. TABULAR FOUNDATION MODELS

Tabular Foundation Models (TFMs) aim to learn transferable knowledge from large-scale tabular datasets and generalize across heterogeneous schemas, datasets, and tasks. In the following, we review representative TFMs in terms of *pretraining data*, *objectives*, *tokenization*, *architecture*, and *adaptation strategy*, and summarize them in Table I.

A. Pre-training Data Construction

Existing TFMs adopt diverse strategies to approximate a broad data distribution, which can be broadly categorized into three paradigms: *synthetic data*, *real-world data*, and *high-quality knowledge bases*.

1) **Synthetic data from priors**: One prominent line of work constructs pre-training data by generating synthetic data from carefully designed priors, assuming that tabular datasets share common generative patterns. TabPFNv2 [4] generates datasets from structural causal models (SCMs), while LimiX [5] further extends this approach with hierarchical SCMs for context-conditional masked modeling, where variables are sampled according to directed acyclic graphs with parameterized functional relationships. Subsequent works such as TabICL [33], TabForestPFN [37], and MITRA [34] introduce tree-based priors on top of SCMs, incorporating hierarchical decision boundaries and the non-linear feature interactions characteristic of gradient-boosted trees and random forests.

TABLE I
OVERVIEW OF REPRESENTATIVE TABULAR FOUNDATION MODELS (TFMS).

Method	Pretraining Data	Pre-training Objective & Task	Tokenization	Model Architecture	Adaptation Strategy	Domain Transferability	Downstream Task	Venue
MITRA [34]	synthetic data (SCM, tree-based priors)	classification, regression	cell	Transformer	FT, ICL	1:N	CLS, REG	NeurIPS 2025
UniTabE [29]	real-world tabular datasets	masked cell prediction, row-wise contrastive learning	name-value tuple	Transformer	FT	N:N	CLS, REG	ICLR 2024
CARTE [35]	large knowledge base	contrastive learning of graphlet and truncation pairs	row	Transformer	FT	N:N	CLS, REG	ICML 2024
PORTAL [36]	real-world tabular datasets	masked cell modeling	row	Transformer	FT	N:N	CLS, REG	NeurIPS 2024 Workshop
TabForestPFN [37]	synthetic data (SCM, tree-based priors)	classification	cell	Transformer	FT, ICL	1:N	CLS	arXiv 2024
TabPFNV2 [4]	synthetic data (SCM)	masked cell prediction	cell	Transformer	ICL	1:N	CLS, REG	Nature 2025
TabICL [33]	synthetic data (SCM, tree-based priors)	classification	row	Transformer	ICL	1:N	CLS	ICML 2025
TabDPT [27]	real-world tabular datasets	masked column prediction	row	Transformer	ICL	N:N	CLS, REG	NeurIPS 2025
TabSTAR [38]	real-world tabular datasets	classification, regression	name-value tuple	Transformer	FT	N:N	CLS, REG	NeurIPS 2025
TABULA [39]	real-world tabular datasets	column-wise reconstruction	name-value tuple	Transformer	FT	N:N	IMP	NeurIPS 2025
TARTE [40]	large knowledge base	contrastive learning of entities and facts	name-value tuple	Transformer	FT	N:N	CLS, REG	TMLR 2025
LimiX [5]	synthetic data (SCM)	context-conditional masked modeling	cell	Transformer	ICL	1:N	CLS, REG, IMP, GEN	arXiv 2025
Real-TabPFN [41]	synthetic, real-world tabular datasets	classification	cell	Transformer	ICL	1:N	CLS	arXiv 2025
TabLLM [16]	text	table-to-text generation	name-value tuple	LLM	FT	1:N	CLS	AISTATS 2023
UniPredict [42]	real-world tabular datasets	table-to-text generation	name-value tuple	LLM	IT	1:N	CLS, REG	arXiv 2023
TP-BERTa [15]	real-world tabular datasets	CLS, REG	name-value tuple	LLM	FT	N:N	CLS, REG	ICLR 2024
TABULA-8B [43]	real-world tabular datasets	tabular prediction	row	LLM	ICL	1:N	CLS, REG	NeurIPS 2024
IngesTables [24]	real-world tabular datasets	attention-based tabular modeling	name-value tuple	Transformer + LLM	FT	N:N	CLS, REG	NeurIPS 2023 Workshop

Adaptation Strategy: “FT” denotes “Fine-Tuning”, “ICL” denotes “In-Context Learning”.

Downstream Task: “CLS” denotes “Classification”, “REG” denotes “Regression”, “IMP” denotes “Imputation”, and “GEN” denotes “Generation”.

2) **Real-world data across diverse domains:** An alternative paradigm relies on large collections of real-world tabular datasets from multiple domains, aiming to learn cross-dataset statistical regularities (e.g., feature co-occurrence, missingness structures, and label distributions). Representative domains include biology, finance, industry, and medicine, *etc.* Models such as TabDPT [27], TP-BERTa [15], and UniTabE [29] are pretrained on curated corpora comprising datasets with diverse feature spaces, scales, and semantics. TabSTAR [38] further highlights the inclusion of datasets with rich textual attributes, creating a more semantically enriched foundation model. Real-TabPFN [41] extends the synthetic-prior pretraining of TabPFNV2 [4] by continued pre-training on real-world datasets, demonstrating that adaptation to real data can further strengthen a PFN-style model.

3) **Large knowledge base:** CARTE [35] and TARTE [40] construct pre-training data from large-scale knowledge bases, where tables encode factual information about the real world. These knowledge-base-derived tables prioritize factual consistency and semantic alignment across tables, effectively

grounding tabular representations in world knowledge. In these methods, the pretraining corpus is not collected as ordinary tabular datasets, but derived from knowledge graphs or relational facts that are transformed into table-compatible structures.

B. Pre-training Objectives and Tasks

Current pre-training tasks include *supervised downstream tasks*, *masked reconstruction*, and *contrastive learning*.

1) **Supervised tasks:** A widely adopted paradigm directly adopts supervised tasks (e.g., classification and regression) as pre-training objectives. Representative works include TabICL [33], MITRA [34], and TabForestPFN [37], which directly optimize supervised prediction tasks during pretraining. In contrast, PFN-style methods such as TabPFNV2 learn to make predictions through synthetic task distributions and masked target inference in an in-context learning framework. This approach relies on the availability of labeled datasets and may bias the learned representations toward the specific task distributions seen during training.

2) **Masked reconstruction**: Another line of work formulates pretraining as a masked modeling problem. UniTabE [29] and PORTAL [36] predict masked cell contents from the remaining context, encouraging the encoder to capture feature dependencies and heterogeneous value semantics. LimiX [5] further develops a context-conditional masked modeling objective under hierarchical SCMs. Unlike standard masked reconstruction methods, TabPFNv2 [4] masks target labels and predicts them through in-context prediction. TabDPT [27] proposes pre-training by masked column prediction, which imposes a stronger requirement to infer high-level column semantics from cross-feature context.

3) **Contrastive learning**: Contrastive learning is an alternative self-supervised objective for pre-training. UniTabE [29] adopts row-wise contrastive learning, encouraging instance-level consistency under feature corruption. CARTE [35] applies contrastive learning on graphlet and truncation pairs, treating subgraphs extracted from tables semantically aligned views. TARTE [40] performs contrastive learning over entity-fact relational structures, aligning entity representations with their factual contexts derived from knowledge bases.

C. Data Tokenization and Representation

Data tokenization and representation determine the basic computational units and levels of semantic granularity. Existing approaches can be broadly categorized into three paradigms: *cell*, *row*, and *name-value tuple*.

1) **Cell-level tokenization**: Cell-level tokenization treats each individual cell value as a basic token, typically augmented with feature-type embeddings. Representative methods include TabPFNv2 [4], MITRA [34], and LimiX [5]. In MITRA [34], cell-level tokenization mainly corresponds to its 2D element-wise variant, while the method also supports a 1D row-wise formulation. This representation provides fine-grained access to raw feature values, allowing models to directly capture local dependencies among entries within a row. It is especially well-suited for reconstruction-based pre-training objectives.

2) **Row-level tokenization**: Row-level tokenization abstracts an entire row into a compact embedding, focusing on instance-level semantics rather than individual feature values. Representative methods include TabICL [33], TabDPT [27], and PORTAL [36]. In particular, PORTAL [36] follows a row-wise processing scheme while using content-specific encodings for text, numerical, and date cells. This scheme encourages models to achieve permutation invariance over columns.

3) **Column name-value tuple**: This paradigm represents tabular data as a set of column name-value tuples, explicitly pairing each feature value with its semantic identifier. Representative methods include TabSTAR [38], TP-BERTa [15], IngesTables [24], and UniPredict [42]. This scheme aligns tabular modeling with language-based representations, facilitating better generalization across heterogeneous schemas.

D. Model Architecture

Most existing TFMs are built upon Transformer architectures, owing to their flexibility in modeling feature interactions

and compatibility with large-scale pretraining. Some works further enhance pre-trained language models with Transformer for richer semantic representation.

1) **Transformer-based architecture**: A representative line of work, exemplified by TabPFNv2 [4], introduces the concept of Prior-Data Fitted Networks (PFNs) into tabular learning. It employs a two-way attention mechanism—comprising feature attention and sample attention, making each cell to attend both to other features in its row and the same feature across its column. Building on this foundation, subsequent methods further refine the Transformer architecture. TabICL [33] proposes a column-then-row sparse attention across all cells by leveraging the column or row inherent structure as a strong inductive bias, thereby enhancing both scalability and efficiency. MITRA [34] adopts both row-based attention and element-based attention across rows and columns. Beyond standard row and column attention, CARTE [35] adapts the Transformer encoder into a graph attentional network over graphlets derived from knowledge bases, extending tabular modeling to relational structures.

2) **Pre-trained language model based architecture**: Pre-trained language models (PLMs) are used to transform tables into a textual or token-based form. TP-BERTa [15] introduces relative-magnitude-based numerical tokenization, mapping numerical values into discrete tokens that can be processed by a RoBERTa-based language model. TabLLM [16] serializes feature names and values into natural-language strings and adapts a pretrained LLM for tabular classification, often with parameter-efficient fine-tuning. TABULA-8B [43] adapts Llama 3-8B for tabular prediction by treating serialized rows as language modeling sequences, emphasizing zero-shot and few-shot generalization without target-task fine-tuning. UniPredict [42] fine-tunes an LLM on multiple datasets with diverse targets by instruction-conditioned generative tabular prediction and target augmentation. Both approaches leverage the semantic priors encoded in PLMs, particularly for datasets involving rich textual information or complex feature semantics.

3) **Hybrid architecture**: It integrates language-based semantic encoders with tabular-specific interaction modules. For example, TARTE [40] combines semantic encoders for column-aware value representations with a Transformer backbone for table modeling. IngesTables [24] encodes feature semantics using cached LLM embeddings and then applies a MapTransformer to model interactions among key-value tuples, enabling transfer across heterogeneous schemas.

E. Adaptation Strategy

Typical adaptation strategies for TFMs include in-context learning, prompt-based adaptation, and fine-tuning.

1) **In-Context Learning**: In-Context Learning (ICL) sets training data as context for testing and predicts in a single forward pass without updating parameters. TabPFNv2 [4] performs ICL with tables, training across datasets and applying it to the entire test dataset rather than to individual samples. TabICL [33] employs a two-stage ICL, which first encodes rows and then combines these embeddings with corresponding

labels for ICL, significantly improving the scalability of ICL for tables. TabDPT [27] performs the ICL-based retrieval by selecting only the top-K most similar training rows as context at inference time. Besides PFN-style models, recent methods such as MITRA [34] and LimiX [5] also adopt in-context prediction as a primary adaptation interface.

2) **Prompt-based adaptation:** TabLLM [16] prompts the LLM with both a natural-language string representing the serialization of a row and a short task-specific prompt. UniPredict [42] prompts by integrating pre-processed meta-data, tabular samples, and task instructions into text inputs for LLMs.

3) **Fine-tuning:** TabSTAR [38] fine-tunes the pre-trained model using Low-Rank Adaptation (LoRA) [44] on a single downstream task. CARTE [35] involves two fine-tuning settings: single-table inference with power-transformed numerical features and bagging, and transfer learning where the model is jointly fine-tuned on source and target tables with outcome alignment and bagging. UniTabE [29] treats the target as a masked column for prediction in classification/regression tasks, or uses task-specific prompts with instructions and questions for reasoning tasks such as Table QA. TABULA [39] follows a pretraining-then-finetuning paradigm for downstream single-cell tasks, including gene imputation, perturbation prediction, and cell-level prediction under federated settings.

F. Recent Advances and Applications

1) **Trustworthiness:** The trustworthiness of TFMs has garnered rising attention, encompassing aspects such as robustness, privacy, attacks, and defense strategies. RTFM [45] proposes a model-agnostic adversarial training framework for TFMs, aiming to make the model adapt to more challenging data regions. Another work [46] proposes test-time attacks and in-context defenses for TFMs, highlighting the importance of trustworthiness in real-world applications.

2) **Applications:** TABULA [39] integrates federated learning into a tabular foundation model for single-cell transcriptomics, combining gene reconstruction and cell-level contrastive objectives under privacy-aware, distributed pretraining.

Saito et al. [47] apply TabPFN as a zero-training tabular foundation model to geotechnical site characterization tasks, including spatial prediction and missing-parameter imputation. It outperforms a hierarchical Bayesian model in accuracy and uncertainty estimation, while achieving faster or more efficient inference.

G. Datasets and Benchmarks

Tabular data is ubiquitous across various domains and real-world applications. Consequently, numerous existing works have focused on curating datasets and benchmarks for TFMs. UniTabE [29] collects data from Kaggle of a 7TB corpus containing 13 billion tabular examples. OpenTabs [48] provides a dataset sourced from the web, including approximately 46 million tabular samples. TabArena [49] manually curates 51 small to medium-sized tabular IID datasets, and evaluates 16 tabular machine learning models alongside 3 TFMs with

a public leaderboard². TabularFM [50] provides an open-source framework³ for constructing and benchmarking TFMs. A recent work [51] proposes to benchmark the privacy leakage of TFMs in synthetic tabular data generation.

V. TIME SERIES FOUNDATION MODELS

Time Series Foundation Models (TSFMs) aim to learn transferable temporal knowledge from large-scale time series and generalize across heterogeneous domains and tasks. In the following, we review representative TSFMs in terms of *pre-training data, objectives, tokenization, architecture, adaptation strategy*, and summarize them in Table II.

A. Pre-training Data Construction

1) **Real-world datasets:** Pre-training data construction primarily relies on large collections of real-world datasets spanning multiple application domains and reflecting the inherent heterogeneity of real-world time series [54]. Representative domains include finance, energy systems, traffic, climate, and others. The selected time series data exhibit substantial variability in temporal resolution, amplitude scale, dominant frequencies, and degrees of stationarity, posing significant challenges for learning transferable representations. CloudOps [67] proposes introducing large-scale time series forecasting datasets from the cloud operations domain, comprising billions of observations, thereby enabling the study of the scaling of TSFMs.

2) **Synthetic data:** A growing line of work explores synthetic time series as a complementary strategy. ForecastFPN [52] explores pre-training on synthetic time series generated from structured priors that include periodic components, global trends, and stochastic noise. Such synthetic data provide a controllable way to enrich pre-training distributions and expose the model to diverse temporal patterns.

B. Pre-training Objectives and Tasks

Pre-training tasks for time series foundation models are primarily designed to capture temporal dependencies. Existing approaches can be broadly grouped into three categories: *supervised predictive tasks, masked reconstruction*, and *next token prediction*.

1) **Supervised predictive tasks:** Most TSFMs that are pretrained directly on time series corpora adopt forecasting-oriented objectives. For example, TimesFM [6] is trained with a point forecasting objective, while Time-MoE [57] further extends this paradigm with multi-resolution forecasting. MOIRAI [55] optimizes probabilistic forecasting objectives over heterogeneous datasets, while Chronos [7] formulates pre-training as autoregressive density estimation over discretized time-series tokens. In contrast, TEMPO [25] is better viewed as a prompt-based adaptation framework built on a pretrained GPT backbone, rather than a representative example of large-scale time-series pretraining.

²<https://tabarena.ai>

³<https://tabularfm.github.io/>

TABLE II
OVERVIEW OF REPRESENTATIVE TIME SERIES FOUNDATION MODELS (TSFMS).

Method	Pretraining Data	Pre-training Objective & Task	Tokenization	Model Architecture	Adaptation Strategy	Domain Transferability	Downstream Task	Venue
ForecastFPN [52]	synthetic data (periodicity)	point forecasting	point	Transformer	/	1:N	FCT	NeurIPS 2023
Lag-Llama [53]	real-world time series datasets	probabilistic forecasting	lag-feature vector	Transformer	/	N:N	FCT	NeurIPS 2023 Workshop
TimeGPT-1 [54]	real-world time series datasets	forecasting	sliding window	Transformer	FT	N:N	FCT	arXiv 2023
UniTime [20]	real-world time series datasets	point forecasting, reconstruction	patch with fixed length	Transformer	ICL	N:N	FCT	WWW 2024
TimesFM [6]	synthetic data and real-world time series datasets	point forecasting	patch with fixed length	Transformer	FT	N:N	FCT	ICML 2024
MOMENT [17]	real-world time series datasets	masked reconstruction	patch with fixed length	Transformer	FT	N:N	FCT, CLS, IMP, AD	ICML 2024
MOIRAI [55]	real-world time series datasets	probabilistic forecasting	patch with adaptive length	Transformer	ICL	N:N	FCT	ICML 2024
Timer [28]	real-world time series datasets	next token prediction	patch with fixed length	Transformer	/	N:N	FCT, IMP, AD	ICML 2024
UNITS [56]	real-world time series datasets	masked reconstruction	patch with fixed length	Transformer	PL	N:N	FCT, CLS, IMP, AD	NeurIPS 2024
Time-MoE [57]	real-world time series datasets	multi-resolution forecasting	point	Transformer	FT, ICL	N:N	FCT	ICLR 2025
WaveToken [58]	real-world time series datasets	next token prediction	wavelet	Transformer	ICL	N:N	FCT	ICML 2025
ROSE [59]	real-world time series datasets	masked reconstruction	patch with fixed length	Transformer	FT	N:N	FCT	ICML 2025
GPT4TS [60]	/	/	patch with fixed length	LLM	FT	1:N	FCT, CLS, IMP, AD	NeurIPS 2023
LLMTime [22]	/	/	point, strings of digits	LLM	/	1:N	FCT	NeurIPS 2023
PromptCast [31]	/	/	point, strings of digits	LLM	FT	1:N	FCT	TKDE 2023
GPT4MTS [61]	real-world large event datasets	multimodal forecasting	patch with reversible instance normalization	LLM	PL	1:1	FCT	AAAI 2024
TIME-LLM [62]	/	/	patch with fixed length	LLM	PL	1:N	FCT	ICLR 2024
AutoTimes [63]	real-world time series datasets	next token prediction	patch with fixed length	LLM	PL, ICL	N:N	FCT	NeurIPS 2024
Chronos [7]	real-world time series datasets and synthetic data	autoregressive density estimation	quantization	LLM	/	N:N	FCT	TMLR 2024
CALF [64]	/	/	text and time series embedding	LLM	FT	N:N	FCT	AAAI 2025
LLM4TS [65]	/	autoregressive time-series alignment	patch with fixed length	LLM	FT	1:N	FCT	TIST 2025
LLM-Mixer [66]	/	/	text and time series embedding	LLM	FT	1:N	FCT	ACL 2025 Workshop
TEMPO [25]	/	point forecasting	patch with fixed length	Transformer + LLM	PL	N:N	FCT	ICLR 2024

Adaptation Strategy: “FT” denotes “Fine-Tuning”, “PL” denotes “Prompt Learning”, “ICL” denotes “In-Context Learning”.

Downstream Task: “CLS” denotes “Classification”, “FCT” denotes “Forecasting”, “IMP” denotes “Imputation”, and “AD” denotes “Anomaly Detection”.

2) *Masked reconstruction*: Inspired by masked modeling, segments or tokens of a time series are masked, and the model is trained to reconstruct the missing parts from the observed context. Representative methods such as MOMENT [17] and UNITS [56] show that masked reconstruction can effectively encourage the model to learn contextual temporal representations from partially observed inputs. UniTime [20] is also related to this family, but it combines reconstruction and forecasting objectives within a language-empowered cross-domain framework guided by domain instructions. ROSE [59] further extends masked modeling to the frequency domain through multi-frequency masked reconstruction, aiming to learn more transferable temporal patterns.

3) *Next token prediction*: Similar to the pre-training objective of language models, the model learns to predict the next token in an autoregressive manner. Representative methods include Timer [28], AutoTimes [63], and WaveToken [58]. This approach leverages sequential token prediction to model long-range dependencies and complex temporal dynamics, facilitating architectural reuse and scalable pre-training.

C. Data Tokenization and Representation

Tokenization for time series defines the temporal granularity at which patterns are modeled. It also determines the balance between expressiveness and computational efficiency. Existing approaches can be broadly categorized into three paradigms:

point-level, *patch-level*, and *temporal feature representation*. These different tokenization strategies reflect a trade-off between temporal fidelity, abstraction level, and inductive bias.

1) **Point-level tokenization**: The most fine-grained strategy treats individual points as tokens, directly modeling raw sequences. This approach preserves temporal resolution and is adopted by methods such as Time-MoE [57] and ForecastFPN [52], making it well-suited for capturing precise temporal alignments. PromptCast [31] further serializes numerical observations into textual prompts, thereby bridging point-wise values with language-style sequence modeling. However, the resulting tokens pose challenges for scalability and modeling long-range dependencies.

2) **Patch-level tokenization**: A more prevalent approach aggregates consecutive time points into patches that summarize local temporal contexts. Models such as TimesFM [6], MOMENT [17], and UniTime [20] adopt this strategy, significantly reducing sequence length while enabling the model to capture higher-level temporal patterns. MOIRAI [55] extends this paradigm by introducing patches with adaptive lengths, allowing dynamic adjustments for varying frequencies.

3) **Temporal feature representation**: This approach derives temporal features rather than raw values for tokenization. WaveToken [58] employs wavelets to encode frequency-localized information, explicitly capturing multi-scale temporal patterns. Similarly, Lag-Llama [53] transforms sequences into lag feature vectors, emphasizing autoregressive dependencies and temporal offsets. By explicitly embedding temporal features into the tokenization process, these methods incorporate strong inductive biases that complement data-driven learning.

D. Model Architecture

Architectures are largely shaped by advances in sequence modeling, featuring two dominant paradigms: *Transformer-based architectures* and *LLM-based architectures*.

1) **Transformer-based architecture**: Transformer-based TSFMs typically adapt the standard Transformer backbone to handle heterogeneous temporal resolutions, variable context lengths, and diverse forecasting horizons. A common design uses patch-based tokenization followed by sequence modeling with Transformer encoders, decoders, or encoder-decoder variants, as in TimesFM [6], MOMENT [17], UniTime [20], MOIRAI [55], and Timer [28]. Some methods further introduce specialized architectural mechanisms to improve scalability or inductive bias. For example, Time-MoE [57] leverages a sparse mixture-of-experts design to increase model capacity while maintaining computational efficiency, whereas ROSE [59] incorporates frequency-aware masked modeling for more transferable temporal representation learning.

2) **LLM-based architecture**: LLM-based architectures reinterpret time series modeling through the lens of language modeling, but differ substantially in how temporal signals are mapped into language-model-compatible inputs. Other methods directly serialize numerical observations into textual sequences, as in PromptCast [31] and LLTime [22]. Some methods align time-series segments or patches with

pretrained language-model backbones, such as GPT4TS [60], TIME-LLM [62], and AutoTimes [63]. Another line of work discretizes continuous values into finite vocabularies and performs language-model-style prediction over quantized tokens, as exemplified by Chronos [7]. For example, LLTime [22] encodes numbers as text and samples possible extrapolations as text completions, whereas TIME-LLM [62] reprograms time-series patches into text prototype representations and augments them with textual prompts.

E. Adaptation Strategy

1) **Direct inference**: Some TSFMs can be used directly for inference on downstream datasets without introducing additional task-specific fine-tuning or prompt optimization. This paradigm is particularly common in zero-shot forecasters which aim to generalize across unseen datasets while preserving the original generative mechanism of the pretrained model. Representative examples include ForecastPFN [52], Lag-Llama [53], LLTime [22], and Chronos [7]. Among them, Chronos [7] first scales and quantizes continuous observations into discrete tokens, and then autoregressively samples token sequences that are mapped back to numerical forecasts.

2) **Prompt-based adaptation**: It treats pre-trained time series models as frozen backbone learners and adapts them to downstream tasks by designing task-specific prompts, either as learnable tokens, textual instructions, or contextual sequences. TEMPO [25] proposes a soft prompt strategy that incorporates decomposed trend, seasonal, and residual information to better accommodate the dynamic nature of time series. UNITS [56] learns prompt tokens from the time series and adapts to new tasks by utilizing these tokens while keeping the pre-trained model weights frozen. TIME-LLM [62] combines prompting with input reprogramming: it first maps time-series patches into text prototype representations and then uses a Prompt-as-Prefix strategy to enrich the input with task-aware textual context. AutoTimes [63] performs lightweight adaptation of a frozen decoder-only LLM under a next-token-prediction objective, and further proposes an in-context forecasting mechanism in which relevant historical time series are used as prompts. Similar to machine translation, PromptCast [31] adopts both input and output prompts, where the input prompt covers the historical observations and prediction targets, and the output prompt handles the desired prediction values for training or evaluation.

3) **Fine-tuning**: Fine-tuning-based adaptation directly updates a subset or all parameters of pretrained TSFMs on downstream data. MOMENT [17] can either be fine-tuned end-to-end or adapted via linear probing, where only the reconstruction or forecasting head parameters are updated while keeping the backbone frozen. GPT4TS [60] adapts a pretrained GPT-style backbone to time series by freezing most self-attention and feed-forward parameters, while updating a small subset of components such as positional embeddings and layer-normalization-related parameters during fine-tuning. LLM4TS [65] employs two typical parameter-efficient fine-tuning methods, including Layer Normalization Tuning [68] and LoRA [44]. ROSE [59] first obtains domain-specific register parameters through pre-training, then applies

a Top-K selection strategy during fine-tuning to supplement downstream data information. To align LLMs for time series forecasting, CALF [64] utilizes three cross-modal fine-tuning techniques including the Cross-Modal Match Module, the Feature Regularization Loss, and the Output Consistency Loss to fine-tune the temporal target branch.

F. Domain and Task Transferability

1) **Domain transferability:** Domain transferability is a central goal of TSFMs, as many recent models are explicitly designed to generalize across datasets with different temporal resolutions, scales, and semantic domains. Representative N:N transfer settings can be found in models such as TimesFM [6], MOIRAI [55], MOMENT [17], Chronos [7], and UniTime [20]. Among them, UniTime [20] proposes a cross-domain learning framework that leverages natural-language domain instructions to encode domain-specific information and facilitate zero-shot transfer to unseen domains. ROSE [59] takes a different approach by introducing a time-series register that stores domain-specific information during pre-training and enables adaptive target-domain transfer through downstream fine-tuning with register selection.

2) **Task transferability:** Task transferability remains a central criterion for evaluating TSFMs. To date, the majority of existing models are primarily designed for forecasting, reflecting the dominance of prediction as the canonical task in temporal modeling. Representative models include TimesFM [6], Time-MoE [57], Chronos [7], MOIRAI [55], and others. In contrast, only a smaller subset of TSFMs explicitly targets broader task generalization, including forecasting, classification, imputation, and anomaly detection. Representative models include MOMENT [17], UNITS [56], GPT4TS [60], and Timer [28]. Although the exact task coverage differs across methods, these models collectively extend TSFMs beyond pure forecasting to broader tasks such as classification, imputation, and anomaly detection. Such efforts move TSFMs closer to a stronger notion of foundation models by pursuing more task-agnostic temporal representations, although consistently transferable performance across heterogeneous time-series tasks remains an open challenge.

G. Recent Advances and Applications

Recent advances in TSFMs can be broadly discussed from the perspectives of *vision transfer*, *multimodal fusion*, *retrieval augmentation*, and *interactive systems*.

1) **Vision models:** A recent trend is to reuse pretrained vision models for time series analysis by converting temporal signals into image-like representations. For example, VisionTS [69] reformulates forecasting as an image reconstruction task, while VisionTS++ [70] further explores cross-modal time series foundation modeling.

2) **Vision-time fusion:** Another direction studies the fusion of time series with visual information. Representative methods include Time-VLM [71], which combines temporal, visual, and textual signals using pretrained vision-language models, and VISTA [72], which leverages visual representations for training-free stock forecasting.

3) **Retrieval augmentation:** Inspired by RAG in LLMs, many recent methods integrate TSFMs with retrieval mechanisms to improve forecasting on unseen patterns. Representative examples include TimeRAG [73], TS-RAG [74], RAG4CTS [75], TimeRAF [76], and Cross-RAG [77]. Beyond single models, Ravuru et al. [78] further propose an agentic RAG framework with a hierarchical multi-agent architecture for time series analysis.

4) **Interactive systems:** Recent work also begins to develop interactive systems around TSFM-related analysis. For example, TSGAssist [79] combines TSGBench [80], LLMs, and RAG to provide an interactive assistant for time-series generation recommendation and benchmarking.

H. Datasets, Benchmarks and Frameworks

Recent efforts on TSFMs have introduced not only large-scale pre-training corpora, but also dedicated benchmarks and toolboxes for systematic evaluation. TSFM-Bench [81] evaluates TSFMs in multiple forecasting settings, including zero-shot, few-shot, and full-shot. LTSM-Bundle [82] provides a benchmark and a toolbox on LLMs for time series forecasting. Some recent works propose benchmarking TSFMs on forecasting tasks in specific applications, such as household electricity load forecasting [83], oil production forecasting [84], and probabilistic electricity price forecasting [85]. Benchmarks for comprehensively evaluating temporal foundation models across multiple tasks (*e.g.*, classification, imputation, anomaly detection) remain scarce.

VI. GRAPH FOUNDATION MODELS

Graph Foundation Models (GFMs) aim to learn transferable knowledge from large-scale graph data and generalize across heterogeneous domains and tasks. In the following, we review representative GFMs in terms of *pretraining data*, *objectives*, *tokenization*, *architecture*, *adaptation strategy*, and *transferability*, and summarize them in Table III.

A. Pre-training Data Construction

A central issue in GFMs concerns not only the source of pre-training data, but also how such data is constructed to support knowledge transfer. Existing studies can be broadly grouped into three categories: text-attributed graphs, text-free graphs, and others. Their differences lie primarily in whether semantic information is available in explicit textual form, how cross-domain commonalities is established, and what types of transferable priors can be induced during pre-training.

1) **Text-attributed graphs (TAGs):** Text-attributed graphs are graphs whose nodes or edges carry explicit textual descriptions, or can be transformed into text through textualization. Typical examples include citation and academic networks, where nodes are associated with titles, abstracts, or documents. The key property lies in their explicit semantic information encoded in natural language, making them well-suited for graph-language modeling. In practice, methods use such data in different ways: OFA [19] and UniGraph [9] mainly exploit text to unify heterogeneous tasks, GraphGPT [114]

TABLE III
OVERVIEW OF REPRESENTATIVE GRAPH FOUNDATION MODELS (GFMS).

Method	Pretraining Data	Pre-training Objective & Task	Tokenization	Model Architecture	Adaptation Strategy	Domain Transferability	Downstream Task	Venue
GraphPrompt [32]	text-free	subgraph similarity	subgraph	GNN	PL	1:1	NC, GC	WWW 2023
HGPrompt [86]	text-free	subgraph similarity	subgraph	GNN	PL	1:1	NC, GC	AAAI 2024
GCOPE [8]	text-free	contrastive pretraining, feature reconstruction	node	GNN	FT, PL	N:N	NC	KDD 2024
MultiGPrompt [87]	text-free	subgraph similarity	encoder layer	GNN	PL	1:1	NC, GC	WWW 2024
OpenGraph [88]	text-free	masked autoencoding	node	GNN	/	N:N	NC, LP	EMNLP 2024
GFT [89]	text-attributed	tree reconstruction	computation tree	GNN	FT	N:N	NC, GC, LP	NeurIPS 2024
AnyGraph [30]	text-free	link prediction	node	GNN	FT	1:N	NC, GC, LP	arXiv 2024
MDGPT [90]	text-free	subgraph similarity	domain	GNN	PL	N:N	NC, GC	arXiv 2024
OMOG [91]	text-attributed	contrastive pretraining	node	GNN	/	N:N	NC, LP	arXiv 2024
GraphMoRE [92]	text-free	topology heterogeneity modeling	node	GNN	FT	1:1	NC, LP	AAAI 2025
GraphAny [93]	text-free	node classification	node	GNN	/	1:N	NC	ICLR 2025
GraphPrompter [94]	text-free	neighbor matching, subgraph reconstruction	subgraph	GNN	ICL	N:N	NC, GC, LP	ICDE 2025
BRIDGE [21]	text-free	subgraph similarity	aligner	GNN	PL	N:N	NC, GC	ICML 2025
GIT [95]	text-attributed	tree reconstruction	task tree	GNN	FT, IT, ICL	N:N	NC, GC, LP	ICML 2025
MDGFM [96]	text-free	subgraph similarity	domain	GNN	PL	N:N	NC	ICML 2025
AutoGFM [97]	text-attributed	disentangled contrastive representation learning	subgraph	GNN	FT	N:N	NC, GC, LP	ICML 2025
GCoT [98]	text-free	link prediction	node	GNN	PL	1:1	NC, GC	KDD 2025
PatchNet [99]	text-free	attribute masking context prediction	node patch	GNN	FT	N:N	NC, GC	KDD 2025
SAMGPT [100]	text-free	subgraph similarity	structure token	GNN	PL	N:N	NC, GC	WWW 2025
UniGraph2 [101]	multimodal	reconstruction	node	GNN	/	N:N	Multimodal Tasks	WWW 2025
RiemannGFM [102]	text-attributed, text-free	geometric contrastive learning	subgraph	GNN	FT	N:N	NC, LP	WWW 2025
GraphCLIP [103]	text-attributed	contrastive learning invariant alignment	subgraph	GNN	PL	N:N	NC, LP	WWW 2025
UniPrompt [104]	text-free	/	prompt graph	GNN	PL	N:N	NC	NeurIPS 2025
GraphKeeper [105]	text-free	continual graph pretraining	node	GNN	FT	N:1	NC, GC	NeurIPS 2025
H ² GFM [106]	text-attributed	text-space node encoding, context-path modeling	node	GNN	/	N:N	NC, LP	arXiv 2025
RWPT [107]	text-attributed	contrastive pretraining	node sequence	GNN	FT	N:N	NC, GC, LP	arXiv 2025
MDGCL [108]	text-free	contrastive pretraining	subgraph	GNN	FT	N:1	NC, GC	arXiv 2025
GILT [109]	text-free	few-shot meta-pretraining	node, edge, graph	GNN	ICL	N:N	NC, GC, LP	arXiv 2025
GMoPE [110]	text-free	contrastive pretraining	node	GNN	FT	N:N	NC, GC, LP	arXiv 2025
GraphGlue [111]	text-free	geometric pretraining	manifold patch	GNN	FT, PL	N:1	NC, GC, LP	ICLR 2026
LLaGA [23]	text-attributed	alignment tuning	node sequence	LLM	IT	N:N	NC, LP	ICML 2024
LangGFM [112]	text-attributed, text-free	instruction tuning	text	LLM	IT, ICL	N:N	NC, GC, LP	arXiv 2024
PromptGFM [113]	text-attributed	multi-task instruction tuning	node	LLM	IT	N:N	NC, LP	arXiv 2025
OFA [19]	text-attributed	graph classification	subgraph	GNN + LLM	PL, ICL	N:N	NC, GC, LP	ICLR 2024
GraphGPT [114]	text-attributed	contrastive alignment, graph matching	subgraph	GNN + LLM	IT	N:N	NC, LP	SIGIR 2024
ZeroG [115]	text-attributed	semantic similarity	prompting node, subgraph	GNN + LLM	/	N:N	NC	KDD 2024
GOFA [116]	text-attributed	generative modeling	node, edge	GNN + LLM	IT	N:N	NC, GC, LP	ICLR 2025
UniGraph [9]	text-attributed	text reconstruction	node, subgraph	GNN + LLM	IT, ICL	N:N	NC, GC, LP	KDD 2025
BooG [117]	text-attributed	super-node/class-hypothesis matching	subgraph	GNN + LLM	FT	N:N	NC, GC, LP	FCS 2025
GRAVER [18]	text-attributed	subgraph similarity	subgraph	GNN + LLM	PL	N:N	NC, GC	NeurIPS 2025
SA ² GFM [26]	text-attributed	subgraph similarity	node, structural entropy	GNN + LLM	PL	N:N	NC, GC	AAAI 2026

Adaptation Strategy: “FT” denotes “Fine-Tuning”, “PL” denotes “Prompt Learning”, “IT” denotes “Instruction Tuning”, “ICL” denotes “In-Context Learning”.

Downstream Task: “NC” denotes “Node Classification”, “GC” denotes “Graph Classification”, “LP” denotes “Link Prediction”.

and GraphCLIP [103] use it for graph-text alignment, while GOFA [116] further structures textual information for graph-aware generation. SA²GFM [26] leverages text via structure-aware prompts derived from entropy-based encoding trees.

Hyper-FM [118] extends this setting to text-attributed hypergraphs, where textual vertex attributes are coupled with higher-order relations for multi-domain foundation modeling.

2) **Text-free graphs**: Text-free graphs do not contain explicit textual information, though they may include non-textual attributes like numerical features derived from raw data. In this setting, pre-training data construction primarily emphasizes structural diversity, feature heterogeneity, and cross-domain regularity, since transferable priors must be learned from topology and non-textual attributes rather than language. Correspondingly, methods such as GraphPrompt [32] and UniPrompt [104] typically organize such data through subgraphs or prompt-based structures, whereas SAMGPT [100], BRIDGE [21], and other multi-domain pre-trained GFMs use heterogeneous graph dataset collections to expose transferable patterns. Nonetheless, graphs inherently exhibit non-Euclidean structures with complex interactions among their entities, making traditional Euclidean space difficult for capturing their rich structural complexity. In contrast, Riemannian manifolds provide a principled framework for modeling such structures, offering expressive tools for measuring graph similarity and quantifying structural connectivity—tasks that are often challenging in Euclidean space. Recent research has begun to explore GFMs through the lens of Riemannian graph learning [92, 102, 111, 119, 120] For example, GraphMoRE [92] further captures such patterns via local topology characterization and personalized mixed-curvature embeddings. GBN [119] develops a unified interpretation of MPNNs from the lens of Riemannian geometry with local bottleneck adjustment. CRGFM [120] describes the graphs by the mixture of curvature-based geometric experts.

3) **Others**: Recent work has expanded pre-training data construction by reformulating the data representation itself. One direction explores multimodal graphs, where graph elements may carry text, images, or other modalities. UniGraph2 [101] is representative of this setting. Another recasts graphs into backbone-compatible interfaces, such as language-readable formats or tabular representations. For example, LangGFM [112] textualizes graphs into standard data formats like JSON and XML, while G2T-FM [121] converts graphs into tables for tabular foundation models. These settings suggest that pre-training data construction in GFMs is increasingly a problem of how to reformulate graph data to expose transferable priors under different foundation model backbones.

B. Pre-training Objectives and Tasks

Pre-training objectives define the transferable priors and therefore shape the scope of their cross-domain and cross-task generalization. Compared with conventional graph pre-training, recent GFMs adopt substantially broader training objectives. They no longer focus solely on task-specific prediction but increasingly emphasize cross-domain alignment, structure reconstruction, unified task formulation, and the reuse of inherited pre-training signals. Existing methods can be broadly grouped into three categories: contrastive and alignment-based objectives, reconstruction and generative objectives, and objective-agnostic pre-training.

1) **Contrastive and alignment**: A major group of methods formulates pre-training around representation alignment, with the objective evolving from cross-domain feature consistency to cross-modal and geometric alignment.

Recent multi-domain methods, such as SAMGPT [100], GCOPE [8], MDGCL [108], MDGFM [96], BRIDGE [21], and GRAVER [18], primarily employ contrastive learning to align features across domains, thereby inducing transferable invariances. Subsequent work extends alignment beyond the representation space: GraphCLIP [103] aligns subgraphs with LLM-generated summaries to inject semantic consistency, while RiemannGFM [102] performs contrastive learning across hyperbolic and hyperspherical views to encode geometric consistency. Overall, this line broadens the notion of invariance in GFM pre-training from feature alignment to semantic and geometric alignment.

2) **Reconstruction and generative**: Another line of GFMs defines pre-training with reconstruction, evolving from masked recovery to graph generation. Earlier methods, including UniGraph [9], GFT [89], GIT [95], UniGraph2 [101], OpenGraph [88], and RWPT [107], mainly reconstruct masked structure, attributes, semantics, or unified task representations. Among these methods, GFT [89] is notable for learning transferable computation trees through joint recovery of features, topology, and semantics. Subsequent work extends this line toward generative modeling, where GraphGPT [114] introduces graph-text grounded prediction, LangGFM [112] casts graph understanding into the language generation space, and GOFA [116] unifies completion, structural understanding, question answering, and retrieval within a single graph generative framework. In summary, this line of research evolves from reconstruction-based pre-training to structured graph generation.

3) **Objective-agnostic**: A smaller but important line of work does not introduce new pre-training objectives. Instead, it improves cross-domain knowledge transfer by more effectively leveraging existing pretrained models and objectives. UniPrompt [104] is the most representative work, as it adapts encoders pretrained by methods such as DGI [122], GRACE [123], GraphMAE [124], or FUG [125], without introducing an independent objective of its own. GMoPE [110] follows a similar philosophy by remaining compatible with multiple upstream objectives, while DCGFM [126] enhances existing GFM pipelines through data-centric pruning under the original backbone objective. G2T-FM [121] further extends this idea by inheriting priors from pretrained tabular foundation models rather than relying on a graph-specific objective. Overall, this line of research shifts the emphasis from objective design to objective reuse and reorganization.

C. Data Tokenization and Representation

Data tokenization and representation determine how raw graph data is converted into transferable units. This design choice is particularly important, as it defines the granularity and format in which structure, semantics, and cross-domain commonalities are encoded. Existing methods can be broadly organized into three categories: graph-structure tokens, auxiliary abstract tokens, and language or sequence tokens.

1) **Graph-structure tokens**: A major line of work tokenizes graphs using native structural units, with the design evolving from individual nodes to contextual subgraphs, and further

to more specialized graph structures. Early methods, such as GraphPrompt [32] and UniGraph [9], mainly use contextual subgraphs as unified task instances. Later work diversifies the tokenization granularity: GFT [89] and GIT [95] elevate computation trees into transferable units, GRAVER [18] adaptively routes neighbor nodes into transferable subgraph vocabularies, RWPT [107] encodes rooted random-walk contexts, PatchNet [99] constructs learnable graph patches, GraphGPT [114] and LLaGA [23] utilize node-centered structural contexts, and RiemannGFM [102] together with H²GFM [106] further extend tokenization to geometric or higher-order relational structures. Overall, this line of research evolves from subgraph-based decomposition to more structured graph-structure tokenization.

2) **Auxiliary abstract tokens:** A second group of methods introduces tokens that are not direct graph components, but rather abstract carriers of task and domain information. Early work, exemplified by OFA [19], uses prompt nodes and class nodes to unify different tasks. Subsequent studies incorporate more explicit domain- and transfer-aware abstractions: SAMGPT [100] uses structure tokens, MDGPT [90] and MDGCL [108] use domain tokens, BRIDGE [21] and MDGFM [96] encode transferable domain priors through aligned abstract tokens. More recent methods shift toward adaptation-oriented abstractions, where GMoPE [110] uses expert prompts, GILT [109] uses support-query tokens, BooG [117] adds class-conditioned super nodes, and GraphPrompter [94] builds task graphs with prompt subgraphs and label nodes. Overall, this line demonstrates that tokenization in GFMs extends beyond graph priors that facilitate cross-domain generalization and adaptation.

3) **Language and sequence tokens:** A third line moves away from explicit graph-native tokens, instead representing graphs through language-like, sequence-like, or otherwise reformulated inputs. Early studies in this direction employ continuous graph encodings: OpenGraph [88] leverages topology-aware projections, AnyGraph [30] learns unified node-level continuous representations, UniPrompt [104] and GCoT [98] utilize prompt-enhanced continuous features, and OMOG [91] organizes such representations for source-model selection. Later work makes the reformulation more explicit: LangGFM [112] textualizes graphs into language-readable formats, PromptGFM [113] maps graph elements into native language tokens, and G2T-FM [121] converts graphs into tabular feature vectors. Overall, this line reflects a shift from graph-specific continuous encoding toward reformulated inputs compatible with general-purpose backbones, enabling broader transfer and cross-modal integration.

D. Model Architecture

Model architecture defines the choice of the base model that performs the main computation in GFMs. Existing methods can be broadly grouped into three classes: GNN-based architectures, LLM-based architectures, and GNN-LLM hybrid architectures. The distinction reflects whether graph generalization is primarily achieved through graph-native message passing, large language models, or a synergistic integration of both.

1) **GNN-based architecture:** Most GFMs remain built on graph-native backbones, where GNNs, graph transformers, or their variants serve as the primary base model. Early methods, such as GraphPrompt [32] and HGPrompt [86], mainly adopt conventional GNN encoders, while MultiGPrompt [87] and UniPrompt [104] extend this paradigm to more unified adaptation settings. Later studies strengthen the backbone with domain-aware encoding or alignment mechanisms, as in SAMGPT [100], GCOPE [8], MDGPT [90], BRIDGE [21], MDGFM [96], and MDGCL [108]. Recent work introduces more specialized architectures: GMoPE [110] incorporates expert routing, GRAVER [18] uses a CoE-MoE network for hierarchical vocabulary routing, GraphGlue [111], RiemannGFM [102], GBN [119], CRGFM [120], and AutoGFM [97] further extend the backbone with geometric modeling and its architecture customization. Overall, this line develops from standard GNN encoders to more specialized graph-native architectures.

2) **LLM-based architecture:** Another line builds GFMs directly on large language models, with the LLM itself serving as the primary base model rather than a text encoder. LangGFM [112] is representative in that it performs graph understanding entirely in language space with an LLM backbone. PromptGFM [113] follows a similar paradigm by using LLMs for graph prediction over language-based graph inputs. More broadly, this line reflects a shift from graph-native modeling to foundation backbones whose primary inductive bias derives from large-scale language or general representation learning.

3) **GNN-LLM hybrid architecture:** A third group of works combines graph encoders with language encoders. Early methods, such as OFA [19] and UniGraph [9], leverage both encoders for attribute alignment and task unification. Later studies enhance this interaction in various ways: GraphGPT [114] and LLaGA [23] couple graph encoders with LLMs through projection layers, GraphCLIP [103] and BooG [117] combine graph encoding with text-level supervision, and UniGraph2 [101] further extends the hybrid paradigm to multimodal settings. More recent models move toward tighter fusion, where GOfA [116] interleaves GNN computation with a frozen LLM, while DCGFM [126] strengthens such hybrid backbones from the data side. Overall, this line evolves from loose graph-language coupling to more integrated hybrid architectures.

E. Adaptation Strategy

Adaptation strategy determines how pretrained priors are transferred to downstream tasks and domains. Recent GFMs adopt broad spectrum of adaptation paradigms, ranging from prompt-based adaptation, lightweight fine-tuning, instruction- or in-context adaptation, and even direct zero-shot transfer.

1) **Prompt-based:** Prompt-based GFMs leverage prompts as the primary adaptation mechanism, ranging from task-specific prompts to domain-aware prompts, and expert-aware prompts. Representative works include GraphPrompt [32], HGPrompt [86], MultiGPrompt [87], and UniPrompt [104], which employ learnable prompts to adapt pretrained models under unified task templates. Later studies extend

this approach to multi-domain settings: SAMGPT [100], MDGPT [90], BRIDGE [21], and GRAVER [18] introduce structural or domain prompts, MDGFM [96] further combines meta prompts with task-specific prompts, and GMoPE [110] incorporates expert prompts with MoE routing. More recent methods adopt richer prompting interfaces: GCoT [98] introduces thought-conditioned prompts, while GraphPrompter [94] enables tuning-free in-context prompting via prompt subgraphs and task graphs.

2) **Fine-tuning-based:** Another line retains fine-tuning as the main adaptation route, but increasingly constrains it to lightweight forms. Early graph-native methods such as GFT [89] and GCOPE [8] mainly rely on downstream fine-tuning over pretrained representations or token vocabularies. Subsequent work makes fine-tuning more transfer-oriented: BRIDGE [21] introduces spectral-regularized fine-tuning after MoE-based source selection, RiemannGFM [102] performs cross-domain fine-tuning in a geometric latent space, and PatchNet [99] adapts pretrained patch-based encoders to downstream tasks. GraphKeeper [105] further extends this line to graph domain-incremental learning by introducing domain-specific graph PEFT together with deviation-free knowledge preservation. Related variants include BooG [117], which supports frozen-encoder transfer with a lightweight MLP head, and AutoGFM [97], which combines architecture customization with downstream adaptation. The overall trend is a shift from conventional end-to-end fine-tuning toward more parameter-efficient and structurally constrained transfer.

3) **Instruction- and in-context adaptation:** A third line adapts GFMs through language-style instruction following or support-query reasoning, reflecting the growing influence of LLM paradigms. Early examples such as OFA [19] already expose a unified prompted interface that supports few-shot and zero-shot transfer. Subsequently, methods deepen this direction in different ways: GraphGPT [114] uses dual-stage instruction tuning, LangGFM [112] adopts LoRA-based instruction tuning in pure language space, and GOFA [116] combines large-scale self-supervised pre-training with downstream instruction fine-tuning. In parallel, GILT [109] formulates adaptation as tuning-free in-context learning over support and query tokens, while UniGraph [9] and GraphPrompter [94] also move toward in-context transfer under graph-specific interfaces. Overall, this line evolves from prompted task unification to instruction-following and in-context adaptation.

4) **Adaptation-free strategies:** A smaller but increasingly important line minimizes or removes downstream adaptation altogether, aiming for direct transfer from pretrained graph priors. OpenGraph [88] and ZeroG [115] emphasize zero-shot transfer to unseen graphs without target-domain fine-tuning, GraphAny [93] performs fully inductive transfer without additional training, and OMOG [91] relies on adaptive model selection and weighted fusion rather than parameter updates. Related interface-driven variants such as G2T-FM [121] also reduce adaptation to in-context inference or minimal task-side tuning. Rather than optimizing task-specific parameters, these methods treat adaptation primarily as selection, retrieval, or direct inference over pretrained representations.

F. Recent Advances and Applications

Recent studies extend GFMs beyond generic graph pre-training toward four directions: *reasoning*, *federated learning*, *trustworthiness*, and *applications in biomedicine*.

1) **Reasoning:** GFM-based reasoning has evolved from in-context reasoning over structured knowledge to tighter integration with retrieval. KG-ICL [127] provides an early foundation for unified KG reasoning. GFM-RAG [128] extends this line to GraphRAG, using pretrained GFMs as graph retrievers for multi-hop reasoning. G-Reasoner [129] further integrates GFMs and LLMs for more general graph reasoning. RAG-GFM [130] then reverses the direction by using retrieval to enhance GFMs themselves, while GFM-Retriever [131] further refines this paradigm through query-specific subgraph retrieval. Overall, this line moves from prompt-based graph reasoning toward retrieval-aware graph foundation modeling, combining structured knowledge, pretraining, and adaptive retrieval for more general and scalable reasoning.

2) **Federated Learning:** The integration of GFMs with federated learning has progressed from paradigm formulation to more structured collaboration. FedGFM [132] establishes federated graph foundation modeling and identifies knowledge entanglement as a central challenge. FedBook [133] advances this line via codebook-based knowledge aggregation, while FedGALA [134] further develops the approach toward continuous graph-language alignment, incorporating communication-efficient prompt tuning. This line evolves from basic federated pre-training to more structured and semantically aligned federated GFMs.

3) **Trustworthiness:** Research on trustworthy GFMs has progressed from open-world reliability to security and privacy auditing. AnomalyGFM [135] and GLIP-OOD [136] extend GFMs to anomaly and OOD detection. MEA-GFM [137], DTGBA [138], and HeTa [139] expose information extraction and backdoor vulnerabilities. More recent work considers stronger threat models and specialized settings, including GFM-BA [140], MGP-MIA [141], GFM4GA [142], and CyberGFM [143].

4) **Biomedicine:** The application of GFMs to biomedicine has progressed from general biomedical knowledge graphs to more specialized imaging, neuroscience, and cellular modeling. TxGNN [144] provides an early foundation for zero-shot drug repurposing over medical knowledge graphs. GraphMSR [145] extends this line to MRI super-resolution, BrainGFM [146] further specializes it to brain foundation modeling, and CellAwareGNN [147] incorporates single-cell genomic context for finer-grained drug indication prediction.

G. Datasets, Benchmarks and Frameworks

As GFMs expand across domains and tasks, datasets, benchmarks, and frameworks have emerged as an important research dimension in their own right. TSGFM [148] represents an early benchmark for text-space GFMs, curating more than 20 datasets and evaluating them under unified settings that distinguish co-training from pre-training and task-specific from cross-task transfer. GFMBench [149] extends

this line from benchmark construction to framework standardization, providing a modular open-source pipeline for preprocessing, training, evaluation, and real-world deployment across diverse GFMs. Building on these developments, GFM-Benchmark [150] further formalizes graph domain shift as a two-dimensional problem spanning both topic domains and format domains, and establishes controlled protocols for cross-topic, cross-format, and joint transfer based on 33 datasets spanning seven topics and six formats. Collectively, these studies indicate a clear progression from dataset curation to framework standardization, and then to a more fine-grained benchmark design for graph domain shift.

VII. CHALLENGES AND FUTURE DIRECTIONS

Despite encouraging progress, foundation models for structured data remain at an early stage. Several fundamental questions are still open, including how these models scale with data and computation, whether a unified modeling paradigm can be developed across diverse structured data modalities, and how they should be effectively integrated with LLMs, VLMs, and emerging intelligent systems. In this section, we discuss these key challenges and outline promising future directions toward more generalizable, scalable, and practically impactful structured data foundation models.

A. Scaling Laws for Structured Data Foundation Models

Beyond architectural and methodological advances, an important yet under-explored direction is to systematically understand the learning dynamics of foundation models for structured data. This includes investigating how these models scale, how knowledge is acquired during training, and how it can be effectively evaluated.

Scaling laws have been empirically validated in LLMs and VLMs [151, 152]. A pivotal question remains: whether such scaling laws also apply to structured data. Unlike text and vision, structured data exhibits strong schema dependence, heterogeneous feature semantics, and explicit relational or temporal constraints, which may fundamentally alter how performance scales with increased model size and data volume. Previous works [27, 153, 154] investigate neural scaling laws in terms of data scale and model scale. Establishing scaling principles that account for structural complexity and inductive bias is crucial and remains under-explored for guiding foundation model design. Beyond scaling behavior, another promising direction is to study emergent learning phenomena in structured data models, such as the presence of “grokking” or sudden generalization (*i.e.*, phase transitions in performance during training). Identifying, characterizing, and quantitatively measuring such phenomena could provide deeper insights into how structured knowledge is internalized. Furthermore, it is important to develop principled approaches for knowledge assessment in both pre-training and fine-tuning data. This includes understanding what types of structural patterns, dependencies, and domain knowledge are captured during training, how they transfer across tasks and domains, and how to evaluate data quality and coverage in a structured-data-specific manner.

B. Unified Foundation Models for Structured Data

Current foundation models for tabular, time-series, and graph data are largely modality-specific in terms of model architectures, pre-training objectives, and adaptation strategies. A key open problem is how to construct a unified foundation model that accommodates diverse structural inductive biases while sharing common representation and reasoning mechanisms. Recent works attempt to bridge the gap between these three types of data, such as leveraging tabular foundation models to tackle graph tasks [121, 155, 156] and leveraging graph foundation models for time series analysis [157].

Progress in this direction requires structure-aware abstractions, such as unified tokenization, modular architectures, and pre-training objectives that capture shared structural principles. Concretely, unified tokenization should provide a flexible interface that can encode heterogeneous entities (*e.g.*, rows, timestamps, nodes, edges) and their relations within a common token space, while preserving modality-specific semantics. Modular architectures may enable the composition of interchangeable components (*e.g.*, temporal encoders, relational reasoning modules, and feature interaction layers), allowing the model to dynamically adapt to different structural patterns without redesigning the entire architecture. Meanwhile, pre-training objectives should go beyond modality-specific tasks (*e.g.*, masked value prediction or link prediction) and instead focus on capturing universal structural properties, such as dependency modeling, invariance under structural transformations, and cross-entity reasoning.

C. Integration with LLMs, VLMs, and Emerging Agentic Systems

Another important direction is understanding how structured data foundation models should interact with LLMs, VLMs, and emerging agentic and embodied systems within unified intelligent ecosystems. Current approaches typically convert structured data into natural language or serialized formats for LLM processing [15, 22, 19], but such reductions often lose explicit structural constraints and relational or temporal inductive biases.

Instead, future systems should support principled, bidirectional collaboration across modalities. LLMs and VLMs can provide semantic understanding and perception, while structured data foundation models focus on relational reasoning, constraint-aware inference, and temporal modeling. This requires shared intermediate representations that preserve structure while enabling cross-modal interoperability. This need becomes more critical in agentic and embodied settings, where models operate in dynamic environments and interact through memory, knowledge graphs, and sensor streams. Here, structured data models can serve as a reasoning backbone for state tracking, planning, and tool coordination, while LLMs and VLMs act as perception and interaction interfaces. Building such systems requires moving beyond simple serialization toward unified representation and coordination mechanisms, enabling scalable integration of semantic understanding, structural reasoning, and embodied decision-making in future agentic AI systems.

VIII. CONCLUSION

This survey principally introduces the existing foundation models for structured data, focusing on three typical data types: tabular, time series, and graphs. We connect the intrinsic characteristics of structured data with the core assumptions underlying foundation models, identifying the challenges faced by current foundation models for structured data. To organize the rapidly growing body of literature, we propose a systematic taxonomy of existing foundation models for structured data, categorizing them by tokenization, model architectures, training and pre-training paradigms, and adaptation strategies. Based on this synthesis, we discuss open challenges and promising future directions for foundation models on structured data.

ACKNOWLEDGMENT

The corresponding author is Jianxin Li. This work is supported by the NSFC through grants No.62225202 and No.62302023.

REFERENCES

- [1] S. Minaee *et al.*, “Large language models: A survey,” *arXiv preprint arXiv:2402.06196*, 2024.
- [2] Z. Li *et al.*, “A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges,” *arXiv preprint arXiv:2501.02189*, 2025.
- [3] B. Van Breugel and M. Van Der Schaar, “Position: Why tabular foundation models should be a research priority,” *arXiv preprint arXiv:2405.01147*, 2024.
- [4] N. Hollmann *et al.*, “Accurate predictions on small data with a tabular foundation model,” *Nature*, vol. 637, no. 8045, pp. 319–326, 2025.
- [5] X. Zhang *et al.*, “Limix: Unleashing structured-data modeling capability for generalist intelligence,” *arXiv preprint arXiv:2509.03505*, 2025.
- [6] A. Das *et al.*, “A decoder-only foundation model for time-series forecasting,” in *ICML*, 2024.
- [7] A. F. Ansari, L. Stella, A. C. Turkmen *et al.*, “Chronos: Learning the language of time series,” *TMLR*, 2024.
- [8] H. Zhao *et al.*, “All in one and one for all: A simple yet effective method towards cross-domain graph pre-training,” in *KDD*, 2024, pp. 4443–4454.
- [9] Y. He *et al.*, “Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs,” in *KDD*, 2025, pp. 448–459.
- [10] S. Somvanshi *et al.*, “A survey on deep tabular learning,” *arXiv preprint arXiv:2410.12034*, 2024.
- [11] J. Ye *et al.*, “A survey of time series foundation models: Generalizing time series representation with large language model,” *arXiv preprint arXiv:2405.02358*, 2024.
- [12] Y. Liang *et al.*, “Foundation models for time series analysis: A tutorial and survey,” in *KDD*, 2024, pp. 6555–6565.
- [13] Z. Wang *et al.*, “Graph foundation models: A comprehensive survey,” *arXiv preprint arXiv:2505.15116*, 2025.
- [14] J. Liu *et al.*, “Graph foundation models: Concepts, opportunities and challenges,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [15] J. o. Yan, “Making pre-trained language models great on tabular prediction,” in *ICLR*, 2024.
- [16] S. Hegselmann *et al.*, “Tabllm: Few-shot classification of tabular data with large language models,” in *AIS-TATS*, 2023, pp. 5549–5581.
- [17] M. Goswami *et al.*, “Moment: A family of open time-series foundation models,” in *ICML*, 2024.
- [18] H. Yuan *et al.*, “Graver: Generative graph vocabularies for robust graph foundation models fine-tuning,” in *NeurIPS*, 2025.
- [19] H. Liu *et al.*, “One for all: Towards training one graph model for all classification tasks,” in *ICLR*, 2024.
- [20] X. Liu *et al.*, “Unitime: A language-empowered unified model for cross-domain time series forecasting,” in *WWW*, 2024, pp. 4095–4106.
- [21] H. Yuan *et al.*, “How much can transfer? bridge: Bounded multi-domain graph foundation model with generalization guarantees,” in *ICML*, 2025.
- [22] N. Gruver *et al.*, “Large language models are zero-shot time series forecasters,” *NeurIPS*, vol. 36, pp. 19 622–19 635, 2023.
- [23] R. Chen *et al.*, “Llaga: Large language and graph assistant,” *arXiv preprint arXiv:2402.08170*, 2024.
- [24] S. Yak *et al.*, “Ingestables: scalable and efficient training of llm-enabled tabular foundation models,” in *TRL Workshop in NeurIPS*, 2023.
- [25] D. Cao *et al.*, “Tempo: Prompt-based generative pre-trained transformer for time series forecasting,” in *ICLR*, 2024.
- [26] J. Shi *et al.*, “Sa²gfm: Enhancing robust graph foundation models with structure-aware semantic augmentation,” in *AAAI*, 2026.
- [27] J. Ma *et al.*, “Tabdpt: Scaling tabular foundation models on real data,” in *NeurIPS*, 2025.
- [28] Y. Liu *et al.*, “Timer: Generative pre-trained transformers are large time series models,” in *ICML*, 2024, pp. 32 369–32 399.
- [29] Y. Yang *et al.*, “Unitabe: A universal pretraining protocol for tabular foundation model in data science,” in *ICLR*, 2024.
- [30] L. Xia and C. Huang, “Anygraph: Graph foundation model in the wild,” *arXiv preprint arXiv:2408.10700*, 2024.
- [31] H. Xue and F. D. Salim, “Promptcast: A new prompt-based learning paradigm for time series forecasting,” *IEEE TKDE*, vol. 36, no. 11, pp. 6851–6864, 2023.
- [32] Z. Liu *et al.*, “Graphprompt: Unifying pre-training and downstream tasks for graph neural networks,” in *WWW*, 2023, pp. 417–428.
- [33] Q. Jingang *et al.*, “Tabicl: A tabular foundation model for in-context learning on large data,” in *ICML*, 2025.
- [34] X. o. Zhang, “Mitra: Mixed synthetic priors for enhancing tabular foundation models,” in *NeurIPS*, 2025.
- [35] M. J. Kim *et al.*, “Carte: Pretraining and transfer for tabular learning,” in *ICML*, 2024, pp. 23 843–23 866.

- [36] M. Spinaci *et al.*, “Portal: Scalable tabular foundation models via content-specific tokenization,” *arXiv preprint arXiv:2410.13516*, 2024.
- [37] F. d. Breejen *et al.*, “Fine-tuned in-context learning transformers are excellent tabular data classifiers,” *arXiv preprint arXiv:2405.13396*, 2024.
- [38] A. Arazi *et al.*, “Tabstar: A tabular foundation model for tabular data with text fields,” in *NeurIPS*, 2025.
- [39] J. Ding *et al.*, “Tabula: A tabular self-supervised foundation model for single-cell transcriptomics,” in *NeurIPS*, 2025.
- [40] M. J. Kim *et al.*, “Table foundation models: on knowledge pre-training for tabular learning,” *TMLR*, 2025.
- [41] A. Garg *et al.*, “Real-tabpfn: Improving tabular foundation models via continued pre-training with real-world data,” *arXiv preprint arXiv:2507.03971*, 2025.
- [42] R. Wang *et al.*, “Unipredict: Large language models are universal tabular classifiers,” *arXiv preprint arXiv:2310.03266*, 2023.
- [43] J. Gardner *et al.*, “Large scale transfer learning for tabular data via language modeling,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 45 155–45 205, 2024.
- [44] E. J. Hu *et al.*, “Lora: Low-rank adaptation of large language models.” *Iclr*, vol. 1, no. 2, p. 3, 2022.
- [45] M. Peroni *et al.*, “Robust tabular foundation models,” *arXiv preprint arXiv:2512.03307*, 2025.
- [46] M. Djilani *et al.*, “On the robustness of tabular foundation models: Test-time attacks and in-context defenses,” *arXiv preprint arXiv:2506.02978*, 2025.
- [47] T. Saito *et al.*, “Applying a tabular foundation model to geotechnical site characterization,” *Geodata and AI*, p. 100040, 2025.
- [48] C. Ye *et al.*, “Towards cross-table masked pretraining for web data mining,” in *WWW*, 2024, pp. 4449–4459.
- [49] N. Erickson *et al.*, “Tabarena: A living benchmark for machine learning on tabular data,” in *NeurIPS*, 2025.
- [50] Q. M. Tran *et al.*, “Tabularfm: An open framework for tabular foundational models,” in *IEEE BigData*, 2024, pp. 1694–1699.
- [51] J. Byun *et al.*, “Risk in context: Benchmarking privacy leakage of foundation models in synthetic tabular data generation,” *arXiv preprint arXiv:2507.17066*, 2025.
- [52] S. Dooley *et al.*, “Forecastpfn: Synthetically-trained zero-shot forecasting,” *NeurIPS*, vol. 36, pp. 2403–2426, 2023.
- [53] K. Rasul *et al.*, “Lag-llama: Towards foundation models for time series forecasting,” in *R0-FoMo Workshop at NeurIPS*, 2023.
- [54] A. Garza *et al.*, “Timegpt-1,” *arXiv preprint arXiv:2310.03589*, 2023.
- [55] G. Woo *et al.*, “Unified training of universal time series forecasting transformers,” in *ICML*, 2024, pp. 53 140–53 164.
- [56] S. Gao *et al.*, “Units: A unified multi-task time series model,” *NeurIPS*, vol. 37, pp. 140 589–140 631, 2024.
- [57] S. Xiaoming *et al.*, “Time-moe: Billion-scale time series foundation models with mixture of experts,” in *ICLR*, 2025.
- [58] L. Masserano *et al.*, “Enhancing foundation models for time series forecasting via wavelet-based tokenization,” *arXiv preprint arXiv:2412.05244*, 2024.
- [59] Y. Wang *et al.*, “Towards a general time series forecasting model with unified representation and adaptive transfer,” in *ICML*, 2025.
- [60] T. Zhou *et al.*, “One fits all: Power general time series analysis by pretrained lm,” *NeurIPS*, vol. 36, pp. 43 322–43 355, 2023.
- [61] F. Jia *et al.*, “Gpt4mts: Prompt-based large language model for multimodal time-series forecasting,” in *AAAI*, vol. 38, no. 21, 2024, pp. 23 343–23 351.
- [62] M. Jin *et al.*, “Time-llm: Time series forecasting by reprogramming large language models,” in *ICLR*, 2024.
- [63] Y. Liu *et al.*, “Autotimes: Autoregressive time series forecasters via large language models,” *NeurIPS*, vol. 37, pp. 122 154–122 184, 2024.
- [64] P. Liu *et al.*, “Calf: Aligning llms for time series forecasting via cross-modal fine-tuning,” in *AAAI*, vol. 39, no. 18, 2025, pp. 18 915–18 923.
- [65] C. Chang *et al.*, “Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters,” *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 3, pp. 1–20, 2025.
- [66] M. Kowsher *et al.*, “Llm-mixer: Multiscale mixing in llms for time series forecasting,” in *TRL Workshop at ACL*, 2025, pp. 156–165.
- [67] G. Woo *et al.*, “Pushing the limits of pre-training for time series forecasting in the cloudops domain,” *arXiv preprint arXiv:2310.05063*, 2023.
- [68] K. Lu *et al.*, “Frozen pretrained transformers as universal computation engines,” in *AAAI*, vol. 36, no. 7, 2022, pp. 7628–7636.
- [69] M. Chen *et al.*, “Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters,” in *ICML*, 2025, pp. 8979–9007.
- [70] L. Shen *et al.*, “Visionts++: Cross-modal time series foundation model with continual pre-trained vision backbones,” *arXiv preprint arXiv:2508.04379*, 2025.
- [71] S. Zhong *et al.*, “Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting,” in *ICML*, 2025, pp. 78 478–78 497.
- [72] T. Khezresmaeilzadeh *et al.*, “Vista: Vision-language inference for training-free stock time-series analysis,” *arXiv preprint arXiv:2505.18570*, 2025.
- [73] S. Yang *et al.*, “Timerag: Boosting llm time series forecasting via retrieval-augmented generation,” in *ICASSP. IEEE*, 2025, pp. 1–5.
- [74] K. Ning *et al.*, “Ts-rag: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster,” *arXiv preprint arXiv:2503.07649*, 2025.
- [75] K. Y. Liang *et al.*, “Retrieval-augmented generation with covariate time series,” *arXiv preprint arXiv:2603.04951*, 2026.
- [76] H. Zhang *et al.*, “Timeraf: Retrieval-augmented foundation model for zero-shot time series forecasting,” *IEEE*

- Transactions on Knowledge and Data Engineering*, 2025.
- [77] S. Lee *et al.*, “Cross-rag: Zero-shot retrieval-augmented time series forecasting via cross-attention,” *arXiv preprint arXiv:2603.14709*, 2026.
- [78] C. Ravuru *et al.*, “Agentic retrieval-augmented generation for time series analysis,” *arXiv preprint arXiv:2408.14484*, 2024.
- [79] Y. Ang *et al.*, “Tsgassistant: An interactive assistant harnessing llms and rag for time series generation recommendations and benchmarking,” *VLDB*, vol. 17, no. 12, pp. 4309–4312, 2024.
- [80] Y. Ang *et al.*, “Tsgbench: Time series generation benchmark,” *VLDB*, vol. 17, no. 3, pp. 305–318, 2023.
- [81] Z. Li *et al.*, “Tsfm-bench: A comprehensive and unified benchmark of foundation models for time series forecasting,” in *KDD*, 2025, pp. 5595–5606.
- [82] Y.-N. Chuang *et al.*, “Ltsm-bundle: A toolbox and benchmark on large language models for time series forecasting,” *ACM SIGKDD Explorations Newsletter*, vol. 27, no. 2, pp. 43–61, 2025.
- [83] M. Meyer *et al.*, “Benchmarking time series foundation models for short-term household electricity load forecasting,” *arXiv preprint arXiv:2410.09487*, 2024.
- [84] A. C. Franco *et al.*, “Forecasting oil production with time-series foundation models—a benchmark study against classical machine learning models,” in *SPE Annual Technical Conference and Exhibition*, 2025, p. D011S010R003.
- [85] G. Marchesi *et al.*, “Assessing time series foundation models for probabilistic electricity price forecasting: Toward a unified benchmark,” *Energies*, vol. 18, no. 23, p. 6269, 2025.
- [86] X. Yu *et al.*, “Hgprompt: Bridging homogeneous and heterogeneous graphs for few-shot prompt learning,” in *AAAI*, 2024.
- [87] X. Yu *et al.*, “Multigprompt for multi-task pre-training and prompting on graphs,” in *WWW*, 2024, pp. 515–526.
- [88] L. Xia *et al.*, “Opengraph: Towards open graph foundation models,” *arXiv preprint arXiv:2403.01121*, 2024.
- [89] Z. Wang *et al.*, “Gft: Graph foundation model with transferable tree vocabulary,” *NeurIPS*, vol. 37, pp. 107 403–107 443, 2024.
- [90] X. Yu *et al.*, “Text-free multi-domain graph pre-training: Toward graph foundation models,” *arXiv preprint arXiv:2405.13934*, 2024.
- [91] J. Liu *et al.*, “One model for one graph: A new perspective for pretraining with cross-domain graphs,” *arXiv preprint arXiv:2412.00315*, 2024.
- [92] Z. Guo *et al.*, “Graphmore: Mitigating topological heterogeneity via mixture of riemannian experts,” in *AAAI*, vol. 39, no. 11, 2025, pp. 11 754–11 762.
- [93] J. Zhao *et al.*, “Fully-inductive node classification on arbitrary graphs,” in *ICLR*, 2025.
- [94] R. Lv *et al.*, “Graphprompter: Multi-stage adaptive prompt optimization for graph in-context learning,” in *ICDE*, 2025, pp. 3917–3930.
- [95] Z. Wang *et al.*, “Towards graph foundation models: Learning generalities across graphs via task-trees,” in *ICML*, 2025.
- [96] S. Wang *et al.*, “Multi-domain graph foundation models: Robust knowledge transfer via topology alignment,” in *ICML*, 2025.
- [97] H. Chen *et al.*, “Autogfm: Automated graph foundation model with adaptive architecture customization,” in *ICML*, 2025.
- [98] X. Yu *et al.*, “Gcot: Chain-of-thought prompt learning for graphs,” in *KDD*, 2025, pp. 3669–3679.
- [99] Y. Sun *et al.*, “Handling feature heterogeneity with learnable graph patches,” in *KDD*, 2025, pp. 1313–1324.
- [100] X. Yu *et al.*, “Samgpt: Text-free graph foundation model for multi-domain pre-training and cross-domain adaptation,” in *WWW*, 2025, pp. 1142–1153.
- [101] Y. He *et al.*, “Unigraph2: Learning a unified embedding space to bind multimodal graphs,” in *WWW*, 2025, pp. 1759–1770.
- [102] L. Sun *et al.*, “Riemanngfm: Learning a graph foundation model from riemannian geometry,” in *WWW*, 2025, pp. 1154–1165.
- [103] Y. Zhu *et al.*, “Graphclip: Enhancing transferability in graph foundation models for text-attributed graphs,” in *WWW*, 2025, pp. 2183–2197.
- [104] Y. Huang *et al.*, “One prompt fits all: Universal graph adaptation for pretrained models,” in *NeurIPS*, 2025.
- [105] Z. Guo *et al.*, “Graphkeeper: Graph domain-incremental learning via knowledge disentanglement and preservation,” in *NeurIPS*, 2025.
- [106] T.-K. Nguyen *et al.*, “H2gfm: Towards unifying homogeneity and heterogeneity on text-attributed graphs,” *arXiv preprint arXiv:2506.08298*, 2025.
- [107] Z. Tang and J. Chen, “Toward a graph foundation model: Pre-training transformers with random walks,” *arXiv preprint arXiv:2506.14098*, 2025.
- [108] Z. Zhao *et al.*, “Towards text-free graph foundation models: Rethinking multi-domain graph contrastive learning,” *arXiv preprint arXiv:2506.22510*, 2025.
- [109] W. Ma *et al.*, “Gilt: An llm-free, tuning-free graph foundational model for in-context learning,” *arXiv preprint arXiv:2510.04567*, 2025.
- [110] Z. Wang *et al.*, “Gmope: A prompt-expert mixture framework for graph foundation models,” *arXiv preprint arXiv:2511.03251*, 2025.
- [111] L. Sun *et al.*, “Multi-domain transferable graph gluing for building graph foundation models,” in *ICLR*, 2026.
- [112] T. Lin *et al.*, “Langgfm: A large language model alone can be a powerful graph foundation model,” *arXiv preprint arXiv:2410.14961*, 2024.
- [113] X. Zhu *et al.*, “Llm as gnn: Graph vocabulary learning for text-attributed graph foundation models,” *arXiv preprint arXiv:2503.03313*, 2025.
- [114] J. Tang *et al.*, “Graphgpt: Graph instruction tuning for large language models,” in *SIGIR*, 2024, pp. 491–500.
- [115] Y. Li *et al.*, “Zerog: Investigating cross-dataset zero-shot transferability in graphs,” in *KDD*, 2024, pp. 1725–

- 1735.
- [116] L. Kong *et al.*, “Gofa: A generative one-for-all model for joint graph language modeling,” in *ICLR*, 2025.
- [117] Y. Cheng *et al.*, “Boosting cross-domain and cross-task generalization for text-attributed graphs from structural perspective,” *Frontiers of Computer Science*, 2025.
- [118] Y. Gao *et al.*, “Hypergraph foundation model,” *TPAMI*, 2025.
- [119] L. Sun *et al.*, “Deeper with riemannian geometry: Overcoming oversmoothing and oversquashing for graph foundation models,” in *NeurIPS*, 2025.
- [120] L. Sun and P. S. Yu, “A riemannian perspective on graph foundation models: curvature as a guiding principle,” *Frontiers of Computer Science*, vol. 20, no. 12, p. 2012370, 2026.
- [121] D. Eremeev *et al.*, “Turning tabular foundation models into graph foundation models,” *arXiv preprint arXiv:2508.20906*, 2025.
- [122] P. Veličković *et al.*, “Deep graph infomax,” in *ICLR*, 2019.
- [123] Y. Zhu *et al.*, “Deep graph contrastive representation learning,” *arXiv preprint arXiv:2006.04131*, 2020.
- [124] Z. Hou *et al.*, “Graphmae: Self-supervised masked graph autoencoders,” in *KDD*, 2022, pp. 594–604.
- [125] J. Zhao *et al.*, “Fug: Feature-universal graph contrastive pre-training for graphs with diverse node features,” *NeurIPS*, vol. 37, pp. 4003–4034, 2024.
- [126] Y. Li *et al.*, “Advancing graph foundation models: A data-centric perspective,” in *KDD*, 2025, pp. 1635–1646.
- [127] Y. Cui *et al.*, “A prompt-based knowledge graph foundation model for universal in-context reasoning,” *NeurIPS*, vol. 37, pp. 7095–7124, 2024.
- [128] L. Luo *et al.*, “Gfm-rag: Graph foundation model for retrieval augmented generation,” in *NeurIPS*, 2025.
- [129] L. Luo *et al.*, “G-reasoner: Foundation models for unified reasoning over graph-structured knowledge,” in *ICLR*, 2025.
- [130] H. Yuan *et al.*, “Rag-gfm: Overcoming in-memory bottlenecks in graph foundation models via retrieval-augmented generation,” in *WWW*, 2026.
- [131] H. Yuan *et al.*, “Retrieving minimal and sufficient reasoning subgraphs with graph foundation models for path-aware graphrag,” *arXiv preprint arXiv:2603.07179*, 2026.
- [132] Y. Zhu *et al.*, “Towards effective federated graph foundation model via mitigating knowledge entanglement,” in *NeurIPS*, 2025.
- [133] Z. Wu *et al.*, “Fedbook: A unified federated graph foundation codebook with intra-domain and inter-domain knowledge modeling,” *arXiv preprint arXiv:2510.07755*, 2025.
- [134] Y. Zhu *et al.*, “Rethinking federated graph foundation models: A graph-language alignment-based approach,” *arXiv preprint arXiv:2601.21369*, 2026.
- [135] H. Qiao *et al.*, “Anomalygfm: Graph foundation model for zero/few-shot anomaly detection,” in *KDD*, 2025, pp. 2326–2337.
- [136] H. Xu *et al.*, “Glip-ood: Zero-shot graph ood detection with graph foundation model,” *arXiv preprint arXiv:2504.21186*, 2025.
- [137] H. Xu *et al.*, “A systematic study of model extraction attacks on graph foundation models,” *arXiv preprint arXiv:2511.11912*, 2025.
- [138] X. Xue *et al.*, “Stealthy dual-trigger backdoors: Attacking prompt tuning in lm-empowered graph foundation models,” *arXiv preprint arXiv:2510.14470*, 2025.
- [139] Y. Wang *et al.*, “Heta: relation-wise heterogeneous graph foundation attack model,” in *IJCAI*, 2025, pp. 3453–3461.
- [140] J. Luo *et al.*, “Towards effective, stealthy, and persistent backdoor attacks targeting graph foundation models,” in *AAAI*, 2026.
- [141] J. Luo *et al.*, “Privacy auditing of multi-domain graph pre-trained model under membership inference attacks,” in *AAAI*, vol. 40, no. 18, 2026, pp. 15 483–15 491.
- [142] J. Chen *et al.*, “Gfm4ga: Graph foundation model for group anomaly detection,” *arXiv preprint arXiv:2601.10193*, 2026.
- [143] I. J. King *et al.*, “Cybergfm: Graph foundation models for lateral movement detection in enterprise networks,” *arXiv preprint arXiv:2601.05988*, 2026.
- [144] K. Huang *et al.*, “A foundation model for clinician-centered drug repurposing,” *Nature Medicine*, vol. 30, no. 12, pp. 3601–3613, 2024.
- [145] Z. Qin *et al.*, “Graphmsr: A graph foundation model-based approach for mri image super-resolution with multimodal semantic integration,” *Pattern Recognition*, p. 112178, 2025.
- [146] X. Wei *et al.*, “A brain graph foundation model: Pre-training and prompt-tuning across broad atlases and disorders,” in *ICLR*, 2026.
- [147] X. Zhang *et al.*, “Cellawaregnn: Single-cell enhanced knowledge graph foundation model for drug indication prediction,” *bioRxiv*, pp. 2026–02, 2026.
- [148] Z. Chen *et al.*, “Text-space graph foundation models: Comprehensive benchmarks and new insights,” *NeurIPS*, vol. 37, pp. 7464–7492, 2024.
- [149] J. Yang *et al.*, “Benchmarking graph foundation models,” in *KDD*, 2025, pp. 5866–5875.
- [150] X. Yu *et al.*, “Evaluating progress in graph foundation models: A comprehensive benchmark and new insights,” *arXiv preprint arXiv:2603.10033*, 2026.
- [151] T. Brown *et al.*, “Language models are few-shot learners,” *NeurIPS*, vol. 33, pp. 1877–1901, 2020.
- [152] J. Kaplan *et al.*, “Scaling laws for neural language models,” *arXiv preprint arXiv:2001.08361*, 2020.
- [153] H. Mao *et al.*, “Position: Graph foundation models are already here,” in *ICML*, 2024.
- [154] Q. Yao *et al.*, “Towards neural scaling laws for time series foundation models,” in *ICLR*, 2025.
- [155] A. Hayler *et al.*, “Bringing graphs to the table: Zero-shot node classification via tabular foundation models,” *arXiv preprint arXiv:2509.07143*, 2025.
- [156] A. Hayler *et al.*, “Of graphs and tables: Zero-shot node classification with tabular foundation models,” in

NPGML Workshop in NeurIPS, 2025.

[157] H. Latif-Martínez *et al.*, “Tsgfm - towards a graph foundation model for time series analysis in network monitoring,” in *TMA*. IEEE, 2025, pp. 1–4.



Qingyun Sun is currently an Associate Professor at the School of Computer Science and Engineering at Beihang University. Her research interests include graph machine learning and data mining. She has published several papers on IEEE TPAMI, IEEE TKDE, Web Conference, AAAI, ICDM, CIKM, etc.



Haonan Yuan is currently a Ph.D. candidate at the Beijing Advanced Innovation Center for Big Data and Brain Computing at Beihang University. His research interests include graph foundation model, dynamic graph learning and OOD generalization.



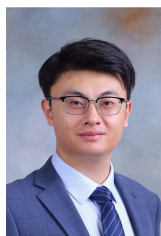
Yi Huang is currently a Ph.D. candidate at the Beijing Advanced Innovation Center for Big Data and Brain Computing at Beihang University. His research interests include information theory and relational deep learning for tabular data.



Ziwei Zhang is currently an associate professor in the School of Computer Science and Engineering at Beihang University. His research interests focus on machine learning on graphs, including graph neural network and automated graph machine learning. He has published 50 papers in prestigious conferences and journals, including ACM SIGKDD, ICML, NeurIPS, AAAI, IJCAI, IEEE TPAMI, and IEEE TKDE.



Xingcheng Fu is currently an assistant professor at the Key Lab of Education Blockchain and Intelligent Technology at Guangxi Normal University. His research interests include graph representation learning, complex networks, and social network analysis. He has published several papers on IEEE TKDE, Web Conference, AAAI, ICDM, CIKM, etc.



Ruijie Wang is currently a Professor at the School of Computer Science and Engineering of Beihang University, China. His research mainly focuses on trustworthy and efficient AI in open and dynamic environments, at the intersection of language, graph, and time-series data.



Haoyi Zhou is currently an Associate Professor in the School of Software at Beihang University, China. His research mainly focuses on time-series data and foundation models.



Jia Wu is an ARC DECRA Fellow at the School of Computing, Macquarie University, Sydney, Australia. He received the PhD from the University of Technology Sydney, Australia. His current research interests include data mining and machine learning. He has published 100+ refereed research papers in the above areas such as TPAMI, TKDE, TNNLS, TMM, NIPS, KDD, ICDM, IJCAI, AAAI and WWW. Dr. Wu was the recipient of SDM’18 Best Paper Award in Data Science Track and IJCNN’17 Best Student Paper Award. He currently serves as an Associate Editor of *ACM Transactions on Knowledge Discovery from Data* (TKDD). He is a Senior Member of IEEE.



Jianxin Li is currently a Professor with the School of Computer Science and Engineering, and Beijing Advanced Innovation Center for Big Data and Brain Computing in Beihang University. His current research interests include social networks, machine learning, big data, and trustworthy computing. Dr. Li has published research papers in top-tier journals and conferences, including the IEEE TKDE, TDSC, JAIR, ACM TOIS, TKDD, KDD, AAAI, and WWW.



Philip S. Yu is a Distinguished Professor and the Wexler Chair in Information Technology at the Department of Computer Science, University of Illinois at Chicago. Before joining UIC, he was at the IBM Watson Research Center, where he built a world-renowned data mining and database department. He is a Fellow of the ACM and IEEE. Dr. Yu was the Editor-in-chief of *ACM TKDD* (2011-2017) and *IEEE TKDE* (2001-2004).