A Survey of Structured Data Foundation Models: A Unified View on Foundation Models for Tables, Relational Databases and Knowledge Graphs

Mojtaba Nayyeri*[1] , Ratan Bahadur Thapa[1] , Daniel Hernández[1] , Arvindh Arun[1] , Steffen Staab[1,2]**

[1]University of Stuttgart

[2]University of Southampton

{mojtaba.nayyeri,ratan.thapa, daniel.hernandez, arvindh.arun, steffen.staab}@ki.uni-stuttgart.de

Abstract

Foundation models for text, images, video, or robot actions are trained with massive amounts of data to work on (human) prompts and sample answers from learned distributions. We survey foundation models that retrieve or predict answers from structured data. We use the term *structured data* as a unifying term to refer to data in tables, relational databases, or knowledge graphs. Such structured data exhibits and relates values; it may come with a schema and knowledge descriptions. Considering the analogy with other foundation models, a foundation model for structured data should be trained on large amounts of found and/or synthetic structured data (a dataset D_1) and it should be "prompted" with a query q that is executed on a structured dataset D_2, which may or may not overlap with D_1. The foundation model should retrieve and/or predict distributions over values, relations, or the schema and knowledge descriptions that the query q has asked for, regardless of whether these are in $D_1 \cup D_2$ or not. While foundation models for tables, relational databases, and knowledge graphs have been explored in recent years and great progress has been achieved, on closer inspection, one finds that these foundation models do not fully cover the task just defined. No existing structured data foundation model retrieves and predicts distributions for values, relations, and schema or knowledge descriptions. By providing a unified view and formalization of structured data foundation models, we provide a yardstick for measuring progress made on structured data foundation models and apply it in a survey of major paradigms.

1 Introduction

Foundation models [Bommasani *et al.*, 2021; Sun and Zheng *et al.*, 2025] have transformed natural language processing and computer vision by enabling broad transfer across domains and tasks through large-scale pretraining and reusable representations. Their defining contribution is not primarily architectural novelty, but the ability to internalize semantic regularities and reuse them across domains with minimal task-specific supervision. In natural language processing, meaning is carried by words and compositional phrases, and in vision, by higher-level visual primitives rather than individual pixels; in both cases, foundation models succeed by identifying the primary carriers of meaning at scale.

Structured data, which includes knowledge graphs [Hogan *et al.*, 2021], relational database [Codd, 1970] and tabular data [Martens *et al.*, 2015], on the other hand, has not benefited from this paradigm to the same extent. It underpins enterprise analytics, scientific data management, and decision-critical systems, but differs fundamentally from unstructured modalities: meaning in structured data is expressed through schema, relational topology, integrity constraints, and empirical data distributions, rather than surface order or local context [Abiteboul *et al.*, 1995]. Humans reason over structured data using identifiers, joins and paths, value distributions, co-occurrence patterns, and explicit constraints rather than sequential tokens. A common strategy to leverage text-centric models, such as CoddLLM [Zhang and Zhang *et al.*, 2025], to improve understanding of data management concepts and support analytics tasks including schema reasoning and text-to-table translation. Although these text-centric approaches often perform well on specific benchmarks, they do not natively enforce relational properties such as schema awareness and constraint satisfaction and rely on linearized representations that can obscure inherent relational structure, which can obscure relational structure and limit robustness and generalization across multi-table schemas [Van Breugel and Van Der Schaar, 2024; Yin *et al.*, 2020].

Recent research has therefore begun adopting foundation model principles from other known modalities to structured data, but progress remains fragmented. Knowledge graph foundation models emphasize inductive generalization over relational structure across graphs with disjoint vocabularies [Galkin and Zhou *et al.*, 2024]. Tabular foundation models focus on distributional and schema-agnostic prediction across heterogeneous tables [Hollmann *et al.*, 2025; Ye *et al.*, 2025a], while relational database models target multi-table reasoning and constrained query synthesis [Fey

** Equal Contribution

and Kocijan *et al.*, 2025; Robinson and Ranjan *et al.*, 2024]. Although all of these approaches pursue similar goals, they are evaluated in isolation and adopt inconsistent assumptions about transferability, invariance, and scale. As a result, it remains unclear what properties warrant the designation of a structured data foundation model.

This survey argues that meaningful progress in structured data foundation models (SDFMs) requires a requirements-driven perspective. We define SDFMs not as specific architectures, but as systems that operate directly on relational inputs and support reusable representations across tasks and schemas. To provide a unified view, we adopt a canonical relational abstraction that unifies tables, relational databases, and knowledge graphs, in Section 2, base the definition of structured data foundation models (SDFMs) on this unified view in Section 3, and derive characteristic, checkable criteria for SDFMs in Section 4. These criteria clarify the objectives that such models are expected to accomplish, independent of implementation choices, and provide a framework for evaluating transferability, invariance, and semantic generalization.

Using the introduced requirements-driven framework, we organize the literature into four paradigms: (i) knowledge graph foundation models (KGFMs) that learn transferable relational patterns; (ii) tabular foundation models (TFMs) and (iii) relational database foundation models (RDBFMs) that aim for schema-agnostic learning across datasets; and (iv) Large language models (LLMs)-centric methods based on serialization, retrieval, or tool use. For each paradigm, we analyze tasks, benchmarks, architectures, and pretraining objectives through the lens of the stated requirements (ref. Definition 4), distinguishing foundation model capabilities from task-specific success and emphasizing open challenges toward unified foundation models for structured data.

2 A Unified View on Structured Data

This section presents structured data as a canonical relational abstraction.

Let $\mathcal{V}$ be a countable domain of atomic values, and $\mathcal{L}$ a set of relation symbols, disjoint from $\mathcal{V}$, each relation symbol $l \in \mathcal{L}$ having fixed arity $ar(l) \in \mathbb{N}_{\geq 1}$.

Definition 1. *A structured dataset is a finite set of labeled tuples $\mathcal{D} \subseteq \bigcup_{l \in \mathcal{L}} \{l\} \times \mathcal{V}^{ar(l)}$, where each $(l, (v_1, \ldots, v_{ar(l)})) \in \mathcal{D}$ represents an atomic fact of type l.*

Consider $\mathcal{V} = \{$“LungCancer”, “TP53”, “Cisplatin”, true$\}$, and $L = \{$Disease, Gene, Treatment$\}$ with arities 1,2,3, respectively. Then, the dataset $\mathcal{D}$ defined by,

$$\mathcal{D} = \{(\text{Disease}, (\text{“LungCancer”})), (\text{Gene}, (\text{“LungCancer”}, \text{“TP53”})), (\text{Treatment}, (\text{“LungCancer”}, \text{“Cisplatin”}, \text{true}))$$

encodes that “LungCancer” is a disease associated with the gene “TP53,” and “Cisplatin” is an approved treatment for the “LungCancer.” This illustrates Definition 1, which abstracts structured data, under which knowledge graphs and relational databases arise as special cases given mild restrictions on relation symbols and value domains, as follows:

Knowledge Graphs. Let $\mathcal{E} \subseteq \mathcal{V}$ be a set of entities and $\mathcal{P} \subseteq \mathcal{L}$ be a set of predicates. A knowledge graph is $\mathcal{G} \subseteq \bigcup_{p \in \mathcal{P}} \{p\} \times \mathcal{E}^{ar(p)}$.

Tabular Data. Let $R = (C_1, \ldots, C_n)$ be column headers. A tabular instance over R is $\mathcal{D} \subseteq \{l_R\} \times \mathcal{V}_\perp^n$, where l_R uniquely identifies R and tuple values are interpreted positionally. CSV and spreadsheet files [Mitlöhner and Neumaier *et al.*, 2016; van, 2019] are external serializations of such instances, preserving column order and values.

Relational Database. Let $\mathcal{V}_\perp = \mathcal{V} \cup \{\perp\}$, where $\perp$ denotes *reserved symbol* Null. A relational schema $\mathcal{R} \subseteq \mathcal{L}$ consists of relation symbols $R \in \mathcal{R}$ with ordered attributes $att(R) = (A_1, \ldots, A_{ar(R)})$ and domains $dom(A_i) \subseteq \mathcal{V}$. A database instance is $\mathcal{D} \subseteq \bigcup_{R \in \mathcal{R}} \{R\} \times \mathcal{V}_\perp^{ar(R)}$, where tuples respect attribute domains. A relational database is the pair $(\mathcal{R}, \mathcal{D})$, and integrity constraints Σ (primary/foreign keys) correspond to embedded dependencies (EGD/TGD), now defined:

Constraints. A finite set Σ of *integrity constraints* is associated with a structured dataset $\mathcal{D}$. Constraint satisfaction is defined under the closed-world assumption [Abiteboul *et al.*, 1995]. We write $\mathcal{D} \models \Sigma$ if every constraint in Σ holds in $\mathcal{D}$ according to its semantics.

Definition 2. *An integrity constraint $\sigma \in \Sigma$ is a* first-order logic *(FOL) sentence of the form*

$$\forall \mathbf{x} \forall \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z}.\psi(\mathbf{x}, \mathbf{z}) \tag{1}$$

where $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are pairwise disjoint tuples of variables, $\varphi(\mathbf{x}, \mathbf{y})$ is a possibly empty conjunction of function-free atoms, and $\psi(\mathbf{x}, \mathbf{z})$ is a nonempty conjunction of function-free atoms over the relation symbols $\mathcal{L}$.

The antecedent and consequent in Equation (1) are the *body*(σ) and *head*(σ) of σ, respectively, and constraints of the form σ are referred to as *embedded dependencies* [Fagin, 1982]. The σ is called *tuple-generating dependency* (TGD) when all the atoms in *head*(σ) are equality-free, and *equality-generating dependency* (EGD) [Beeri and Vardi, 1984] when *head*(σ) is a single equality atom.

Query. Let $\mathbf{x}$ denote the tuple of free variables of φ, and $\varphi(\mathbf{x})$ denote the query condition expressed as a FOL formula.

Definition 3. *Let $\mathcal{D}$ be a structured dataset over a set of relation symbols $\mathcal{L}$ with value domain $\mathcal{V}$. A query q is a FOL formula $\varphi(\mathbf{x})$ with free variables $\mathbf{x}$ over $\mathcal{L}$. The evaluation of q on $\mathcal{D}$, denoted $q(\mathcal{D})$, is the set of all tuples $\mathbf{v} \in \mathcal{V}^{|\mathbf{x}|}$ such that $\varphi(\mathbf{x})$ is satisfied in $\mathcal{D}$ under the assignment $\mathbf{x} \mapsto \mathbf{v}$:*

$$q(\mathcal{D}) = \{\mathbf{v} \in \mathcal{V}^{|\mathbf{x}|} \mid \mathcal{D} \models \varphi[\mathbf{x} \mapsto \mathbf{v}]\}. \tag{2}$$

This notion of query answering generalizes tasks commonly targeted in existing SDFMs, such as link/node prediction, table completion/imputation, and, when $\mathbf{x}$ is empty, boolean testing or constraint checking.

3 Structured Data Foundation model

A structured data foundation model is a model that learns general representations and reasoning capabilities over structured datasets, enabling transfer across datasets, schemas, and tasks. Let $\mathfrak{D}$ denote the set of structured datasets and Q the set of queries over $\mathfrak{D}$.

Definition 4. *A* structured data foundation model *is a parameterized model $\mathcal{F}_\theta$ pretrained on a large dataset $\mathcal{D}_1 \subseteq \mathfrak{D}$*

Model	(A) Pretraining & Adaptation (How)					(B) Query Capabilities (What)				(C) Transferability (What)	
	Unsupervised pretraining		Supervised fine-tuning / adaptation		RL	Value/Cell	Label	Schema	Query ans.	R	A
	Data	Approach	Data	Approach							
ULTRA	KG	Link Pred. Loss	Target KG (opt.)	Cont. Training	No	✓	✗	✗	✗	Bin. Rel. Patt.	NBFNet
TRIX	KG	Link (ent. + rel) Pred. Loss	Target KG (opt.)	Cont. Training	No	✓	✓	✗	✗	Bin. Rel. Patt*.	NBFNet
MOTIF	KG	Link Pred. Loss	Target KG (opt.)	Cont. Training	No	✓	✗	✗	✗	HO Motifs.	HCNets+NBFNet
GAMMA	KG	Link Pred. Loss	Target KG (opt.)	Cont. Training	No	✓	✗	✗	✗	Bin. Rel. Patt.	G-NBFNet
ULTRAQuery	KG	Link. Pred. Loss	KG Complex Queries	BCE loss (CLQA queries)	No	✓	✗	✗	✓	Bin. Rel. Patt.	NBFNet
FLOCK	KG	Link (ent. + rel) Pred. Loss	Target KG (opt.)	Cont. Training	No	✓	✓	✗	✗	Prob. rel. inv.	Seq encod, BiGRU
POSTRA	TKG	Link Pred. Loss	Target KG (opt.)	Cont. Training	No	✓	✗	✗	✗	T.Bin. Rel. Patt.	T-NBFNet
SEMMA	Rel. Attr. KG	Link Pred. Loss	Target KG (opt.)	Cont. Training	No	✓	✗	✗	✗	Bin. Rel. Patt + Txt Sem.	LLM+NBFNet
HYPER	Hypergraph KG	Link Pred. Loss	Target KHG (Opt.)	Cont. Training	No	✓	✗	✗	✗	Positional rel. interact.	HCNets
TabPFN 2.5	Synth. tab. data	PFN meta-train. (ICL)	Real tab. data (opt.)	FT (Real-TabPFN)	No	✓	✓	✗	✗	Synth. func. prior	PFN-Transf.
TabICL	Synth. tab. data	ICL meta-train.	Target tab. (ctx exm.)	Prompt / ICL (no upd.)	No	✓	✓	✗	✗	Synth. tab. prior	Col→Row ICL-Transf.
TabDPT	Real tab. corp.	SSL (rand. col pred.)	Targ. tab. (ctx + retriv.)	Retrieval-aug. ICL	No	✓	✓	✗	✗	Real col. deps	Retriev. ICL-Transf.
SAP-RPT-1 (ConTextTab)	Large real tabl.	STNP	Target table (w ctx)	Semantics-aware ICL	No	✓	✓	✓	✗	Schema semantics	Transformer
TabGLM	Tabl. + txt/meta	Graph-Txt Consistency Min	Labeled tab tasks	Joint sup. obj.	No	✓†	✓	✗	✗	Structure+Txt Semantic	GNN+Transf.
UniTabE	Large tab. corp.	Mask-cell + contras. SSL	Downstream tasks	Prompt-based FT	No	✓	✓	✓	✓	Cell Dist+Map	Transf+LSTM
TableGPT2	Code	Cont. Pretrain.	Instr. TableQA. code	CL, MFA, IT	No	✓	✓	✓	✓	Column semantics	Qwen2.5
TabuLa-8B	Text	Causal LM pretrain	TabLib corp.	SFT on serial. tab.	No	✓†	✗	✓	✗	TabLib distrib	LLAMA3
Relational Transformer (RT)	RDBs (RelBench)	Masked token pred.	Target DB/task (opt.)	FT (opt.)	No	✓	✓	✓	✗	Schem. sem. / Func. Dep.	Relational attn
Griffin	Single-tab. data	Rand. msk cell comp.	Single-tab. + RDB tasks	Joint SFT → task FT	No	✓	✓	✓	✗	Schem. sem. / Func. Dep.	Attn + Msg. Pass.
KumoRFM	Public DBs + synthetic	Rel. ICL pretrain.	Target DB + PQL (opt.)	ICL + optional FT	No	✓	✓	✓	✓	Schem. sem. / Func. Dep. / Structure	Rel. Graph Transf.

Table 1: **Structured data foundation models** characteristics summary (see section 4 for formalization and section 5 for detailed reviews) in the terms of **pretraining/adaptation**, **query capabilities**, and **transferability**: Three classes of SDFMs are included: KGFMs (first 9 models), TFMs (8 models in the middle), and RDBFMs (last 4 models). **R** = transfer Regularities. **A** = Architectural Inductive Bias, **Bin. Rel. Patt.** = Binary Relational Patterns, **Bin. Rel. Patt*.** = Binary Relational Patterns with variable binding, **T.Bin. Rel. Patt.** = Binary Temporal Relational Patterns, **HO Motifs.** = Higher-order Motifs ($\geq$ 3 relations), **G/T-NBFNet** = Geometric/Temporal Neural Bellmanford, **CL, MFA, IT** = Contrastive Learning, Multtask Feature Alignment, Instruction Tuning. **Semantics-aware table-native pretrain** = STNP. † TabPFN can handle missing values in evaluations, but it is not a dedicated imputation model.

such that, for any $q \in Q$ and any target dataset $\mathcal{D}_2 \subseteq \mathfrak{D}$, it produces an output $\mathcal{D}_3$ satisfying $\mathcal{F}_\theta(q, D_2) = \mathcal{D}_3 \simeq q(D_2)$, where $\mathcal{D}_3$ represents the prediction of $\mathcal{F}_\theta$ for q on $\mathcal{D}_2$.

Drawing the analogy to text language models, $\mathcal{D}_1$ takes the role of pre-, mid-, and posttraining[1] data, comprising a large set of knowledge graphs, tables, and/or relational databases, and $q(D_2)$ takes the role of a prompt, with the aim to predict correct answers that may not just be retrieved. The "any target dataset $\mathcal{D}_2 \subseteq \mathfrak{D}$" condition in Definition 4 is idealized; in practice, transferability will degrade when pretraining and target datasets diverge.

4 SDFM Characteristics

4.1 Query Capabilities

Let $\mathcal{D} \subseteq \bigcup_{l \in \mathcal{L}} \{l\} \times \mathcal{V}_\perp^{ar(l)}$ be a structured dataset over relation symbols $\mathcal{L}$. We break down the generic querying task $q(\mathcal{D})$ specified in Definition 4, into specific tasks addressed by related work on SDFMs: value prediction, label prediction, schema reasoning, and query answering.

Value Prediction or Cell Completion. Given a tuple $l(v_1, \ldots, ?, \ldots v_{ar(l)}) \in \mathcal{D}$ with a missing or undefined value ?, value prediction is the query q such that $q(\mathcal{D}) = \{v \in \mathcal{V} \mid l(v_1, \ldots, v, \ldots v_{ar(l)}) \in \mathcal{D}\}$.

[1]Post-training preference data might take shapes different from Definition 1. We do not suggest a shape for preference data, as this would be too speculative at this point.

Depending on the specification of q, it may not only predict a single value, but also entire columns, rows, or tables. In databases, missing values are typically represented by $\perp$.

Label Prediction or Relation/Link Prediction. Given a tuple of values $(v_1, \ldots, v_k) \in \mathcal{V}^k$, the label prediction is the query q such that $q(\mathcal{D}) = \{l \in \mathcal{L} \mid l(v_1, \ldots, v_k) \in \mathcal{D}\}$.

Schema Reasoning. Schema Reasoning is the query $q = \varphi(\mathbf{x})$ over $\mathcal{D}$ that returns all tuples satisfying the specified structural or semantic property: $q(\mathcal{D}) = \{\mathbf{v} \in \mathcal{V}^{|\mathbf{x}|} \mid \mathcal{D} \models \varphi[\mathbf{x} \mapsto \mathbf{v}]\}$. The $\varphi(\mathbf{x})$ specifies the semantic property (e.g., key constraints, type consistency, or valid cross-relation reference) to be enforced, and the mapping $[\mathbf{x} \mapsto \mathbf{v}]$ enforces it. For the "LungCancer" dataset ($\mathcal{D}_1$), constraints like "all Treatment entries must reference a valid Disease," or "all Gene entries must be linked to a Disease" can be expressed as FOL queries:

$$\sigma_1 : \forall x, y, z.\ \text{Treatment}(x, y, z) \rightarrow \exists z'.\ \text{Disease}(z') \wedge z = z',$$
$$\text{and } \ \sigma_2 : \forall x, y.\ \text{Gene}(x, y) \rightarrow \exists x'.\ \text{Disease}(x') \wedge x = x'.$$

Then, schema reasoning for SDMFs $\mathcal{F}_\theta$ (Definition 4) is to evaluate these queries (annotated with the resp. $\varphi(\mathbf{x})$) on $\mathcal{D}_2$ to produce $\mathcal{D}_3$ such that $\mathcal{F}_\theta(q_{\sigma_1}, D_2) = \mathcal{D}_3^{\sigma_1} \simeq q_{\sigma_1}(D_2)$ and $\mathcal{F}_\theta(q_{\sigma_2}, D_2) = \mathcal{D}_3^{\sigma_2} \simeq q_{\sigma_2}(D_2)$, the sets of tuples satisfying the constraints. Tuples violating σ_1 or σ_2 should be absent from $\mathcal{D}_3$ or flagged.

Query Answering. Query answering, as per Def. 3, returns all tuples that satisfy a formula of the form $\varphi(\mathbf{x})$ over $\mathcal{D}$.

4.2 Pretraining Properties

Pretraining properties describe how a SDFM acquires general capabilities. A model $\mathcal{F}_\theta$ satisfies these properties by combining: (i) *self-supervised pretraining*, optimizing θ to capture latent structural regularities without labels; (ii) *supervised fine-tuning*, adapting θ on a labeled dataset to minimise task-specific loss; (iii) *reinforcement learning*, optimizing θ as policy to maximize an expected reward signal.

4.3 Transferability Properties

Transferability properties describe the ability of an SDFM model to generalize beyond the training data. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be structured datasets, as defined in Definition 4.

Cross Data . A model exhibits cross-data transferability if it generalizes learned patterns from the pretraining dataset $\mathcal{D}_1$ to another dataset $\mathcal{D}_2$, even when the datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ partially overlap or completely differ. This includes the following cases, characterized by the relationship between the relation symbols (schema) and value domains (content) of $\mathcal{D}_1$ and $\mathcal{D}_2$: (i) *Compositional transfer*, where $\mathcal{D}_1$ and $\mathcal{D}_2$ share the same or largely overlapping sets of relation symbols, but the value domain of $\mathcal{D}_2$ contains novel combinations of entities, relations, or column types that do not occur jointly in $\mathcal{D}_1$; (ii) *Cross-schema transfer*, where the sets of relation symbols of $\mathcal{D}_1$ and $\mathcal{D}_2$ are partially overlapping or disjoint, while their value domains may be overlapping or distinct, and $\mathcal{D}_2$ introduces previously unseen relation symbols or new configurations thereof relative to $\mathcal{D}_1$; (iii) *Cross-domain transfer*, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are drawn from different application domains, with differing value domains and possibly different sets of relation symbols, but exhibiting shared structural regularities or semantically related relation symbols; and (iv) *Cross-modal transfer*, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are represented using different structured data modalities (e.g., knowledge graphs versus relational or tabular data), leading to substantially different sets of relation symbols and value domains, yet requiring reasoning over conceptually aligned relational information.

Cross Task. A model exhibits cross-task transferability if representations and reasoning strategies learned from one query type $q_1(\mathcal{D}_1)$ to generalize to a different query type $q_2(\mathcal{D}_2)$ without retraining.

5 Modeling Paradigms

This section summarizes the foundation models for structured data, contrasting paradigms in terms of query capabilities, pretraining strategies, transferability, and architectural inductive biases.

5.1 Knowledge Graph Foundation Model

KGFMs are cross-domain, vocabulary-agnostic models that learn structurally invariant reasoning patterns from several KGs and transfer them zero-shot to entirely new graphs with unseen entities and relations [Arun and Kumar *et al.*, 2025; Galkin *et al.*, 2023]. Existing KGFMs initially target triple-based KGs [Galkin *et al.*, 2023], which are then extended to hyper-relational and temporal graphs and settings that fuse text/attributes with structure [Huang *et al.*, 2025; Lee and Whang, 2025; Pan and Nayyeri *et al.*, 2025; Arun and Kumar *et al.*, 2025]. Below, we review key aspects of KGFMs within the lens of the defined foundation model characteristics (section 4).

Query Capabilities Existing KGFMs have so far been evaluated on a narrow set of reasoning tasks, with ***value prediction*** (entity/link prediction) as the dominant one. Models like ULTRA are trained to answer missing-entity queries in a fully inductive setting, i.e., on entirely unseen multi-relational graphs [Galkin *et al.*, 2023]. A closely related ***label prediction*** variant is relation prediction (inferring the relation type between entity pairs), where Flock reports strong zero-shot performance, while entity classification remains comparatively underexplored [Kim *et al.*, 2025]. Beyond single-edge queries, ***query answering*** over conjunctive multi-hop patterns has been enabled by stacking a reasoning layer on top of a link predictor (e.g., UltraQuery), allowing zero-shot complex query answering on new graphs [Galkin and Zhou *et al.*, 2024]. Overall, the limited task coverage in current KGFM work leaves cross-task transferability largely untested and open.

Pretraining Property KGFMs are typically self-supervised via **unsupervised pretraining** on multiple KGs using objectives derived from known facts, most often link prediction/KG completion with contrastive (ranking) losses over corrupted head/tail negatives [Galkin *et al.*, 2023]. This is still done on a modest scale of only a few million triples (e.g., ULTRA pretrains on FB15k-237, WN18RR, CoDEx-M) [Galkin *et al.*, 2023]. Crucially, domain diversity and careful KG corpus curation matter more than sheer triple count, with evidence that adding new domains yields larger gains than adding more same-domain triples (scaling law) [Feng *et al.*, 2025]. Some methods enrich pretraining with masked graph modeling [He *et al.*, 2025], or text–graph alignment for attributed KGs, and training is usually inductive (e.g., multi-graph batching/masking and regularizers to reduce ID overfitting [Kim *et al.*, 2025]).

For adaptation, ***supervised fine-tuning*** continues training for a few epochs on the target KG's known triples, often improving MRR/Hits@K metrics beyond zero-shot, while *parameter-efficient tuning* becomes especially relevant in KGFM+LLM hybrids where the LLM can be frozen and only graph modules updated [Arun and Kumar *et al.*, 2025]. Data quality and relation-pattern coverage are crucial, motivating careful KG corpus curation. Future web-scale graph pretraining will hinge on handling curation, quality, and noise.

Transferability Property Regarding ***transferable regularities***, structured KGFMs target *double equivariance* (to relabel entities and relations), so predictions depend on *structure (not IDs)* and transfer *relation motifs/relation interaction patterns* (**cross-schema transfer**), with transferability fundamentally limited by the motif set the architecture can represent [Huang *et al.*, 2025]. *ULTRA* transfers *binary relation patterns* by building a *relation graph* whose edges encode four head/tail co-occurrence types (*h–t*, *t–h*, *h–h*, *t–t*) as a 4-channel adjacency. A GNN [Zhu and Zhang , 2021] produces query-conditioned relation embeddings to score links on unseen KGs, effectively transferring *2-atom "soft rule" motifs* such as $p_1(x,y) \wedge p_2(y,z) \rightarrow q(x,z)$, but patterns requiring *higher-order motifs* (joint interactions of $\geq$3 relations) are hard unless they decompose into pairwise interactions [Huang *et al.*, 2023]. TRIX [Zhang *et al.*, 2025] increases expressivity by making interactions *entity-aware* (sparse adjacency that preserves *variable bindings*), separating substructures that look identical to ULTRA, and improving zero-shot accuracy under complex overlaps.

Architecture inductive bias: across KGFMs, designs avoid raw IDs and enforce symmetry via (i) *GNN-based* relation/entity message passing (extended to *hypergraphs* with argument-position edges [Feng *et al.*, 2025] and to *temporal KGs* [Pan and Nayyeri *et al.*, 2025] with time-conditioned updates), (ii) *Transformer/sequence*, e.g., FLOCK [Kim *et al.*, 2025], encodes sampled random walks with a recording scheme to preserve roles and uses stochasticity to break symmetries, and (iii) *Hybrid (structure+semantics) models* fuse structural models with language modules: SEMMA [Arun and Kumar *et al.*, 2025] combines ULTRA-like structural patterns with LLM-derived relation-text embeddings via fusion (e.g., co-attention/gating), improving cases where structure alone lacks an analogue.

Across paradigms, KGFMs balance symmetry/equivariance for transfer while preserving expressivity, closely tied to Weisfeiler–Lehman-style discrimination and permutation-equivariant set/graph networks [Huang *et al.*, 2023].

Benchmark Datasets To evaluate cross-KG generalization (section 4), benchmarks now use multi-KG collections spanning diverse domains, typically pre-training on a few standard KGs and testing on many others. For instance, ULTRA and other works [Galkin *et al.*, 2023] pre-train on FB15k-237, WN18RR, and CoDEx-M and are evaluated on 50+ unseen KGs (about 1K–120K nodes and 5K–1M edges). Benchmarking remains challenging [Arun and Kumar *et al.*, 2025]:many "different" KGs still originate from a few sources (Freebase/WordNet/Wikidata/NELL/YAGO), so repeating schemas can cause overlap/leakage and inflate apparent generalization (cross-schema generalization in subsection 4.3). More truly domain-diverse, larger, realistic collections (including specialized/non-public KGs) are needed. Even so, current results indicate KGFMs can outperform transductive models when tested on completely unseen graphs [Galkin *et al.*, 2023].

5.2 Tabular Foundation Model

Tabular data is pervasive (healthcare, finance, science, cybersecurity), yet tabular deep learning still often lags gradient-boosted trees [Van Breugel and Van Der Schaar, 2024]. Recent position papers argue that, as in NLP/vision, scaling model capacity and pretraining on diverse tables could unlock broader tabular capabilities [Van Breugel and Van Der Schaar, 2024]. TFMs pursue this by learning transferable knowledge across many tables to enable multiple downstream tasks on new datasets with minimal retraining, using inductive biases for mixed numeric/categorical features and tabular structure, while avoiding LLM-based table adaptation issues such as handling continuous values, calibration, and high cost.

Query Capabilities Regarding ***Value prediction/cell completion***, TFMs can perform zero-shot imputation when pretrained for missingness, e.g., TabPFN [Hollmann *et al.*, 2023] is trained on synthetic tables with missing values and supports missing-data handling, and TabImpute [Feitelberg and Saha *et al.*, 2025] is a pretrained transformer tailored for fast, accurate zero-shot imputations without fitting or tuning. For ***label prediction***, TFMs primarily target classification/regression (predicting a target column/row from the others) and achieve state-of-the-art performance, e.g., TabPFN via a single forward pass, and in-context pretrained models such as TabICL and TabDPT report strong results across large benchmark suites [Hollmann *et al.*, 2023; Qu *et al.*, 2025; Ma and Thomas *et al.*, 2025]. ***Schema reasoning and FOL-style query answering*** remain largely open: current TFMs typically assume a fixed schema and focus on within-table prediction rather than inferring/validating semantics (keys/foreign keys, constraints) or answering general logical queries over relational/tabular data.

Pretraining Property TFMs are typically ***pretrained*** on large synthetic or unlabeled tables and/or diverse real-world corpora: for example, TabPFN/TabICL train on millions of synthetic tables spanning many feature–label relationships, while real-data TFMs scale up using broad collections such as TabDPT (OpenML: 123 datasets, $\sim$32M rows, $\sim$2B cells) and table corpora that include mixed numeric/categorical, and sometimes text-rich columns (e.g., TabSTAR) [Hollmann *et al.*, 2025; Ma and Thomas *et al.*, 2025]. Pretraining commonly uses masked value reconstruction (predict masked/missing cells), meta-learning/in-context objectives (simulate many supervised tasks so the model predicts from a small labeled context in one forward pass), and sometimes contrastive alignment to tie together semantically related table elements [Hollmann *et al.*, 2025; Kim and Lefebvre *et al.*, 2025]. After pretraining, TFMs adapt via in-context use (no weight updates, strong in low-data settings), full fine-tuning on a new table, or parameter-efficient tuning (adapters/LoRA/prompt vectors or feature-extractor use), with RL-based refinement emerging for multi-step table reasoning/manipulation in LLM-style tabular models (e.g., trajectory/reward-driven tuning) [Tanna and Seth *et al.*, 2025; Yang and Huang *et al.*, 2025].

Transferability Property

Regarding ***Transferable Regularities***, a core promise of TFMs is to learn transferable inductive biases from many tables, capturing both the joint data distribution (how columns co-vary, including higher-order/transitive associations) and reusable implicit function mappings between columns that generalize to new tables and domains. Empirically, this appears as in-context learning: models like TabPFN and TabICL can take a small labeled table as input context and predict accurately on a new task without gradient updates. TabPFN is even shown to approximate Bayesian inference under a rich prior, yielding strong (often well-calibrated) predictions on unseen tasks. Transfer can extend beyond "standard tables", reframing graph node classification [Hayler *et al.*, 2025] as a tabular problem enables zero-shot use of TFMs that can outperform specialized GNNs. TFMs have also shown strong results for time-series forecasting when time-indexed features are treated as tabular inputs. More broadly, incorporating text/metadata context (e.g., column names/descriptions) further improves semantic transfer, supporting the view that TFMs reuse general patterns rather than task-specific rules [Kim and Lefebvre *et al.*, 2025].

Regarding ***Architecture Inductive Bias***, tables are inherently 2D (rows $\times$ columns) and typically unordered. Thus, TFMs need inductive biases for row/column permutation invariance and feature–population modeling. TabPFN uses a two-way Transformer where each cell attends within-row (feature interactions) and within-column (across-sample distributions), enabling training on small tables while extrapolating to larger ones without architectural changes. TabPFN [Hollmann *et al.*, 2023; Ye *et al.*, 2025b] also keep this alternating attention and adds efficiency optimizations (e.g., flash attention, half-precision norms) to scale to millions of cells. For very large tables, TabICL [Qu *et al.*, 2025] introduces a column-then-row pipeline: a lightweight column-wise model produces row embeddings, then a Transformer performs in-context inference over those embeddings, scaling to hundreds of thousands of samples. Handling heterogeneous types is addressed via categorical embeddings and numeric encodings/tokenizers (beyond naive LLM tokenization) [Van Breugel and Van Der Schaar, 2024], and some work builds graphs over tables and applies GNNs to capture instance correlations and higher-order feature interactions [Li *et al.*, 2025]. Finally, knowledge-driven hybrids like TARTE fuse textual schema/value semantics (e.g., column names, units) with tabular values to inject a "world-knowledge" prior [Kim and Lefebvre *et al.*, 2025], e.g., knowing that a column labeled "age" is numeric and bounded or that "USD" implies a currency range. Overall, state-of-the-art tabular architectures blend Transformers with custom modules to respect tabular structure, achieve permutation invariances, and incorporate semantic context.

Benchmark Datasets TFM evaluation is increasingly done on multi-dataset benchmarks that stress cross-table generalization. TALENT (300+ datasets spanning sizes, feature types, and classification/regression tasks) shows recent pretrained tabular models can often match or surpass tree ensembles like XGBoost, though ensembles and tree methods still dominate on some datasets and with heavy tuning [Ye and Liu *et al.*, 2024]. TabArena is a "living" benchmark that continuously curates datasets/models, with early results suggesting TFMs shine on small datasets where prior knowledge helps [Erickson *et al.*, 2025]. Protocols typically report aggregate performance (e.g., average rank/win rate across many tasks) and relate outcomes to dataset meta-features (e.g., categorical–numeric mix), aiming to measure foundation-level transfer rather than single-dataset overfitting.

5.3 Relational Foundation Model

Relational databases pose distinct challenges for foundation models, as information is spread across interlinked tables with heterogeneous attributes and key constraints, requiring multi-tuple reasoning. Early work modeled databases as graphs, i.e., rows as nodes and keys as edges [Robinson and Ranjan *et al.*, 2024]. Recent models adopt (i) schema-aware sequence models or (ii) graph-based GNN/transformer architectures [Dwivedi *et al.*, 2025]; both designed to preserve schema invariants and enable cross-schema transfer [Vogel *et al.*, 2022; Wehrstein *et al.*, 2025].

Query Capabilities. Relational foundation models support multiple predictive and reasoning queries in one framework. (i) ***Value Prediction / Cell Completion***: models impute missing values and repair incomplete records by propagating signals across related tables, supporting data cleaning and recommendation tasks [Wang and Wang *et al.*, 2025]. (ii) ***Label Prediction***: supervised prediction of target attributes uses multi-table context to support churn, fraud, and sales forecasting, learning join patterns and temporal dynamics [Fey and Kocijan *et al.*, 2025]. (iii) ***Schema Reasoning***: tasks such as schema matching, primary/foreign keys discovery, and constraint validation remain largely unaddressed but represent future opportunities. (iv) ***query Answering***: general SQL/FOL-style queries remain largely unimplemented; existing predictive interfaces (e.g., KumoRFM) focus on forecasts or recommendations.

Pretraining Property. (i) ***Unsupervised Pretraining***: RDBFMs are pretrained on heterogeneous structured corpora (e.g., WikiDBs [Vogel *et al.*, 2024]), including multi-table benchmarks, e.g., RelBench [Robinson and Ranjan *et al.*, 2024] and mixed single-/multi-table datasets (e.g., Griffin [Wang and Wang *et al.*, 2025]) to capture diverse schemas, numerical/textual features, and relational patterns. (ii) ***Pretraining Approach***: self-supervised objectives such as masked cell prediction and reconstruction teach models to integrate schema metadata, relational links, and cross-table dependencies [Dwivedi *et al.*, 2025]. (iii) ***Supervised Fine-Tuning***: adapts pretrained models to domain-specific predictive tasks such as classification, regression, and forecasting (e.g., KumoRFM). (iv) ***Reinforcement Learning*** remains largely unexplored.

Transferability Property. (i) ***Transferable Regularities***: RFMs learn patterns of relational interaction driven by schema semantics, relational attention, and cross-table context that generalize to unseen schemas and tasks without dataset-specific fine-tuning [Ranjan and Hudovernik *et al.*, 2025]. (ii) ***Architecture Inductive Bias***: schema-aware sequences and graph-based message passing encode relational structure, supporting cross-schema transfer and structured reasoning [Dwivedi *et al.*, 2025].

Benchmark Datasets. RFMs are evaluated on curated multi-database collections such as RelBench, WikiDBs, WikiDBGraph [Wu *et al.*, 2025], and enterprise ERP datasets [Klein and Biehl *et al.*, 2024].These benchmarks measure schema reasoning and cross-schema generalization, while evaluation on declarative SQL query execution remains limited.

5.4 LLMs for Structured Data

Structured data store explicit typed facts, whereas LLMs encode parametric knowledge from unstructured text. LLM-based hybrid approaches exploit structured data as external memory or supervision [Pan and Razniewski et al., 2023], but direct application reveals limitations in context length, relational reasoning, numeric precision, and grounding in up-to-date information. We review LLM approaches for KGs, databases and tabular data, highlighting tasks, training, transfer, architectures, and motivating specialized structured-data foundation models (SDFMs).

LLMs for Knowledge Graph Reasoning. LLMs face KG structural gaps: (i) KGs provide explicit relational facts beyond LLM parametric knowledge, and (ii) LLMs excel at reasoning and language generation but struggle with long paths, dense nodes, and incomplete knowledge [Cui *et al.*, 2025] (***Query***). To address this, KG-augmented approaches combine LLM reasoning with specialized retrievers: (i) the LLM generates outputs, while (ii) a KG retriever encodes relations, identifies relevant nodes, and propagates along informative paths, enabling zero-shot generalization to unseen KGs [Cui *et al.*, 2025; He *et al.*, 2024; Omeliyanenko *et al.*, 2023] (***Transfer***). Alternatively, graph textualization linearizes nodes, edges, and attributes into text, allowing LLM-only reasoning [Lin and Yan *et al.*, 2024], or cross-modal frameworks like BioBRIDGE align embeddings via KGs without fine-tuning [Wang and Wang *et al.*, 2024] (***Inductive Bias***). Limitations remain: (i) LLMs lack native graph structure encoding, (ii) long paths or dense graphs exceed context windows, and (iii) hallucination of nonexistent relations is possible [Cui *et al.*, 2025]. These issues motivate graph-aware foundation models that internalize structure.

LLMs for Tabular Learning. Tabular data poses distinct challenges: (i) heterogeneous numeric/categorical types in 2D layout, and (ii) tables often exceeding LLM context windows [Wu *et al.*, 2025] (***Query***). Linearization of rows with column headers enables classification, reasoning, completion, and question-answering over tables, while prompting and chunking mitigate length limits. Early models like TaBERT pretrain text and tables jointly to capture structure [Yin *et al.*, 2020] (***Pretraining***). Recent foundation models, e.g., TabuLa-8B, pretrain on billions of rows from millions of tables, achieving strong zero- and few-shot generalization beyond XGBoost [Chen, 2016] and TabPFN [Hollmann *et al.*, 2025] (***Transfer***). Hierarchical encodings and the use of metadata further improve schema transfer [Fang and Xu *et al.*, 2024] (***Inductive Bias***). Existing limitations include: (i) loss of 2D alignment, (ii) challenges in numeric and logical reasoning, (iii) poor generalization to previously unseen schemas, and (iv) high computational cost, motivating tabular SDFMs treating tables as first-class inputs.

LLMs for Relational Databases. Applying LLMs to relational databases typically involves translating natural language into SQL or retrieving structured data. Although LLMs can generate syntactically correct SQL, they often struggle with complex schemas, multi-table joins, integrity constraints, and live or private data [Peixian and Zhuang *et al.*, 2025; Qin and Luo *et al.*, 2024] (***Query***). To address these limitations, database-augmented LLMs integrate: (i) selection and value retrieval modules, (ii) live database memory to ground outputs, and (iii) guided prompt pipelines, enabling queries beyond the model's original training distribution [Qin and Luo *et al.*, 2024] (***Transfer***). Hybrid approaches further encode schemas or query plans using GNNs before passing them to LLMs, enforcing foreign key relationships and reducing semantic errors [Wu *et al.*, 2025] (***Inductive Bias***). Key challenges remain, including ensuring transactional consistency, scaling to large databases and preserving privacy, wich motivate SDFMs that natively encode schema, constraints, and query execution.

6 SDFMs: Outlooks and Perspectives Ahead

6.1 Foundation Model Queries

Foundation-model queries will evolve into general primitives for structured data workflows rather than standalone predictions. Crucially, these models are designed to predict individual relational facts (atoms) as well as derived or combined facts, enabling queries that involve joins, instance-of reasoning, and hierarchical or compositional relationships. Potential applications include (i) ***cross-dataset integration***, executing queries over the union of multiple sources, e.g., $q(\mathcal{D}_1 \cup \mathcal{D}_2) = \mathcal{D}_3$, supporting schema alignment and conflict resolution; (ii) ***learned ETL pipelines*** as query, where extraction, transformation, and loading operations are specified and executed via learned query functions that preserve structure and type constraints; (iii) ***stateful query interaction***, interactive and incremental querying, where models maintain state across sessions and incorporate streaming updates into consistent results; and (iv) ***privacy-preserving federated querying***, where structured data remains locally stored but foundation models coordinate secure, aggregated answers without centralizing raw records. Realizing these applications requires models that represent schema, provenance, and uncertainty explicitly and support composable query invocation.

6.2 Open Challenges

Several foundational research challenges remain: (i) **heterogeneous query language support**, reconciling formal languages (SQL, SPARQL, GQL), programmatic interfaces, and natural language while preserving formal semantics; (ii) ***multi-query composition and optimization***, where a batch of queries $\{q_1, \ldots, q_n\}$ executes on the same dataset efficiently with shared intermediate computation; (iii) ***interactive query dialog***, formalizing how subsequent queries can reference prior results, e.g., $q_i(\mathcal{D}_2 \cup q_{i-1}(\mathcal{D}_2))$ while tracking provenance, consistency, and rollback semantics; (iv) ***designing training regimes*** that integrate large-scale pretraining, task-specific fine-tuning, and preference or safety optimization without catastrophic forgetting across structured domains; and (v) ***benchmarks for structured transfer and precision***, measuring cross-schema generalization, numeric and logical correctness, and grounding against authoritative sources; and (iv) ***systems and privacy constraints***, ensuring scalable retrieval, transactional consistency, and provable privacy guarantees.

Ethical Statement

There are no ethical issues.

Acknowledgments

This work is funded by the German Research Foundation (DFG) - SFP 1574 - 471687386, and BMBF ATLAS project (031L0304A).

References

[Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of databases*, volume 8. Addison-Wesley Reading, 1995.

[Arun and Kumar *et al.*, 2025] Arvindh Arun and Sumit Kumar *et al.* SEMMA: A semantic aware knowledge graph foundation model. In *Proc. of EMNLP*, 2025.

[Beeri and Vardi, 1984] Catriel Beeri and Moshe Y Vardi. A Proof Procedure for Data Dependencies. *Journal of the ACM (JACM)*, 31(4):718–741, 1984.

[Bommasani *et al.*, 2021] Rishi Bommasani *et al.* On the opportunities and risks of foundation models. *arXiv preprint*, 2021. https://arxiv.org/abs/2108.07258.

[Chen, 2016] Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.

[Codd, 1970] EF Codd. A relational model of data for large shared data banks. comm. acm 13, 6, 1970.

[Cui *et al.*, 2025] Yuanning Cui, Zequn Sun, Wei Hu, and Zhangjie Fu. KGFR: A foundation retriever for generalized knowledge graph question answering. *arXiv preprint*, 2025. https://arxiv.org/abs/2511.04093.

[Dwivedi *et al.*, 2025] Vijay Prakash Dwivedi *et al.* Relational deep learning: Challenges, foundations and next-generation architectures. In *Proc. of KDD, Vol. 2*, 2025.

[Erickson *et al.*, 2025] Nick Erickson *et al.* Tabarena: A living benchmark for machine learning on tabular data. *arXiv preprint*, 2025. https://arxiv.org/abs/2506.16791.

[Fagin, 1982] Ronald Fagin. Horn Clauses and Database Dependencies. *Journal of the ACM (JACM)*, 29(4):952–985, 1982.

[Fang and Xu *et al.*, 2024] Xi Fang and Weijie Xu *et al.* Large language models (LLMs) on tabular data: Prediction, generation, and understanding - a survey. *TMLR*, 2024.

[Feitelberg and Saha *et al.*, 2025] Jacob Feitelberg and Dwaipayan Saha *et al.* TabImpute: Accurate and fast zero-shot missing-data imputation with a pre-trained transformer. In *EurIPS 2025 Workshop: AI for Tabular Data*, 2025.

[Feng *et al.*, 2025] Yifan Feng, Shiquan Liu, and Xiangmin Han et al. Hypergraph foundation model. *arXiv preprint*, 2025. https://arxiv.org/abs/2503.01203.

[Fey and Kocijan *et al.*, 2025] Matthias Fey and Vid Kocijan *et al.* KumoRFM: A foundation model for in-context learning on relational data, 2025. https://kumo.ai/research/kumo_relational_foundation_model.pdf.

[Galkin and Zhou *et al.*, 2024] Michael Galkin and Jincheng Zhou *et al.* A foundation model for zero-shot logical query reasoning. *NeurIPS*, 2024.

[Galkin *et al.*, 2023] Mikhail Galkin *et al.* Towards foundation models for knowledge graph reasoning. *NeurIPS Workshop: New Frontiers in Graph Learning*, 2023.

[Hayler *et al.*, 2025] Adrian Hayler, Xingyue Huang, and Ismail Ilkan Ceylan *et al.* Of graphs and tables: Zero-shot node classification with tabular foundation models. In *New Perspectives in Graph Machine Learning*, 2025.

[He *et al.*, 2024] Xiaoxin He, Yijun Tian, and Yifei Sun *et al.* G-retriever: retrieval-augmented generation for textual graph understanding and question answering. In *NeurIPS*, 2024.

[He *et al.*, 2025] Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In *Proc. of KDD. Vol. 1*, 2025.

[Hogan *et al.*, 2021] Aidan Hogan *et al.* Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

[Hollmann *et al.*, 2023] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *ICLR*, 2023.

[Hollmann *et al.*, 2025] Noah Hollmann, Samuel Müller, and Frank Hutter *et al.* Accurate predictions on small data with a tabular foundation model. *Nature*, 2025.

[Hollmann *et al.*, 2025] Noah Hollmann *et al.* Accurate predictions on small data with a tabular foundation model. *Nature*, 2025.

[Huang *et al.*, 2023] Xingyue Huang, Miguel Romero, Ismail Ceylan, and Pablo Barceló. A theory of link prediction via relational weisfeiler-leman on knowledge graphs. *NeurIPS*, 2023.

[Huang *et al.*, 2025] Xingyue Huang, Pablo Barcelo, and Michael M Bronstein *et al.* How expressive are knowledge graph foundation models? In *ICML*, 2025.

[Huang *et al.*, 2025] Xingyue Huang *et al.* Hyper: A foundation model for inductive link prediction with knowledge hypergraphs. *arXiv preprint arXiv:2506.12362*, 2025.

[Kim and Lefebvre *et al.*, 2025] Myung Jun Kim and Félix Lefebvre *et al.* Table foundation models: on knowledge pre-training for tabular learning. *arXiv preprint*, 2025. https://arxiv.org/abs/2505.14415.

[Kim *et al.*, 2025] Jinwoo Kim, Xingyue Huang, and Krzysztof Olejniczak *et al.* Flock: A knowledge graph foundation model via learning on random walks. *arXiv preprint*, 2025. https://arxiv.org/abs/2510.01510.

[Klein and Biehl *et al.*, 2024] Tassilo Klein and Clemens Biehl *et al.* Salt: Sales autocompletion linked business tables dataset. In *NeurIPS Workshop*, 2024.

[Lee and Whang, 2025] Jaejun Lee and Joyce Jiyoung Whang. Structure is all you need: Structural representation learning on hyper-relational knowledge graphs. In *ICML*, 2025.

[Li *et al.*, 2025] Cheng-Te Li, Yu-Che Tsai, Chih-Yao Chen, and Jay Chiehen Liao. Graph neural networks for tabular data learning: A survey with taxonomy and directions. *ACM Computing Surveys*, 2025.

[Lin and Yan *et al.*, 2024] Tianqianjin Lin and Pengwei Yan *et al.* Langgfm: A large language model alone can be a powerful graph foundation model. *arXiv preprint*, 2024. https://arxiv.org/abs/2410.14961.

[Ma and Thomas *et al.*, 2025] Junwei Ma and Valentin Thomas *et al.* TabDPT: Scaling tabular foundation models on real data. In *NeurIPS*, 2025.

[Martens *et al.*, 2015] Wim Martens, Frank Neven, and Stijn Vansummeren. SCULPT: A schema language for tabular data on the web. In *Proc. of WWW*, 2015.

[Mitlöhner and Neumaier *et al.*, 2016] Johann Mitlöhner and Sebastian Neumaier *et al.* Characteristics of open data csv files. In *OBD*. IEEE, 2016.

[Omeliyanenko *et al.*, 2023] Janna Omeliyanenko *et al.* CapsKG: enabling continual knowledge integration in language models for automatic knowledge graph completion. In *ISWC*. Springer, 2023.

[Pan and Nayyeri *et al.*, 2025] Jiaxin Pan and Mojtaba Nayyeri *et al.* Towards foundation model on temporal knowledge graph reasoning. *arXiv preprint*, 2025. https://arxiv.org/abs/2506.06367.

[Pan and Razniewski et al., 2023] Jeff Z. Pan and Simon Razniewski et al. Large language models and knowledge graphs: Opportunities and challenges. *TGDK*, 2023.

[Peixian and Zhuang *et al.*, 2025] Ma Peixian and Xialie Zhuang *et al.* Sql-r1: Training natural language to sql reasoning model by reinforcement learning. In *NeurIPS*, 2025.

[Qin and Luo *et al.*, 2024] Zongyue Qin and Chen Luo *et al.* Relational database augmented large language model. *arXiv preprint*, 2024. https://arxiv.org/abs/2407.15071.

[Qu *et al.*, 2025] Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *ICML*, 2025.

[Ranjan and Hudovernik *et al.*, 2025] Rishabh Ranjan and Valter Hudovernik *et al.* Relational transformer: Toward zero-shot foundation models for relational data. In *EurIPS Workshop: AI for Tabular Data*, 2025.

[Robinson and Ranjan *et al.*, 2024] Joshua Robinson and Rishabh Ranjan *et al.* Relbench: A benchmark for deep learning on relational databases. *NeurIPS*, 2024.

[Sun and Zheng *et al.*, 2025] Jiankai Sun and Chuanyang Zheng *et al.* A survey of reasoning with foundation models: Concepts, methodologies, and outlook. *ACM Computing Surveys*, 2025.

[Tanna and Seth *et al.*, 2025] Aditya Tanna and Pratinav Seth *et al.* TabTune: A unified library for inference and fine-tuning tabular foundation models. *arXiv preprint*, 2025. https://arxiv.org/abs/2511.02802.

[Van Breugel and Van Der Schaar, 2024] Boris Van Breugel and Mihaela Van Der Schaar. Position: Why tabular foundation models should be a research priority. In *ICML*. PMLR, 2024.

[van, 2019] Wrangling messy CSV files by detecting row and type patterns. *Data Mining and Knowledge Discovery*, 2019.

[Vogel *et al.*, 2022] Liane Vogel, Benjamin Hilprecht, and Carsten Binnig. Towards foundation models for relational databases [vision paper]. In *NeurIPS: First Table Representation Workshop*, 2022.

[Vogel *et al.*, 2024] Liane Vogel, Jan-Micha Bodensohn, and Carsten Binnig. Wikidbs: A large-scale corpus of relational databases from wikidata. *NeurIPS*, 2024.

[Wang and Wang *et al.*, 2024] Zifeng Wang and Zichen Wang *et al.* Biobridge: Bridging biomedical foundation models via knowledge graphs. In *ICLR*, 2024.

[Wang and Wang *et al.*, 2025] Yanbo Wang and Xiyuan Wang *et al.* Griffin: Towards a graph-centric relational database foundation model. In *ICML*, 2025.

[Wehrstein *et al.*, 2025] Johannes Wehrstein *et al.* Towards foundation database models. CIDR, 2025.

[Wu *et al.*, 2025] Zhaomin Wu, Ziyang Wang, and Bingsheng He. Wikidbgraph: Large-scale database graph of wikidata for collaborative learning. *arXiv preprint arXiv:2505.16635*, 2025.

[Wu *et al.*, 2025] Fang Wu *et al.* Large language models are good relational learners. *arXiv preprint*, 2025. https://arxiv.org/abs/2506.05725.

[Yang and Huang *et al.*, 2025] Saisai Yang and Qingyi Huang *et al.* TableGPT-R1: Advancing tabular reasoning through reinforcement learning. *arXiv preprint*, 2025. https://arxiv.org/abs/2512.20312.

[Ye and Liu *et al.*, 2024] Han-Jia Ye and Si-Yang Liu *et al.* A closer look at deep learning methods on tabular datasets. *arXiv preprint*, 2024. https://arxiv.org/abs/2407.00956.

[Ye *et al.*, 2025a] Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at TabPFN v2: Understanding its strengths and extending its capabilities. In *NeurIPS*, 2025.

[Ye *et al.*, 2025b] Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at tabpfn v2: Understanding its strengths and extending its capabilities. In *NeurIPS*, 2025.

[Yin *et al.*, 2020] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. In *ACL*, 2020.

[Zhang and Zhang *et al.*, 2025] Jiani Zhang and Hengrui Zhang *et al.* CoddLLM: Empowering large language models for data analytics, 2025. https://arxiv.org/abs/2502.00329.

[Zhang *et al.*, 2025] Yucheng Zhang *et al.* Trix: A more expressive model for zero-shot domain transfer in knowledge graphs. In *LoG*, 2025.

[Zhu and Zhang , 2021] Zhaocheng Zhu and Zuobai Zhang . Neural bellman-ford networks: A general graph neural network framework for link prediction. *NeurIPS*, 2021.