arXiv:2608.28980v1 [cs.CL] 29 Aug 2026

The Illusion of Replacement: Rethinking Specialized Machine Learning Models in the Foundation Model Era

Kiyan Rezaee

Abstract

Can the specialized architectures that machine learning has traditionally built for structured data be replaced by language-based models? This question is examined through a review of 159 papers (2016–2026) across nine modalities, with predictive accuracy considered alongside structural representation and computation. A distinction is made between *performing a task* and *preserving and computing the structure that makes the task tractable*, and existing approaches are organized into eight representational regimes, ranging from language-only systems to fully specialized architectures. Language-mediated models are found to be highly competitive in specific settings, including extreme few-shot prediction, discretized symbolic tasks, textually annotated knowledge graphs, and large-scale single-modality pretraining. However, whenever structural representation or computation is directly evaluated rather than accuracy alone, no evidence of general architectural replacement is found. Instead, a recurring pattern is observed across independent research communities: when language alone is insufficient, the missing structure is reintroduced through a graph module, structural tokens, specialized attention, or another non-linguistic component. In this sense, specialization more often *relocates* than disappears. Moreover, although performance of language-based models is improved by scaling, whether the gap to a structure-aware architecture can eventually be eliminated remains untested.

1 Introduction

Three research communities with no shared authorship, benchmarks, or citation practice have recently made similar engineering decisions. Protein-language modeling reinjected Foldseek[1]-derived structure tokens into sequence-only models vocabularies once sequence-only representations proved insufficient [77, 91]; tabular learning paired language-model string embeddings with an explicit graph-attention scaffold rather than relying on the language model alone [45]; and chemistry reintroduced graph-based tools to SMILES-based models after they struggled to reliably track ring systems and branching [11]. In each case, a language-based representation was tried first, found insufficient under careful evaluation, and then supplemented with a non-linguistic structural channel. Three unrelated communities converging on the same corrective strategy is evidence of something more general about what language-based representations can and cannot carry.

The reason specialized representations existed in the first place is central to this pattern. Convolutional networks encode translation equivariance because natural images exhibit local spatial regularities [15]; permutation-invariant architectures were developed for sets because their elements have no canonical ordering [47]; and message-passing networks were framed within a broader theory of

[1]Foldseek is a fast structural-alignment tool that converts a protein's 3D backbone conformation into a short discrete alphabet, letting structural similarity be encoded and searched as if it were a sequence.

relational inductive biases [8]. These choices encode assumptions about the hypothesis space a model should explore, and the no-free-lunch theorem explains why such assumptions are unavoidable: without assumptions about the data-generating process, no learning algorithm can outperform all others across all possible tasks [99]. For decades, one of the field's principal strategies was therefore to place useful structural assumptions directly into the architecture.

Large language models challenge this design paradigm. At sufficient scale, a single pretrained model can process tables, graphs, time series, and other structured inputs after they have been rendered into a language-mediated representation, often without a task-specific architecture. The important question is not whether such a model can produce a plausible prediction but whether the structural assumptions that a specialized architecture would encode explicitly are actually acquired and exploited when they are instead described, demonstrated, or implied through language. We therefore ask: **To what extent can language-based models acquire and exploit structural inductive biases that have traditionally been encoded explicitly in specialized machine learning architectures?**

This question cannot be answered by predictive accuracy alone. It decomposes into several questions that can have different answers: Is language a general representation for structured data, or primarily an interface to systems that perform the underlying computation? Can an inductive bias that a specialized architecture enforces by construction instead be recovered by stating or demonstrating it in a prompt? If a language-based model matches a specialized model's accuracy, what exactly has been replaced? And can existing evidence distinguish genuine structural generalization from benchmark familiarity, encoding artifacts, or external computation? These questions motivate the framework developed throughout this review and are revisited directly in Section 7.

Read modality by modality, the empirical record appears difficult to reconcile. Positive results are real: TabLLM narrows the gap to gradient-boosted trees in the extreme few-shot regime [34]; text-based knowledge-graph completion can outperform structural embedding baselines [94]; and text-symbol spatial grids reach 84–91% accuracy where the corresponding pixel-based formulation reaches only 60–73% [3]. Yet controlled evaluations reveal a different picture. Tan et al. [81] remove the language-model component from three widely cited time-series models and find that the resulting models, despite having roughly 1000× fewer parameters, match or outperform the originals. Fatemi et al. [21] hold the graph, task, and model fixed while varying only the textual encoding, producing a 61.8% change in accuracy on a property that a genuinely structure-aware representation should preserve. Egressy and Stühmer [19] show that causal positional encoding makes decoder-only transformers non-permutation-invariant, meaning that a model can describe a structural property while its own computation violates it. Finally, Bordt et al. [10] show that some apparently strong few-shot tabular results cannot be separated from benchmark memorization. These findings point to different failure points in the same chain: a model may *describe* a structural regularity without *preserving* it, preserve information without *computing* the corresponding invariant, or achieve benchmark performance without efficiently *learning* the intended relationship.

This distinction exposes the tension of *generality* and *structural fidelity* in the literature. A language-based system can be genuinely general-purpose which the same model can process tables, graphs, and forecasts, while failing to implement the specific invariance or compositional operation required by a task. Conversely, it can match a specialized model's accuracy without matching its computational efficiency or robustness [81]. The term "replacement" therefore conceals several non-equivalent claims, which we formalize in Table 1. A model may functionally reproduce a specialized system's outputs without reproducing its structural computation; it may match accuracy without matching robustness; or it may orchestrate a specialized model effectively without replacing the computation that model

performs. This last distinction is increasingly important because recent work suggests that language models are often most effective when used to *orchestrate* specialized tools and pipelines rather than replace them [1, 61]. Meanwhile, specialized architectures continue to scale independently [5, 30, 68]. Orchestration is therefore a genuine capability, but it should not be conflated with architectural substitution.

Existing survey literature makes this distinction difficult to see because it is predominantly organized by modality or engineering design. Surveys of language models for tabular data [20, 100], graphs [42, 72], time series [112], chemistry [33], proteins [103], and point clouds [85] provide valuable modality-specific accounts, but offer little basis for asking whether the same phenomenon recurs across domains. Sun et al. [78] moves toward a cross-modal perspective by organizing methods across three modalities according to tokenization, architecture, pretraining, and adaptation. However, its taxonomy primarily describes how systems are constructed rather than what structural computations they can be shown to perform. Related surveys of in-context learning [17, 57, 115] focus on prompting and training strategies, and geometric deep learning [12] provides the theoretical foundations of specialized architectures without asking whether language can substitute for the inductive biases they encode. To our knowledge, no existing survey systematically evaluates language-mediated methods across modalities using a common distinction between describing, preserving, computing, and learning structural properties.

This review addresses this gap by examining language-mediated computation across nine modalities including tabular data, graphs, time series, vision, chemistry, code, knowledge graphs, point clouds, and protein structure, through a systematic corpus of 159 papers identified using the PRISMA-based [65] search described in Section 2.1. Our focus is not simply whether language models can achieve competitive performance, but where the structural information resides and what the model can actually do with it. We therefore separate two questions that are frequently conflated. The first concerns the *representational regime*: whether structure is encoded in language alone, supplied through demonstrations or generated programs, or provided by an external specialized component; this is captured by the eight-level taxonomy introduced in Section 3. The second concerns the *computational role* of that information: whether a model can *describe* a structural regularity, whether its representation *preserves* it, whether its computation *respects* it, and whether it can *learn* it efficiently from realistic data. Systems that contain a language model but retain jointly trained or specialized components, such as Chronicle [69], TabPFN [37], or PatchTST [62], provide important counterfactuals to claims of language-mediated replacement because their performance cannot be attributed to language alone.

In this work, results from the literature are connected to compare and distinguish genuine substitution from changes in interface, computation, or experimental conditions. It also motivates a more precise interpretation of the recurring pattern that when structural information or computation is repeatedly reintroduced through a non-linguistic channel after a language-mediated approach reaches its limits, the relevant question is which stage of the representation-to-computation pipeline has failed, and whether the missing capability can in principle be recovered within the language-mediated regime.

The remainder of the paper is organized as follows. Section 2 formalizes the distinction between describing, preserving, computing. Section 3 introduces the representational-regime taxonomy and the eight senses of "replacement" used throughout the review. Section 4 applies these frameworks across modalities, while Sections 5 and 6 examine whether current benchmarks can distinguish genuine structural generalization from contamination, format familiarity, and information leakage. Section 7 synthesizes the cross-modal evidence and evaluates whether the recurring convergence toward hybrid systems is best understood as a mechanistic limitation rather than a collection of

domain-specific failures. Finally, Section 8 identifies the central experiment that current evidence cannot yet resolve, holding information content and sample budget fixed while varying only the channel through which structural bias is supplied, and Section 9 concludes the work.

2 Scope, Methodology, and Problem Formulation

2.1 Survey methodology

We conducted this systematic literature review following the PRISMA [65] framework to identify, screen, and synthesize research on the use of large language models for structured and non-linguistic data. The literature search covered works published from 2016 through 2026 and was conducted across multiple complementary search streams covering tabular data, graphs, time series, vision and multimodal data, in-context learning and learning theory, universal representations, and reported limitations of language-based approaches. The search queries combined terms referring to large language models and foundation models with modality- and problem-specific terms, including `("large language model" OR "LLM" OR "foundation model") AND ("tabular" OR "table")`, `("large language model" OR "LLM") AND ("graph" OR "graph learning")`, `("large language model" OR "LLM") AND ("time series" OR "forecasting")`, `("large language model" OR "LLM") AND ("vision" OR "multi modal")`, and corresponding queries targeting in-context learning, structural generalization, inductive bias, universal representation, benchmark contamination, chemistry, code, knowledge graphs, point clouds, proteins, and related domains. Searches were complemented by backward and forward citation tracing and targeted searches for theoretical and recent works published during 2025–2026. Retrieved studies were screened for relevance based on their relationship to the review's central question and were retained only after bibliographic and full-text information was independently verified through resolvable scholarly sources, including arXiv, ACL Anthology, OpenReview, and publisher platforms. The final corpus comprises 159 papers, which form the basis of the taxonomy, cross-modal synthesis, and subsequent analysis.

2.2 Positioning relative to existing surveys

The literature contains substantial surveys of LLMs for individual modalities, including tabular data [20, 100], graphs [42, 72], time series [112], chemistry [33], protein sequences [103], and point clouds [85]; these studies primarily organize the literature by modality-specific architectures, representations, adaptation strategies, or downstream tasks. Most notably, Sun et al. [78] reviews LLM-based methods for tabular data, time series, and graphs, providing a broader cross-modal engineering taxonomy based on data representation, model architecture, pretraining, and adaptation. They mentioned that reducing a task to a language-mediated representation often loses explicit structural or relational inductive bias, without testing that claim directly.

The graph-specific surveys, Liu et al. [50], and, Wang et al. [96], treat graph foundation models on their own architectural terms without a language-substitution comparison. Mahowald et al. [55]'s formal-versus-functional-competence distinction is the closest interdisciplinary analogue to this review's central framework, a comparable "knowing versus doing" structure applied to linguistic competence rather than to structured non-linguistic data.

Related surveys of in-context learning [17, 57, 115], geometric deep learning [12] addresses complementary aspects of the problem but do not connect architectural inductive bias, representation,

computation, and learning within a common framework. No survey located in this search spans more than two or three of this review's nine modalities under one consistent framework, and none proposes an analogue of the four-way describe/preserve/compute/learn distinction. The contribution of this review is therefore the use of that common framework to synthesize evidence across modalities, connect empirical findings with learning theory, and identify the controlled experimental gap that remains unresolved (Sections 3.3, 6 and 8).

2.3 What the field actually disagrees about

Some disagreements in the literature are genuinely empirical, whereas others arise because different studies answer different questions while using similar language to describe their conclusions. The question of whether language can replace specialized machine learning spans, at minimum, four stages of the learning process. First, can a model *describe or recognize* a structural regularity, such as permutation invariance or relational structure? Second, does its language-based representation *preserve* the information required to exploit that regularity? Third, does the model's computation actually *respect* the relevant structural constraint, or can it describe a property correctly while violating it in its own predictions? Fourth, can the model *learn* and generalize that structure efficiently from a realistic amount of data? Vafa et al. [88]'s inductive-bias probe illustrates why these distinctions matter: a transformer achieves near-perfect accuracy on orbital-trajectory prediction while failing to internalize the underlying physical structure, breaking when evaluated on computationally different but physically equivalent targets. High task accuracy can therefore coexist with failure to recover the structure that would make that accuracy robust. Section 3 formalizes these four stages as the organizing framework for Section 4; here, the central point is that success at one stage should not be treated as evidence for success at another.

Four recurring disagreements can then be distinguished by their source: *empirical*, when competing claims are supported by measured results that have not yet been reconciled under shared conditions; *mechanistic*, when researchers agree on what a system achieves but disagree about why; *definitional*, when the same term refers to different substantive claims; and *evaluative*, when the dispute concerns whether a reported result would survive a level of scrutiny that has not yet been systematically applied. We separate these categories explicitly because they require different forms of resolution.

The first disagreement is empirical and concerns the scope of the substitution claim. One line of work advances the claim in a strong form: Song et al. [76] present language models as universal regressors, while Wang et al. [95] frame them as universal tabular classifiers. A second line of work tests this claim under controlled conditions and finds important qualifications. Removing the language-model component from three widely cited time-series forecasters does not reduce performance [81]; changing only the textual encoding of an otherwise fixed graph task produces a 61.8% change in accuracy [21]; and causal positional encoding makes standard decoder-only transformers provably non-permutation-invariant regardless of what those models can describe [19]. This is therefore an empirical and representational disagreement: both sides can point to measured results. As discussed in Sections 4 and 7, the evidence favors a narrower interpretation of the strongest universal claims once computational cost, information availability, and contamination are controlled. The important point here is that statements such as "language models are universal predictors" and "language models do not preserve structure" can currently both appear as conclusions in the literature because they are often established under different experimental conditions.

The second disagreement is mechanistic. Researchers broadly agree that specialized architectures can encode useful structural constraints, but disagree about whether those constraints must be

embedded in the architecture itself. Our working hypothesis is that structural assumptions generally need to be reflected in the computation rather than merely supplied at inference time. This position, however, is not uncontested. Mittal et al. [60] show that a standard transformer, when supplied with an appropriate *inferential* bias through training and querying rather than an architectural bias, can match custom permutation-invariant architectures on exchangeable sequence modeling. This constitutes the most direct counter-evidence to the stronger form of our working hypothesis identified in the search. It suggests that, for at least some structural properties, the distinction between architectural and inferential bias may matter more than the distinction between architectural and language-mediated bias. Whether this generalizes across the modalities and structural properties considered here remains an open question addressed in Section 6.

The third disagreement is definitional. Work on language-model-driven feature engineering, model search, and pipeline construction [1, 61], as well as position papers arguing that graph foundation models are "already here" [56], may use "replace" to mean that a language model now performs a role previously carried out by a human or specialized system. In this review, by contrast, "replace" refers to whether the language model's own computation implements the structural function, rather than delegating that function to an external tool. We do not attempt to privilege one usage over the other. Instead, Section 3 makes the distinction explicit, and Table 1 separates eight senses of "replacement" precisely because many apparent disagreements are disagreements about which claim is being made.

The fourth disagreement is evaluative and, in at least one prominent case, has already been addressed through direct re-examination. Tabula-8B was originally reported to outperform XGBoost and TabPFN by 5–15 percentage points in few-shot tabular prediction [23]. A subsequent re-analysis found that 92.2% of the reported gain could be recovered by instruction-tuning the same base model without tabular exposure, alongside evidence of train–test contamination that standard deduplication had failed to detect [28]. Similar concerns have been raised through subsequent re-analyses [10, 75]. Yet comparable audits have not been performed across most tabular and time-series benchmarks in Table 3, or across the graph, vision, and adjacent-modality benchmarks considered in this review. For these settings, the question of how much positive evidence would survive equivalent scrutiny remains open rather than being settled by the cases that have already been audited.

These disagreements are therefore not constructed to organize the discussion. The empirical disagreement requires controlled experiments such as the one specified in Section 8; the mechanistic disagreement requires further theoretical analysis; the definitional disagreement requires explicit terminology; and the evaluative disagreement requires audits that have not yet been conducted systematically. Understanding why these particular questions emerged requires examining the representational history from which the current tension arose.

2.4 Historical evolution

The four disagreements above are rooted in a representational history that predates LLMs. Figure 1 summarize this history as a sequence of eras defined by what researchers believed a given representational strategy could substitute for. The eras overlap, and the transitions shown in both should therefore be interpreted as approximate.

Between 2016 and 2021, the dominant strategy was to encode known structure directly into the computation graph, because doing so remained one of the most reliable ways to generalize from limited domain-specific data. Group-equivariant convolutions incorporated symmetry into visual

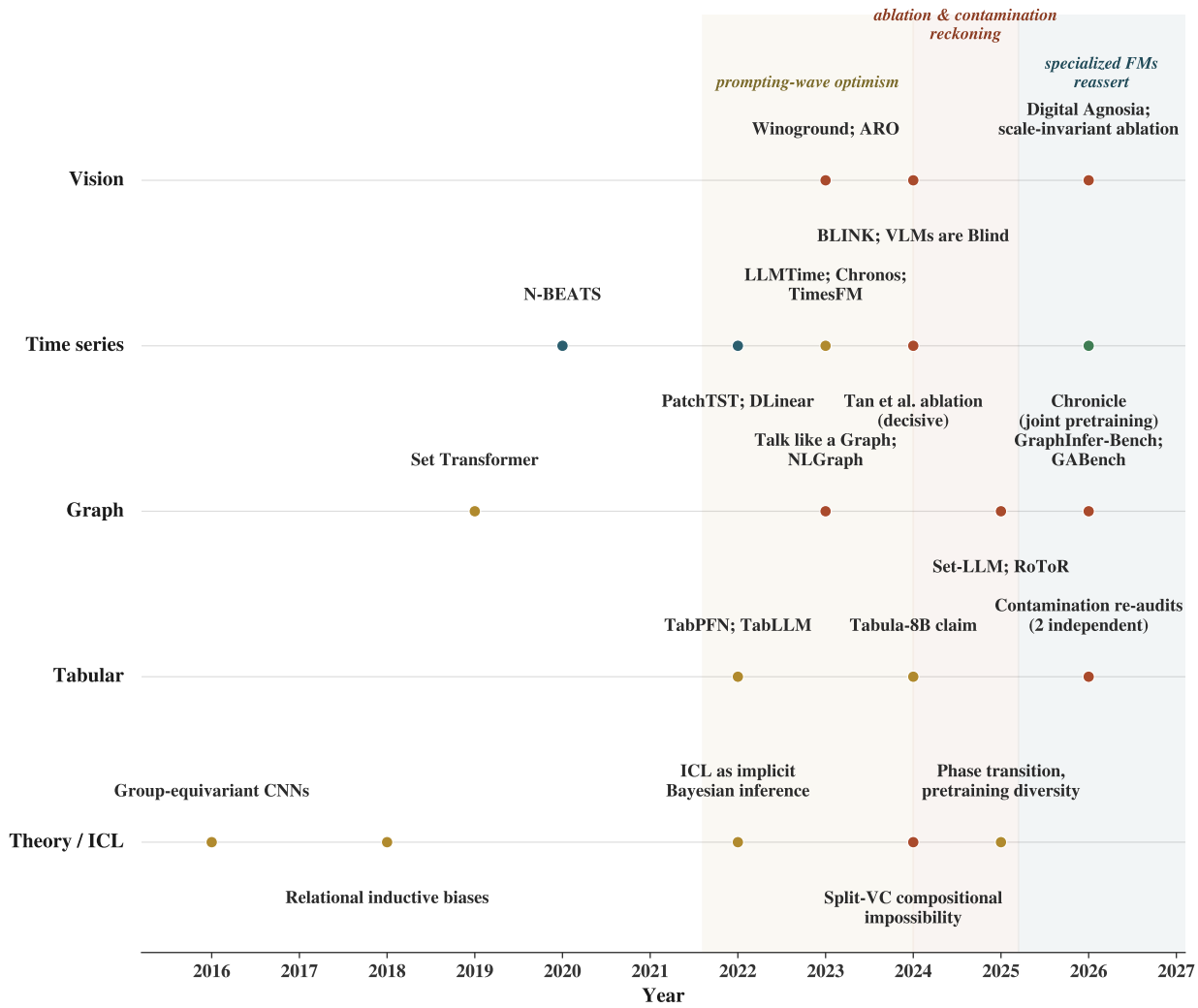

Figure 1: Historical evolution of research on language-mediated learning of structured data from 2016 to 2026. The timeline traces the transition from explicitly encoded structural inductive biases, through language-based representations and prompting, to controlled ablations, contamination analyses, theoretical investigations, and renewed interest in specialized and jointly pretrained foundation models.

representations [15]; permutation-invariant architectures were developed for sets [47]; and message-passing networks were unified under a broader framework of relational inductive biases [8]. In parallel, another line of work pursued generality through architecture rather than language. Perceiver IO [40] and Gato [71] demonstrated that a common architecture could process substantially different input and output types, showing that the aspiration toward general-purpose models predates, and does not require, a language-based interface. What these approaches could not readily provide was transfer of a learned solution across modalities: each new domain could still require substantial architectural specialization. That cost created the conditions for the next era's central hypothesis.

From roughly 2022 onward, researchers increasingly investigated whether this specialization could be reduced by representing structured objects as text and prompting or lightly fine-tuning pretrained language models. Examples include TabLLM [34] for tabular data, LLMTime [31] for time series, and early graph-verbalization approaches [21, 92] for relational data. The appeal was straightforward:

if a sufficiently capable pretrained model could acquire useful structural behavior from pretraining or contextual information, some of the architectural specialization of the preceding era might no longer be necessary. The optimism of this period is reflected in position work arguing that graph foundation models were already emerging as a distinct paradigm [56]. What many early results did not establish, however, was whether the language model itself was responsible for the observed gains, rather than the information supplied to it or the particular format in which that information was encoded.

This question motivated the next phase of the literature. During 2024–2025, controlled ablations increasingly tested whether the language-model component was actually load-bearing. Most prominently, removing or reinitializing the language-model component of three widely cited time-series forecasters did not reduce performance [81]. In parallel, contamination audits examined whether apparent few-shot advantages instead reflected information already present in pretraining [10, 27]. Theoretical work supplied a complementary account of why language-model components might fail to provide the expected structural advantage: Kozachinskiy et al. [46] established a width- and precision-independent ceiling on the compositional operations achievable by a single attention layer. Together, these results changed the evidentiary standard. High benchmark accuracy remained important, but it was no longer sufficient evidence that a language model had acquired the structural properties required by the task.

The field's subsequent development reflects this more cautious standard. By 2025–2026, specialized non-linguistic architectures continued to scale independently, including TabPFN-2.5, TabICLv2, GraphBFF, and TiRex [5, 9, 30, 68]. A different strategy also emerged: instead of adapting a pretrained language model to structured data after the fact, Chronicle jointly pretrains language and time-series representations through shared attention blocks [69]. This represents a distinct hypothesis from both explicit specialization and post hoc language-mediated adaptation. At the same time, cases in which language-mediated systems provide clear practical value increasingly involve orchestration, using language models to direct specialized models or tools, rather than replacing the underlying computation (Section 4.6).

The resulting trajectory is therefore better understood as a single question becoming progressively more precise. The field first asked whether structured data could be expressed through language; it then asked whether language models could perform competitively on such representations; subsequently, it began asking why they succeeded or failed; and it now faces the more precise question of what, exactly, is transferred, preserved, computed, and learned through a language interface. The disagreements identified above are the unresolved residue of this progression. Section 3 introduces the framework used throughout the remainder of the review to analyze them.

3 A Unifying Taxonomy: Representation Regimes and the Four-Way Distinction

A central challenge in interpreting this literature is that "language-based" does not describe a single modeling regime. A system that uses an LLM to generate Python code and invoke XGBoost, a model that projects visual tokens into an LLM's embedding space, and a model that receives a structured object only as plain text may all be described as LLM-based, despite relying on fundamentally different mechanisms. Consequently, evidence that supports one regime is often used to make claims about another. Resolving this requires two taxonomies that answer two different questions. The first, developed in Section 3.1, is structural: it asks *where* the computation that must exploit a task's

structure actually happens. The second, developed in Section 3.3, is functional: given that a result occupies a particular structural regime, it asks *what* the system does with the information available there: describes it, preserves it, computes over it, or learns from it. Section 3.2 sits between the two, because it is impossible to say what a result has "replaced" without knowing both where its computation is located and what it does there.

3.1 Eight representational regimes

We organize existing approaches into eight representational regimes, summarized in Figure 2. A regime is defined by a single question: given a task that requires exploiting some structural property of the input, where does the computation that exploits it actually take place? A system can serialize a graph as plain text while using an ordinary decoder-only transformer trained by next-token prediction, and the regime question is still open until we ask whether that transformer's own forward pass, an artifact it generates, an external tool it calls, or a separate encoder entirely is responsible for the structural computation.

Because this question has three qualitatively different answers, the eight regimes group into three tiers rather than tracking one quantity end to end. In Levels 1–4, the language model's own forward computation over natural-language tokens is asked to carry out the structural computation directly, and what changes across these four levels is how much linguistic scaffolding is supplied to that same computation: nothing beyond the task description (Level 1); a structured serialization of the object itself (Level 2); an explicit natural-language statement of the relevant structural property (Level 3); and worked demonstrations (Level 4). This is the one genuinely monotonic sub-ordering in the taxonomy, and it is monotonic in a narrow, specific sense that each level strictly increases the task-relevant information available to the same computational mechanism, not the quality of that mechanism's reasoning. In Levels 5–6, responsibility for computation moves outside the language model's own forward pass: at Level 5 the model generates an executable artifact, typically code, that performs the transformation; at Level 6 the model instead searches over, selects, or calls an external tool or specialized model [1, 61, 109]. This boundary overlaps with, but is not identical to, the reasoning-versus-tool-augmentation axis that Mialon et al. [59] use to organize a broader survey of augmented language models. Their axis asks whether a capability comes from a reasoning strategy or an external tool call, largely independent of any particular non-linguistic modality, whereas the regime ladder asks specifically where the computation carrying a task's *structural* bias is located, which is why Level 7, a case their reasoning/tool axis does not distinguish from ordinary multimodal input, receives a dedicated tier here rather than being folded into either of theirs. Level 7 is not a further increment along the Level 5–6 axis in any case. Here the structural computation is performed by a separate, typically pretrained encoder (a visual tokenizer [87], a graph-encoder projector [13, 82], or a point-cloud encoder [104]), and language never had causal access to the structural transformation in the first place; it operates only on the encoder's output. Level 8 removes the language model entirely. The three-tier grouping in Figure 2 reflects this directly: it is a labeled ordinal scale tracking where computational responsibility has moved, not a claim that every adjacent pair differs by the same underlying quantity. Level 7 in particular should not be read as "more" of whatever Level 5 or 6 are; it is a different mechanism, not a further degree of the same one.

Two boundaries matter most for interpreting the evidence in Section 4. The first separates language-based reasoning (Levels 1–4) from language-based orchestration (Level 6): an LLM that generates feature transformations, searches over model configurations, or constructs and refines machine-learning pipelines demonstrates that language can effectively *direct* specialized machine learning, but

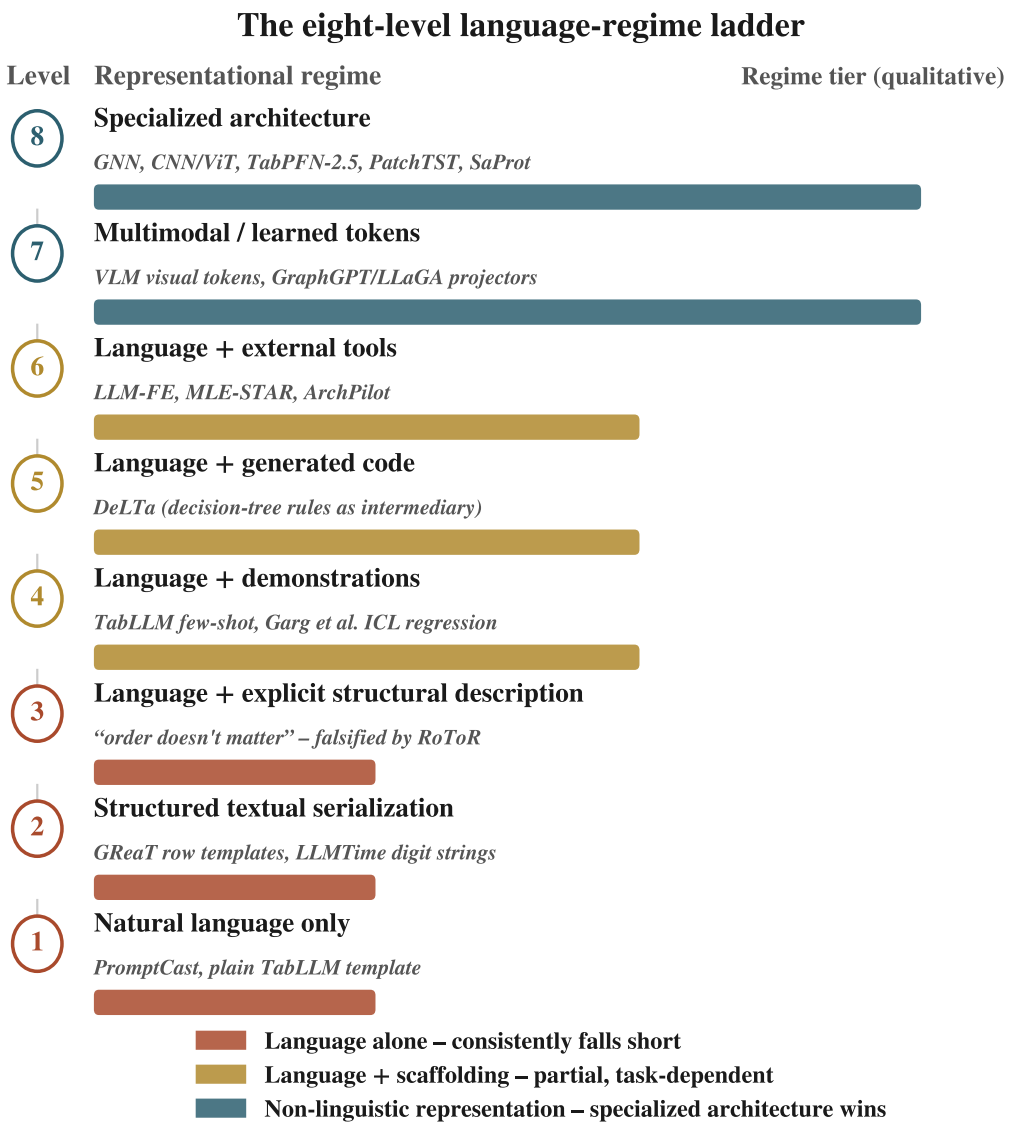

Figure 2: The eight representational regimes considered in this review. The regimes are grouped into three broad tiers according to the role of language: language-only approaches (Levels 1–3), language-based approaches with additional scaffolding or specialized components (Levels 4–6), and approaches in which the task-relevant structure is represented outside natural language (Levels 7–8). The boundary between Levels 6 and 7 marks the transition from language as the computational medium to non-linguistic representations as the primary carrier of structural information.

the structural computation remains with the specialized models or tools it invokes, and this is evidence for orchestration, not for replacement. The second separates language-mediated representations (Levels 1–6) from non-linguistic ones (Levels 7–8): a system that contains an LLM is not thereby a language-mediated system in the sense this review tests, if the structural information it exploits is carried by a separate encoder the LLM never processes as text.

The regimes are not mutually exclusive at the level of a paper. Table 2 tags several methods at two levels simultaneously, and it does so because the underlying results genuinely occupy two regimes rather than one: TabLLM [34] reports both a plain-serialization baseline and a few-shot

demonstration condition (Levels 2/4); SimKGC's textual scoring model sits at the same boundary [94]; LLM-FE [1] both generates code and searches with it (Levels 5/6). This is resolved the same way Section 2.1 resolves it for the review as a whole that the unit classified is the reported result, not the paper, so a single method can legitimately occupy different cells for different components or conditions. The more informative boundary case is Time-LLM [43], marketed and named around "reprogramming" time series through language. Table 2 classifies its actual mechanism at Level 7, not Levels 1–4, because the structural computation it performs on the input sequence does not run through interpretable language tokens at all, and the ablation evidence reviewed in Section 4.3, that removing its language-model component does not hurt performance [81], is exactly the pattern this classification predicts. if the mechanism were genuinely Level 1–4, removing the language component should remove the computation itself, not leave it intact. This is the sense in which the taxonomy is explanatory and it generates a testable expectation before the ablation is run.

The same boundary recurs across every modality examined in Section 4, not only time series: tabular prediction separates plain serialization (Level 2) from numeric in-context learning with no language at all (Level 8, the TabPFN family); graph learning separates textual encodings of the same graph (Level 2) from learned graph-encoder projections (Level 7, GraphGPT, LLaGA); and vision separates frontier vision-language models operating on learned visual tokens (Level 7) from text-symbol grid representations in which structure has already been discretized into language before the model ever sees it (Levels 2–3). That the same boundary does the same explanatory work in tabular, graph, time-series, and vision literatures sharing no common authorship or terminology is the basis for treating it as a property of the representational regime rather than of any single modality.

The same check, extended to the five adjacent modalities in Section 4.5. Point clouds and knowledge graphs reproduce the boundary directly: PointLLM pairs a specialized point-cloud encoder with a language interface, placing the structural representation at Level 7 while language handles only captioning and interaction [104], and knowledge-graph agents with extensive tool access remain at Level 6 regardless of that access, continuing to underperform specialized graph models at Level 8 [80]. Protein structure produces a genuine boundary case that Su et al. [77] and Wang et al. [91] inject Foldseek-derived structure tokens (the output of a separate, non-linguistic structural-alignment tool) directly into a sequence model's vocabulary, so the structural computation originates at Level 7 but is consumed as ordinary sequence tokens at Level 2. The eight discrete levels do not separate this cleanly, and they should not be forced to: it is a compositional case, a Level 7 computation feeding a Level 2 representation, read the same way Section 3.1 already reads TabLLM and SimKGC above, and it sharpens rather than weakens the taxonomy by showing what a compositional reading looks like when the composition happens inside the vocabulary rather than across separate components. Code is the weakest fit of the nine modalities: the reviewed evidence [32] concerns whether models correctly simulate program execution, a describe-versus-compute question in the sense of Section 3.3 rather than a regime-boundary one, and no non-linguistic Level 8 baseline is reported for comparison. For code, the regime boundary is therefore untested rather than confirmed, and it is reported here as such.

3.2 What "replace" means

Knowing which regime a result occupies is not sufficient to say what, if anything, it has replaced. "Replacement" is not one claim: Table 1 distinguishes eight senses, and they do not reduce to points on a single scale.

Four of the eight track distinct properties a specialized model has: functional (predictive performance),

Table 1: Eight distinct senses in which a language-mediated model could be said to "replace" a specialized architecture (Section 3). Treating these as one claim is the second most common overclaim identified in this review, after conflating representational levels (Figure 2).

Notion	Definition	Evidence status
Functional	Comparable predictive performance	Sometimes, few-shot only
Statistical	Comparable sample efficiency	No: Yang et al. [105]'s bounds and the tabular dimensionality result agree
Computational	Comparable FLOPs/latency for equal accuracy	No: up to 1000× worse for no gain [81]
Robustness	Comparable OOD/perturbation robustness	No: spatial fragility [74]; domain fragility in TSFMs
Structural	Comparable exploitation of the relevant bias	No: Yoon et al. [108]; Egressy and Stühmer [19]; Kozachinskiy et al. [46]
Practical	Comparable deployment cost, latency, calibration	No: orders of magnitude more expensive per prediction
Universal	Works across broad task classes without redesign	Partially: language's one genuine structural advantage
Orchestration	Directs or composes specialized models effectively	Yes, increasingly: a distinct Level-6 capability, not replacement

statistical (sample efficiency), computational (FLOPs and latency for equal accuracy), and practical (deployment cost, latency, calibration). A language-mediated system can match a specialized model on any one of these without matching it on the others. Tan et al. [81]'s ablated forecasters match the functional performance of their language-mediated counterparts at roughly 1000× fewer parameters, which is simultaneously evidence *against* computational replacement for the original models (they needed far more compute for the same result) and silent on statistical replacement (neither variant was tested for sample efficiency against a matched-bias architecture). Robustness and universality are broader claims again, concerning behavior outside the training distribution and generality across task classes respectively, and are correspondingly rarer to find satisfied even where the narrower properties hold.

Structural replacement concerns whether a language-mediated system reproduces the inductive bias that a specialized architecture would explicitly encode. In other words, similar outputs alone are not sufficient: the question is whether the underlying structural principle is also preserved. This notion has an asymmetric relationship with the other seven forms of replacement. That asymmetry is not simply an empirical pattern found in the reviewed literature; the two directions of the relationship can each be justified by an independent argument. The forward direction follows the same generalization-bound logic that motivates encoding inductive bias architecturally in the first place (Section 2.4): a computation that genuinely respects a structural constraint operates over an effectively smaller, constrained hypothesis class, and standard results relate that constraint directly to sample efficiency and to correct behavior across the full orbit of structure-preserving transformations, not merely the specific instances a benchmark happens to test [79, 105]. Structural replacement is therefore a principled basis for expecting the other properties to follow, grounded in the same complexity argument that justifies architectural bias generally specific to this review. The reverse direction is not inductive at all but a basic identifiability point: an output value alone

underdetermines the mechanism that produced it. Matching a specialized system's accuracy on a fixed evaluation set is equally consistent with genuine structural computation, with pretrained world knowledge the comparison did not control for [10], with a representation that preserves the needed information without the model computing anything invariant over it [19], or with an information advantage unavailable to the specialized baseline [94]; no quantity of matched output values on a finite test set can distinguish among these, and only a direct test of the invariance itself, of the kind discussed in Section 6.1, can. Section 4 shows this is not a hypothetical concern: nearly every reviewed case of apparent functional or statistical replacement, when a direct structural test has actually been run, either fails it or has simply never been subjected to one.

The eighth sense, orchestration, is not a weaker form of replacement but a different relationship, and treating it as a point on the same scale is the source of much of the field's loosest rhetoric. When a language model searches over feature transformations [1] or refines a modeling pipeline [61], it satisfies none of the other seven senses, because the specialized computation it directs is not performed by the language model at all; this is Level 6 of Section 3.1, and it corresponds to the sense of "replace" that position pieces describing graph or tabular foundation models as already here typically have in mind [56]. That usage is locally coherent (something has changed about who designs the pipeline), but it answers a different question than the one this review asks, which is whether the language model's own computation implements the structural function, not whether a language model is present somewhere in the loop that produces it.

A related and equally common confusion is treating the disappearance of a *named* specialized architecture as evidence that specialization itself has disappeared. It has more often relocated. Su et al. [77] and Wang et al. [91] remove nothing from a protein language model's sequence-only design without also reinjecting Foldseek-derived structure tokens into its vocabulary; Kim et al. [45] pairs language-model string embeddings with an explicit graph-attention scaffold rather than a bare LLM; and chemistry language models trained directly on SMILES strings needed graph-based tools reattached once sequence-only representations proved unable to track ring systems and branching [11]. In each case, a specialized, non-linguistic component was removed from the headline architecture, and a specialized, non-linguistic component was added back somewhere else in the system: in the vocabulary, in an attention scaffold, in a reattached tool. None of these is evidence of representational, architectural, or computational replacement in the senses defined above; **each is evidence that specialization remained exactly as necessary as before, relocated to a place the system's name no longer advertises**.

3.3 The central distinction: describe, preserve, compute, learn

Where a result sits in Section 3.1 says nothing about what the system does with the information once it has it. A result can occupy Level 4 (language, with demonstrations, as the computational medium) and still fail a task for any of several unrelated reasons, and distinguishing those reasons requires a second set of questions. Claims that a model "understands" a structure conflate at least four of them:

(i) **Describe.** Can the model correctly state or explain a structural property, such as permutation invariance or graph isomorphism, independent of any particular input? This is a claim about declarative knowledge, evaluable by asking the model to explain the property in the abstract, and it is the weakest of the four: a model can pass it while being given, on a specific instance, a representation that does not contain the information the property requires.

(ii) **Preserve.** After a structured object is serialized or otherwise transformed for a specific instance, does the resulting representation retain the information the target computation needs, not all information about the object, but the information relevant to the property in question?

(iii) **Compute.** Given a representation that does preserve the relevant information, does the model's forward computation implement a function that actually respects the structural property, or can it violate the property in its output even when the input was sufficient to respect it?

(iv) **Learn.** Given a realistic sample and compute budget, can the model acquire and generalize the relevant structural function, as opposed to needing substantially more data than an architecture that encodes the corresponding bias directly? This is a claim about sample and compute efficiency, not about whether learning happens at all.

These four properties are best read as dimensions a system can satisfy in any combination, not as a strict sequence in which each stage is a prerequisite for the next, and the literature already documents failures in every direction. Describe does not imply preserve: Fatemi et al. [21] hold the graph, task, and model fixed and vary only the textual encoding, producing a 61.8-point swing in accuracy that is not explicable by any change in what the model can describe (the description is unaffected) but only by what the specific encoding did or did not preserve. Preserve does not imply compute: Egressy and Stühmer [19] show that causal positional encoding makes standard decoder-only transformers provably non-permutation-invariant regardless of whether the input representation preserves the relevant relational information, so a model can receive everything it needs and still compute the wrong function. Compute does not require describe: mechanistic evidence shows transformers implement specific computational procedures, induction mechanisms for copying, task-level representations for simple mappings, gradient-descent-like computation in linear attention [2, 64, 89], with no accompanying natural-language account of what they are doing. And learn does not require any of the other three to be separately established, as when transformers acquire simple function classes purely from in-context examples [24]. Reporting only one of these four properties and treating it as evidence for the others is, together with conflating regimes in Section 3.1, the most common source of overclaiming identified in this review. Theoretical results reviewed in Section 6 correspond to the computation and learning questions specifically: architectural expressivity and computational limitations address (iii), while sample-complexity and phase-transition results address (iv) [26, 46, 79, 105].

The relationship between this distinction and the eight regimes is complementary. The regime ladder is a structural taxonomy and it asks where the computation is located. The describe/preserve/compute/learn distinction is a functional taxonomy: given a location, it asks what the system actually does there. The two are logically independent: a Level 2 serialization can fail on preservation alone (Fatemi et al.'s result), while a Level 7 learned-encoder system can succeed on preservation and computation simultaneously because a separate, specialized network is responsible for both, with the language component contributing nothing to either. This is precisely why the two taxonomies are necessary together: regime alone would misclassify Time-LLM as a language-computation success because language is nominally present, and the four-way distinction alone has no way to explain *why* removing that presence does not remove the computation, since it never states where a given describe, preserve, compute, or learn result is actually coming from.

Figure 3 reports the resulting synthesis across the modalities in Section 4: descriptive competence is close to uniform, while the fraction of cases in which preservation, computation, and learning are

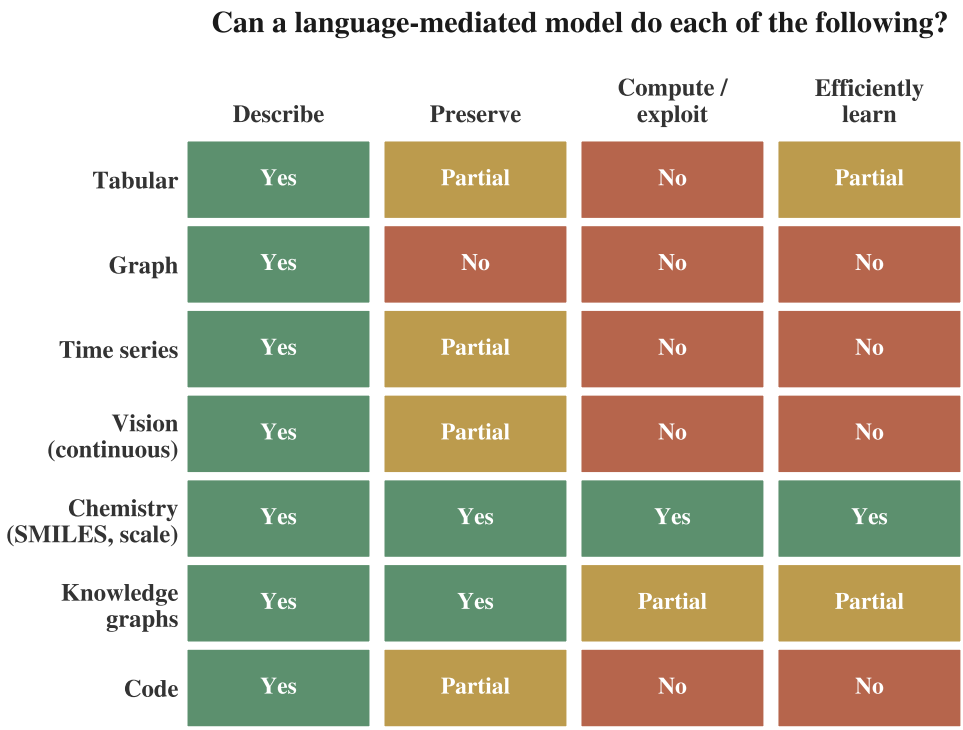

Figure 3: The four-stage distinction applied across the modalities reviewed in Section 4. Cells represent the qualitative evidence synthesized in this review rather than meta-analysis. The framework separates descriptive competence from representation preservation, structural computation, and efficient learning, allowing evidence at each stage to be evaluated independently.

also established falls sharply and unevenly. The figure should be read as a synthesis of the evidence reviewed later no meta-analysis.

This framework does not imply that language-mediated models are incapable of structural computation in general; Section 6 reviews theoretical results establishing exactly when they can. The narrower claim is that whether a model computes or merely describes a given structure is task-, representation-, and architecture-dependent, and is not reliably predicted by model scale or language fluency alone, which is why Section 4 evaluates each of the four properties separately for every modality, rather than inferring three of them from evidence about the fourth.

4 Major Methodological Families: A Modality-by-Modality Synthesis

We apply the same analytical template across modalities: first, we identify the specialized architecture and inductive bias traditionally used for the task; second, we examine the language-mediated alternative and determine which representational regime it actually occupies (Section 3.1); third, we assess the strongest positive evidence for that alternative alongside controlled, adversarial, or contamination-audited evidence that qualifies it; fourth, we identify what specialized computation remains once the language component is isolated; and finally, following the distinction in Section 3.3, we ask whether the evidence supports the claims that the system can describe, preserve, compute,

Table 2: Representative methods across the four core modalities, focusing on their representational regime and principal limitation. The representation levels follow the taxonomy in Figure 2.

Modality	Method	Representation	Main limitation
Tabular	TabPFN-2.5 [30]	Numeric ICL (L8)	Not language; limited to small–medium tables
	TabICLv2 [68]	Numeric ICL (L8)	Not language; primarily classification-focused
	TabLLM [34]	Text serialization (L2/4)	Overtaken by tree-based models beyond ∼8 shots
	Tabula-8B [23]	Text serialization (L4)	Most reported advantage traced to contamination [28]
	LLM-FE [1]	Language + code (L5/6)	Not a language-native predictor; delegates to specialized search
Graph	GraphBFF [9]	Numeric transformer (L8)	Not language; relies on industrial/proprietary data
	Talk like a Graph [21]	Text serialization (L2)	Performance varies substantially with encoding choice
	GraphGPT [82]	Learned projector (L7)	Structural information is carried by a non-linguistic encoder
	SimKGC [94]	Text serialization (L2/4)	Gains rely on textual side information rather than graph topology
Time series	Chronos [4]	Numeric tokenization (L8)	Not language; the "language" framing is metaphorical
	TiRex [5]	Numeric, recurrent (L8)	Demonstrates the value of specialized architecture rather than LLM lineage
	Time-LLM [43]	Reprogrammed embeddings (L7)	Ablated version outperforms the full model [81]
	Chronicle [69]	Joint pretraining (L8)	Small scale
Vision	CNN/ViT specialists	Pixel/patch input (L8)	Domain-specific; lacks a general language interface
	GPT-4V / frontier VLMs	Learned visual tokens (L7)	Weak performance on several low-level visual reasoning tasks [22, 70]
	SpatialVLM	Learned tokens + specialist-CV labels (L7)	Depends on specialized vision models for supervision
	Text-symbol grid encoding [3]	Structured serialization (L2/3)	Limited to layouts that can be discretized into symbolic representations

or learn the relevant structure. Table 2 summarizes the resulting classification across modalities, and Section 4.8 synthesizes the mechanisms that recur across them.

4.1 Tabular Data

Tabular learning has traditionally relied on tree-based ensembles and, more recently, transformers pretrained on synthetic priors. Both approaches encode inductive biases suited to sparse and nonlinear feature interactions rather than the smooth sequential structure on which language models

are pretrained. TabPFN [37] provides a strong non-linguistic baseline: it performs in-context learning directly over feature–value pairs, without language, and can match tuned gradient-boosting ensembles on small datasets at low inference cost. TabPFN-v2, TabPFN-2.5, and TabICLv2 extend this paradigm to datasets with up to 10,000 rows [30, 38, 68]. Thus, the baseline against which language-mediated approaches should be compared is itself a strong general-purpose learner rather than a narrow classical model.

Language-mediated approaches show their clearest advantage in the opposite regime: when labeled data are extremely scarce. TabLLM [34] and LIFT [16], which serialize rows and use either prompting or few-shot demonstrations (Levels 2 and 4), can exploit pretrained knowledge when only a handful of labeled examples are available. The advantage, however, is narrow. As more labeled data become available, gradient-boosted trees overtake LLM-based prediction while requiring substantially less computation [39]. More importantly, stronger claims of broad superiority have not survived independent scrutiny. Tabula-8B initially reported a 5–15 percentage-point advantage over XGBoost and TabPFN in few-shot settings [23]; Gorla and Puduppully [28] subsequently found that 92.2% of the reported improvement could be recovered by instruction-tuning the same base model without tabular exposure, while also identifying train–test contamination that standard deduplication had failed to detect. Independently, Silvestri et al. [75] identified contamination in four of eight widely used tabular benchmarks, and Bordt et al. [10] showed that LLMs can memorize popular benchmarks while performing poorly on genuinely in-context statistical tasks as dimensionality increases.

Broader evaluations further narrow the positive claim. Using 142 datasets spanning IID, temporal, and grouped distribution shifts, Purucker et al. [67] found tabular foundation models to be strongest in the small-to-medium IID regime that dominates existing benchmarks, while tree-based and deep-learning methods remain competitive or superior under other conditions. A separate, as yet unreplicated study reports that LLM performance degrades systematically with dimensionality while classical baselines remain comparatively stable [25].

Tabular data consequently illustrates a narrow form of functional replacement. TabLLM and LIFT demonstrate comparable predictive performance in extreme few-shot settings, but only within a regime in which strong non-linguistic baselines do not necessarily have the same advantage. Tabula-8B's stronger replacement claim did not survive contamination analysis, and no study in this modality directly demonstrates structural replacement. Thus, the evidence supports descriptive competence and, under restricted conditions, functional performance; it does not establish that language-mediated systems preserve or compute the same inductive biases as specialized tabular learners, nor that they learn those biases more efficiently.

4.2 Graph-Structured Data

Graph neural networks encode permutation invariance by construction that graph properties should remain unchanged under permutations of node labels or input ordering, and message-passing architectures are designed to respect this constraint. Egressy and Stühmer [19] show why decoder-only language models do not inherit this guarantee automatically. Causal positional encoding and input ordering introduce dependencies that permutation-invariant computation must eliminate. The failure is empirical as well as architectural. Holding the graph, task, and model fixed while changing only the textual encoding, Fatemi et al. [21] report a 61.8-point swing in accuracy. Thushalika et al. [86] similarly find reliable graph-isomorphism detection under one node labeling but substantial degradation after relabeling the same graph. Herbst et al. [35] further show that fine-tuning does not simply eliminate this sensitivity that larger non-fine-tuned models can be more robust to serialization

changes, whereas fine-tuning may reduce sensitivity to node relabeling while increasing sensitivity to formatting and structural changes. The issue is therefore not merely the choice of prompt; the representation itself may fail to preserve the invariances required by the task.

Strong results from Level 7 systems do not contradict this conclusion. GraphGPT and LLaGA route structural information through learned graph encoders that project graph representations into the LLM's embedding space [13, 82]. Their success therefore provides evidence for specialized structural encoders coupled to language models, not for natural language as the computational substrate of graph reasoning. Recent comparative evaluations reinforce this interpretation: GraphInfer-Bench finds LLM-based approaches weaker than conventional GNNs on graph-comparison tasks [66], while a large agentic benchmark shows that LLM agents with extensive tool access (Level 6) continue to struggle with sophisticated graph analysis [80].

Knowledge-graph completion provides the clearest positive result and, simultaneously, one of the clearest confounds. SimKGC [94] substantially outperforms TransE, ComplEx, and RotatE using a text-based scoring model. However, its inputs include textual descriptions of entities and relations that are unavailable to the structural baselines. The comparison therefore introduces an information advantage and does not establish that language recovers graph topology from triples alone. Similarly, Yao et al. [106] find that fine-tuned smaller models can outperform zero-shot frontier models on the same task, suggesting that task-specific adaptation rather than general linguistic reasoning accounts for much of the observed advantage.

Graph data therefore provides the clearest illustration of the distinction between describing and preserving structure. Models can describe permutation invariance fluently while violating it in their own predictions when the representation and computation are tested directly. In the terminology of Table 1, this is one of the few modalities in which structural replacement has been subjected to a direct test, and the test fails. SimKGC's apparent functional replacement is also weakened once its textual information advantage is controlled, while the strongest graph–LLM systems generally retain an explicit non-linguistic structural pathway at Levels 7–8.

4.3 Time-Series Forecasting

Time-series forecasting relies on inductive biases concerning temporal order, locality, and dependence across time. Specialized architectures encode these properties directly, whereas a pretrained language model has no inherent reason to respect them. This modality therefore provides some of the strongest controlled evidence against the claim that a pretrained language model automatically supplies the required temporal inductive bias.

Tan et al. [81] evaluate three widely used LLM-based forecasters (Time-LLM [43], GPT4TS/OneFitsAll [114], and CALF [52]) and systematically remove or replace their language-model components. Across thirteen datasets and two metrics, the resulting language-free variants outperform the original models in most comparisons while using roughly 1000× fewer parameters and up to three orders of magnitude less training time. Randomly reinitializing the pretrained weights also matches or exceeds the performance of the pretrained models. Most strikingly, shuffling the input sequence has little effect on predictions despite temporal order being central to the task. The result is decisive for the three architectures tested, although whether the finding generalizes to the broader family of LLM-adapted forecasters remains open.

Time-series foundation models reinforce this conclusion from the non-linguistic side. Chronos is described using a "language of time series" framing but operates on quantized numerical values

through a T5 architecture without natural-language input [4]. TiRex uses an xLSTM architecture without an LLM component and achieves state-of-the-art performance on GIFT-Eval [5]. Independent evaluations likewise report that LLM-based approaches can underperform numerical methods in epidemic forecasting [41], while train–test leakage and temporal correlation have been identified as sources of inflated performance in time-series foundation-model benchmarks [58].

A genuinely different direction has also emerged. Rather than adapting a pretrained language model to time series, Chronicle jointly pretrains language and time-series representations using shared attention blocks [69]. This approach is not contradicted by the ablation results of Tan et al. [81]; it implements a different hypothesis in which the two modalities are learned jointly rather than one being attached to a pretrained language model. The result remains preliminary, however, given the model's relatively small scale and the absence of independent replication.

Time series therefore provides the clearest separation between describing a temporal structure and computing over it. Representing a sequence in a language-model-compatible format is straightforward, but Tan et al. [81] directly test whether the language component contributes to the computation and find little evidence that it does for the architectures examined. The result is correspondingly negative for computational replacement: removing the language component yields equal or better accuracy at substantially lower cost. It is also negative for structural replacement, since a basic perturbation that should matter to a genuinely temporal computation (input shuffling) has little effect.

4.4 Vision and Spatial Reasoning

Vision requires distinguishing at least three regimes that are often conflated: systems that convert visual information into language before reasoning (Levels 2–3); vision-language models that process images through learned visual tokens connected to a language model (Level 7); and specialized vision architectures (Level 8). Most current vision-language systems occupy the second regime. Their performance therefore cannot be interpreted as evidence that language itself preserves or computes visual structure.

BLINK reports only 51.26% accuracy for the best evaluated vision-language model across fourteen computer-vision tasks reformulated as visual question answering, compared with 95.7% for humans [22]. A complementary study reports roughly 58% average accuracy across four frontier models on simple pixel-geometry tasks, again substantially below human performance [70]. A larger and more recent benchmark reaches a similar conclusion: across spatial-reasoning tasks involving depth, orientation, scale, and navigation, the best evaluated model achieves 54.93% accuracy in multiple-choice settings and 40.93% in open-ended settings, compared with 87.57% and 64.93% for humans, respectively [97]. Mechanistic evidence provides a possible explanation that predictions change relatively little when visual patches are randomly permuted, suggesting that the downstream language model does not fully exploit the spatial organization encoded by the visual representation [48].

Grid2Matrix identifies a more specific bottleneck. It describes a "Digital Agnosia" phenomenon in which the visual encoder retains substantially more information than is expressed through the model's language output [113]. Controlled experiments varying encoder architecture, positional encoding, and training objective across the LLaVA family report persistent spatial-reasoning deficits, suggesting that scale or encoder selection alone does not resolve the problem [3]. This is precisely the preservation-versus-computation distinction introduced in Section 3.3: a visual encoder may preserve spatial information that the subsequent language-mediated computation fails to exploit reliably.

Positive results nevertheless demonstrate that language can work effectively once spatial structure has already been discretized. Text-symbol grid representations reach 84–91% accuracy on a spatial-localization task for which the identical task posed in pixels achieves 60–73% [3]. SpatialEval similarly finds text-only reasoning outperforming vision-augmented models on synthetic maze and map tasks [93]. These approaches explicitly encode spatial structure into symbolic representations before language processing begins. They therefore provide evidence for language as a substrate for already-discretized structure, rather than for language as a general replacement for continuous visual computation.

Vision consequently sharpens the distinction between representation and computation. For raw visual inputs, structural replacement has been tested directly through perturbations such as patch permutation and has not been established. For already-discretized spatial structure, text-symbol representations come substantially closer to functional replacement because the representation has been designed to preserve the information required by the task before language processing begins. Describing and reasoning over spatial relations is therefore within language models' capabilities once the relevant structure is made symbolic; preserving and exploiting continuous geometric structure through language-mediated computation remains substantially less reliable.

4.5 Adjacent Modalities: Does the Pattern Generalize?

Five additional modalities provide a broader test of whether the preceding pattern is specific to tabular, graph, time-series, and vision data or recurs across other forms of structured information. The evidence is considerably thinner. So the conclusions below should be regarded as directional.

Chemistry. Chemistry provides the strongest positive result in the review. MoLFormer, a 1.1-billion-molecule transformer trained on SMILES without explicit graph structure, outperforms supervised and self-supervised graph neural networks across ten molecular-property benchmarks [73]. This constitutes a genuine case in which a sequence representation can functionally substitute for a graph-specific architecture at sufficient scale. The result, however, does not generalize uniformly within the modality. A 2026 study finds that smaller SMILES-only models exhibit "structural blindness" to rings and branching, with performance recovering only after graph-based information is reintroduced [11]. A 2025 survey similarly concludes that general-purpose LLMs remain insufficient for many scientific chemistry tasks, with recent systems increasingly orchestrating specialized tools rather than replacing them [33]. Chemistry therefore illustrates both the potential and the limitation of scale. MoLFormer's result establishes functional replacement at large scale, but structural replacement has not been directly tested [73].

Code. Code provides a particularly clean test of the describe-versus-compute distinction because code is already a symbolic language. CRUXEval finds frontier models achieving only 67% and 63% accuracy when predicting the input/output behavior of short Python functions, despite strong performance on code generation [32]. Fluent code generation therefore does not imply reliable simulation of program execution. No non-linguistic specialized baseline is reported for this task in the reviewed literature, so the regime boundary for code remains untested.

Point Clouds and 3D Data. Point-cloud results closely parallel the vision findings. Point-cloud-augmented multimodal models outperform text-only models on some spatial-reasoning tasks but

continue to fail on basic binary spatial relations [111]. PointLLM achieves strong 3D captioning by combining a specialized point-cloud encoder with a language model [104]. This is a Level 7 configuration: the structural representation remains in the non-linguistic encoder, while language primarily supplies the interaction interface.

Protein Structure. SaProt [77] and S-PLM [91] reintroduce structure-derived information into sequence-based protein models after sequence-only representations proved insufficient for capturing relevant structural relationships. The mechanism parallels the corrective pattern observed in chemistry and graph learning that once the structural information becomes important to the task, an explicit structural channel is restored. These systems therefore illustrate relocated specialization rather than its replacement.

The adjacent modalities support the same qualification observed in the four core domains. Language- or sequence-based representations can work well when the relevant structure is shallow, redundant with the representation, or recoverable through sufficiently large domain-specific pretraining. When the task depends critically on topology, execution semantics, or geometric relationships, the strongest systems continue to retain or restore an explicit structural channel.

4.6 Language as Orchestrator: The Genuine Growth Area

The strongest recent evidence for the practical value of language-mediated systems concerns orchestration and not replacement. At Level 6 of Section 3.1, LLMs generate, select, evaluate, and refine specialized computational components rather than performing the underlying structural computation themselves. LLM-FE generates and evaluates feature-transformation programs within an evolutionary search and consistently improves over previous automated feature-engineering methods [1]. MLE-STAR retrieves baseline models and iteratively refines them through targeted, ablation-guided experimentation, earning medals in 64% of evaluated Kaggle competitions [61]. At industrial scale, a planner-guided multi-agent system reduced feature-engineering time from three weeks to one day [84].

These results establish a meaningful and increasingly practical role for language models that searching over specialized methods, composing tools, evaluating alternatives, and refining machine-learning pipelines. The underlying structural computation, however, remains with the specialized models being selected or invoked. The evidence therefore supports orchestration in the specific sense of Table 1, not the other seven senses of replacement defined there. An important open question follows: whether the component-ablation logic that isolated the contribution of language models in time-series forecasting [81] can similarly determine how much of the observed orchestration benefit is attributable to the language model itself rather than to the search and tool-use process it enables.

4.7 Why These Comparisons Are (Not) Scientifically Valid

Claims that one method "outperforms" another are meaningful only when the conditions of comparison are explicit. Three factors recur across the modalities reviewed above and determine whether such comparisons are informative. First is the *data regime*. Language-mediated methods can exhibit advantages in extreme few-shot settings but lose those advantages as labeled data accumulate [39]. Second is the *compute budget*. The ablation of Tan et al. [81] shows that removing the language-model component can reduce parameter count by roughly 1000× without reducing predictive performance,

making accuracy-only comparisons potentially misleading. Third is *information availability*. SimKGC and MoLFormer both benefit from information (textual descriptions in the former and large domain-specific pretraining corpora in the latter) that may not be available to their specialized baselines. In such cases, a comparison confounds representational choice with information access. Comparisons that fail to control or report data regime, compute budget, and information availability should therefore be interpreted cautiously. A reported performance difference may reflect unequal data, computation, or information rather than a genuine advantage of one representational strategy over another.

4.8 Cross-Modal Synthesis

Considered individually, the nine modalities above appear to constitute nine distinct application areas. Considered comparatively, however, they converge on a much smaller set of recurring mechanisms. These mechanisms provide the strongest evidence for the review's central argument.

The clearest convergence is a three-part pattern that recurs wherever the evidence is sufficiently developed to permit controlled comparison. First, a strong non-linguistic specialized baseline can outperform language-mediated alternatives when evaluated under matched conditions: TabPFN in tabular learning, TiRex in time series, GNNs in graph learning, and specialized vision architectures in visual reasoning. Second, language-mediated systems can obtain narrow advantages under particular conditions, such as extreme few-shot settings in tabular learning or access to additional textual information in knowledge-graph completion. Third, when direct controls are available, either the apparent advantage disappears under closer scrutiny, as in Tabula-8B, or the language component turns out not to be responsible for the underlying computation, as in the ablation of Tan et al. [81].

A second convergence is even more striking that at least three modalities independently adopted the same corrective strategy. CARTE introduces a graph-attention scaffold around language-derived representations; SaProt and S-PLM reintroduce structure-derived tokens; and chemistry systems reattach graph-based tools after sequence-only representations prove insufficient. These are not identical architectures, but they implement the same methodological response that when a language- or sequence-based representation fails to capture a task-relevant structural property, an explicit non-linguistic structural channel is restored. Section 7.1 examines whether this recurring response is better explained as a mechanistic property of the representational regime than as a collection of domain-specific engineering decisions.

The modalities diverge, however, in a way that is consistent with the taxonomy. When task-relevant structure can be discretized into a symbolic representation before language processes it (as in text-symbol spatial grids or SMILES-based molecular representations), language models can perform well because the representation preserves much of the information required by the task. When the relevant structure is continuous, relational, or defined by an invariance that serialization does not automatically preserve (as in raw visual geometry, graph topology, or temporal ordering), the same strategy is substantially less reliable, and strong systems tend to retain or restore a non-linguistic structural pathway.

This pattern suggests that *discretizability* may be an important moderator of when language-mediated representations can substitute for specialized architectures. The critical question is not simply whether an object can be written as a sequence, but whether the task-relevant structural relations survive that transformation in a form that the subsequent computation can exploit. Chemistry provides an important qualification. MoLFormer's success at very large scale, together with the structural

failures observed in smaller SMILES-only models, suggests that sufficiently large domain-specific pretraining can partially compensate for representational limitations. Whether such compensation remains possible under matched compute, data, and information budgets is not yet established.

Finally, no modality in this review provides unambiguous evidence of *structural replacement* (that is, of a language-mediated computation implementing the same structural constraint that a specialized architecture would encode explicitly) that survives a direct test. The strength of this negative conclusion differs across modalities. Time series, graphs, and raw-pixel vision subject the claim to direct tests: component ablation, controlled serialization and permutation analysis, and patch-permutation experiments, respectively. In each case, structural replacement fails the test. By contrast, tabular data, chemistry, code, and many of the benchmarks summarized in Table 3 have not undergone an equivalent direct structural test. Their apparent successes may instead be weakened by information advantages, narrow data regimes, or benchmark contamination, but these findings do not constitute direct tests of structural replacement.

The evidence for the review's central claim is consequently strongest where the relevant structural property has been tested directly and weaker where the literature has relied primarily on benchmark performance. Section 7 returns to this distinction and asks the harder question left open by the modality-level evidence that whether the observed convergence reflects limitations of current language models or a more fundamental limitation of language-mediated computation itself.

5 Datasets and Benchmarks

A dataset, a task, and a benchmark are not the same thing, and the difference matters for what follows. A *dataset* is a collection of examples; a *task* is the prediction problem defined over it; a *benchmark* is a standardized combination of dataset, task, split, metric, and baseline that licenses a specific comparison. A benchmark score is therefore always a claim about performance under one particular combination of these choices, not a direct measurement of representation, computation, or replacement in the sense of Section 3. This distinction is not pedantry because Section 3.2 shows that a single accuracy number cannot, on its own, distinguish functional replacement from statistical, computational, or structural replacement, and a benchmark's design determines which of these a given score can possibly speak to before any model is ever run on it.

Table 3 summarizes the benchmarks this review relies on most, chosen to illustrate what each protocol can and cannot distinguish. Read against the regime ladder and the four-way distinction, a pattern emerges that this section treats as a finding in its own right that almost every entry directly evaluates predictive performance, and almost none evaluates anything else. TabArena, TALENT, and GIFT-Eval measure in-domain accuracy; BLINK and MMVP measure accuracy on reformatted or targeted subsets; WN18RR and FB15k-237 measure link-prediction accuracy under a comparison that, as discussed below, is not actually matched. Only two exceptions in the entire reviewed corpus depart from pure task-level evaluation toward representation- or replacement-level evidence: Tan et al.'s component ablation, which is a one-off experimental protocol rather than a standing benchmark, and the direct invariance-violation metric used by Yoon et al. [108] and Egressy and Stühmer [19] and discussed further in Section 6.1. Everything else in this review's evidence base infers describe, preserve, compute, and learn from a single number that was designed to measure none of them individually.

Table 3: Benchmarks referenced across the review, selected to illustrate what each protocol can and cannot distinguish (Section 5).

Benchmark	Modality	Size / scope	Known limitation
TabArena / TALENT	Tabular	200+ real datasets	Skews toward small–medium IID tables where tabular foundation models already win [67]
BeyondArena	Tabular	142 datasets, IID + temporal + grouped shift	Purpose-built to expose the IID-only bias of prior benchmarks
GIFT-Eval	Time series	23 datasets, 144K series, 7 domains	Includes a dedicated non-leaking pretraining split, itself a response to contamination in earlier TSFM benchmarks
NLGraph / GraphQA	Graph	Synthetic reasoning tasks (connectivity, shortest path, etc.)	No GNN baseline in most reported comparisons; complexity confound not isolated from encoding choice
GraphInfer-Bench	Graph	42,000 samples, 6 real graphs	Recent (2026); limited independent replication
BLINK	Vision	14 classic CV tasks reframed as VQA	Multiple-choice format may itself bottleneck scores independent of visual competence
MMVP	Vision	CLIP-blind image pairs	Diagnoses CLIP-specific pooling/resolution issues; generalization to non-CLIP encoders untested
MLE-Bench (Lite)	Tabular / AutoML agents	Kaggle competition subset	"Lite" subset is easier than the full benchmark; medal rates may not transfer
WN18RR / FB15k-237	Knowledge graphs	Standard KGC benchmarks	Rich textual entity descriptions available to text-based methods but not to pure embedding baselines; confounds the SimKGC comparison (Section 4.2)
CRUXEval	Code	800 short Python functions	Execution-reasoning gap may not transfer to longer, real-world programs
MoleculeNet [101]	Chemistry	700K+ compounds, 17 datasets across 4 categories	Predates language-mediated methods; Ross et al.'s ten-benchmark subset is not independently re-audited for contamination here
ProteinGym [63]	Protein	250+ deep mutational scanning assays	Zero-shot fitness prediction only for most assays; Su et al. [77]'s ten-task suite extends beyond this benchmark alone
ModelNet40 [102]	Point clouds	12,311 synthetic CAD models, 40 categories	Synthetic and clean; Zhang et al. [111]'s binary-spatial-relation failures are documented on separate, more realistic benchmarks

Benchmark bias. Where benchmark design does shape the comparison, the evidence located in this review points in one direction more than the other. *Regime selection* favors language-mediated methods where it is least informative to do so. TabArena and TALENT are dominated by small-to-medium IID tabular datasets, exactly the regime in which tabular foundation models perform most strongly, and Purucker et al. [67]'s broader, purpose-built alternative shows that advantage weakening substantially under temporal and grouped distribution shift. *Contamination* inflates reported performance in the same direction that independent audits found train–test overlap in widely used tabular benchmarks [28, 75] and leakage in time-series foundation-model benchmarks [58], in each case a bias that specifically favors whichever pretrained model had prior exposure to the test distribution. *Information asymmetry* does the same in knowledge-graph completion. WN18RR and FB15k-237 supply textual entity and relation descriptions to text-based methods that structural embedding baselines such as TransE, ComplEx, and RotatE never receive, so SimKGC's reported advantage reflects an unequal comparison, not solely a representational one.

The reverse direction is real but operates through a different mechanism. A dedicated search of the wider evaluation-methodology literature, beyond this review's own 159-paper corpus, finds systematic output-format bias documented across state-of-the-art LLMs on exact-match and structured-output tasks [53]. A scoring artifact that penalizes a linguistically fluent but format-variable answer regardless of whether the underlying computation was correct, and that a specialized model producing output in the required format by construction never incurs. This is not the same claim as "benchmarks favor specialized architectures because the tasks are numerically or structurally demanding"; it is a claim about the scoring mechanism itself disadvantaging language-mediated output independent of the reasoning it reflects, and it is the clearest evidence this review locates running against language-mediated methods. The two biases are not symmetric in the sense of canceling each other out: regime selection, contamination, and information asymmetry each shape which *tasks* get chosen or how *information* is distributed, while output-format bias shapes how a correct answer gets *scored*, a narrower but real effect operating alongside, not instead of, the biases documented above.

A separate class of benchmark simply cannot support a replacement claim in either direction, which is a different problem from bias. NLGraph and GraphQA report no GNN baseline in most comparisons, so no result on either benchmark can establish whether language-mediated reasoning is competitive with, better than, or worse than the specialized computation it would need to replace. MLE-Bench's Lite subset is easier than the full benchmark it is drawn from, so medal-rate results such as Nam et al.'s 64% should not be read as transferring automatically to the harder setting. BLINK's multiple-choice format may bottleneck scores independently of the visual competence it is meant to measure. Each of these is a documented limitation in Table 3.

What current benchmarks do not test. Two gaps follow directly from the pattern above and connect this section forward to Section 8. First, information preservation and structural computation, the second and third stages of Section 3.3, are evaluated almost nowhere: Section 6.1 shows only two papers in this review's core matrix use a metric designed to detect a preserved-but-uncomputed structural property rather than an accuracy proxy that cannot distinguish the two. Second, replacement in the structural or computational sense of Table 1 is tested almost nowhere: Tan et al.'s ablation is the only genuine component-level test located in this review, and Section 4.6 notes the same gap for Level-6 orchestration systems, whose reported end-task success cannot currently be decomposed into the language model's contribution versus the specialized tool's. Coverage is also uneven by modality: chemistry, protein structure, and point clouds now each have exactly one benchmark entry in Table 3 (MoleculeNet, ProteinGym, and ModelNet40 respectively)

against several apiece for the four core modalities, which mirrors rather than resolves the thinner evidence base Section 4.5 already discloses for those three; a single benchmark supports far less scrutiny of contamination, baseline fairness, or regime selection than the multiple entries available for tabular, graph, time-series, and vision data. Code remains the one modality with no benchmark entry at all, consistent with Section 4.5's finding that no non-linguistic baseline is reported for it in the reviewed literature. No benchmark in this review's corpus, in any modality, was purpose-built to test serialization sensitivity or invariance directly. Section 8.1 specifies the controlled experiment this absence motivates. A design that holds information content and sample budget fixed while measuring the direct invariance-violation metric Section 6.1 shows.

6 Evaluation Methodology and Theoretical Foundations

6.1 What current metrics can and cannot distinguish

A benchmark score is an observation about model behavior, not a direct measurement of the mechanism that produced it. Table 4 summarizes the evaluation metrics used across the literature reviewed in Section 4, and reading it against Sections 3.1 and 3.3 exposes a hierarchy the metrics themselves do not make explicit. Accuracy, error, and task-specific scores measure *output performance* means whether a prediction matches a label. Zero- and few-shot deltas measure a form of *generalization behavior*: whether performance survives reduced adaptation. Neither level says anything about *representation fidelity* (whether the information a task requires survived the transformation into a language-mediated form) or about *computational behavior* (whether the model's forward pass respects the structural property that fidelity would make available). The strongest level, *mechanistic replacement*, asks whether a specialized component can actually be removed without loss under matched conditions, and almost nothing in Table 4 operates at this level. Two systems can therefore be *predictively equivalent* (similar accuracy) while differing entirely in whether they are representationally equivalent (retain the same task-relevant information), computationally equivalent (implement the same function), or architecturally substitutable in the stricter senses Table 1 distinguishes. Collapsing these into a single accuracy comparison is the same conflation Section 3.2 identifies at the conceptual level, located here in the measurement instrument itself.

Direct tests of structural behavior narrow this gap only rarely. Measuring the maximum change in a model's output under a structure-preserving transformation, $\max_\pi |f(x) - f(\pi(x))|$, is a representation- and computation-level test. It asks whether the model's behavior is invariant. Only two studies in the reviewed literature use such a measure [19, 108]. The strongest available evidence at the mechanistic-replacement level is not a metric at all but a controlled ablation (removing or reinitializing a specific component while holding data, task, and evaluation fixed, as in Tan et al. [81]) because only a counterfactual comparison of this kind can establish that a component was unnecessary. Aggregate benchmark scores compound the problem that averaging across datasets of different structural difficulty, as TabArena and TALENT do for tabular data (Section 5), can report a healthy mean while concealing failure concentrated in the harder, less-represented cases [67]. The literature has therefore extensively measured whether models are accurate, and rarely measured whether they compute the structure that would explain that accuracy. Section 8 returns to this gap and specifies the controlled design needed to close it.

Table 4: Evaluation metrics used across the reviewed literature. Accuracy-based metrics dominate but cannot, by construction, distinguish "the model computed the right structure" from "the model reached the right answer by another route"; the direct invariance-violation metric is the only one in this table designed specifically to make that distinction (Section 3.3).

Metric	What it measures	Blind spot
Held-out accuracy / F1	Whether predictions match labels on unseen examples	Cannot distinguish memorization or format familiarity from genuine structural learning (Section 4.1)
Zero-shot / few-shot delta	Advantage over a baseline with less adaptation	Confounded by pretraining-corpus contamination unless a dated, held-out split is used
Ablation delta (component removed)	Causal contribution of a specific model component	Rarely reported; the single instance that matters most in this review (Tan et al. 81) is an outlier in the literature, not the norm
Fitted scaling-law exponent	How loss decreases with model/data scale	Says nothing about whether the *gap to a matched-bias architecture* shrinks: exponents are absolute, not relative (Section 6.3)
Direct invariance violation, $\max_\pi \lvert f(x) - f(\pi(x)) \rvert$	Whether output is literally invariant under a relevant transformation	The only metric here that operationalizes the description-vs-computation distinction directly; used by only two papers in this review (RoToR, Set-LLM)
Contamination-adjusted accuracy	Accuracy after removing detected train/test overlap	Detection methods are themselves imperfect and likely undercount contamination [27]

6.2 Why the empirical pattern has a theoretical floor

"Theoretical floor" names a specific, bounded claim here, not a claim of impossibility in general. For certain structural properties, formal results establish a computational or sample-complexity cost that does not vanish merely by increasing model scale within the same representational regime, unless the model also acquires the relevant structure through its architecture, its representation, or its training data. This is weaker than saying language-mediated computation cannot in principle solve these tasks, and stronger than an empirical trend that might simply reflect insufficient scale to date; the distinction matters because the two readings license different predictions about what more compute would do, and this subsection is careful throughout about which of the two a given result actually supports. The starting point is the No Free Lunch theorem: without assumptions about the target function, no learner has a universal advantage, so reliable generalization from finite data requires an inductive bias matched to the task's structure [99]. This result is formal but general; it does not say which architectures pay which costs for which structures, which is what the results

below establish case by case.

A clarification is necessary before comparing biases at all. A language model has inductive biases of its own, arising from its attention architecture, its positional encoding, its tokenization, its pretraining objective, and the distribution of its pretraining data. The relevant question is therefore not specialized bias versus no bias, but explicit, domain-matched bias built into an architecture by construction versus general-purpose bias whose match to a specific structural property is incidental and must be checked, not assumed. Every result below should be read as characterizing whether the bias transformers already have happens to match a given structural property, not as showing that transformers compute unconstrained by any bias.

For some function classes, formal results say the match is close. Kim et al. [44] prove that transformers can be minimax-optimal for function classes such as smooth nonparametric regression and Bayesian posterior mixtures, provided the pretraining distribution is matched to the target tasks. Wakayama and Suzuki [90] prove that the posterior-variance component of in-context risk decays exponentially with the number of demonstrations under stated assumptions. Both are formal results for specific function classes, not empirical regularities and not claims about in-context learning in general; together they provide a theoretical account for the positive results this review documents in low-data settings such as tabular prediction (Section 4.1), where a broad pretraining distribution can supply a useful statistical prior without a task-specific architecture.

This positive picture is itself contested, and a dedicated search for counterevidence located a direct challenge. A body of theoretical work argues that in-context regression succeeds because transformers implement a known algorithm (ordinary least squares or gradient descent) during the forward pass. Hill et al. [36] report empirical evidence against this specific mechanistic story: transformers trained for in-context least-squares regression fail to generalize once the prompt distribution shifts, a pattern inconsistent with genuinely implementing OLS, and their behavior instead correlates with spectral signatures of the training distribution, consistent with a form of memorization rather than algorithm execution. This work does not contradict Kim et al.'s minimax-optimality result directly (the two examine different function classes and different notions of what "implementing an algorithm" requires), but it is a direct, located counterexample to the stronger informal claim, made elsewhere in this literature, that strong in-context regression performance is evidence the model has learned a generalizable procedure rather than a distribution-specific shortcut. The theoretical literature on why in-context learning succeeds is not settled even for the function classes where it is reported to succeed.

The situation is different for stronger compositional or relational structure. Kozachinskiy et al. [46] formally prove that one-layer softmax attention cannot solve three-way matching, function composition, or binary-relation composition, regardless of width or numerical precision, a limitation independent of scale by construction, since the proof holds for arbitrarily large width and precision. The same paper constructs an alternative mechanism, Strassen attention, and proves it solves all three tasks, which is the detail that makes this a claim about architecture rather than about capacity: the barrier is removable, but only by changing what the attention mechanism computes, not by enlarging the existing one. This is the clearest formal theorem in this review's theoretical corpus and the closest direct theoretical counterpart to the empirical failures documented in graph-structured data (Section 4.2).

Permutation invariance illustrates the same architecture-versus-scale distinction with a matching efficiency result. Tabaghi and Wang [79] prove that a permutation-invariant architecture can universally represent identifiable multiset functions using a latent dimension of $2DN$ (vector dimension

D, multiset size N), a formal, constructive bound, and a substantial improvement over the $O(N^D)$ bound implied by earlier sum-decomposable models. A standard transformer carries no equivalent guarantee and must learn permutation-invariant behavior from data if it acquires it at all. Chiu et al. [14] report the empirical consequence that explicitly encoding permutation invariance matches a transformer's in-context-learning performance on a permutation-invariant task using roughly an order of magnitude fewer parameters. A formal bound and an empirical efficiency result, read together, do not show that transformers cannot learn structured functions; they show that an architecture encoding the relevant structure by construction reaches the same performance at a measurably lower parameter cost.

Two further results including, Webson and Pavlick [98] show experimentally that language models can learn effectively from prompts whose stated content is actively misleading, indicating that a prompt's semantic content is not reliably coupled to the computation the model performs. Yoon et al. [108] report a directly relevant engineering finding that prompting and formatting alone were insufficient, in their experiments, to reliably induce order-invariant behavior, and achieving it required modifying the positional mechanism. Neither is a theorem; both are consistent with, and offer empirical support for, the distinction in Section 3.3 between describing a structural concept and computationally implementing it; Yoon et al.'s result specifically locates where architecturally encoded bias would need to be restored (the positional mechanism) once description and prompting are shown insufficient on their own.

The pretraining distribution is a further candidate locus for structural bias, and the evidence here is again a mix of formal and empirical results applying to a narrower setting than in-context learning in general. Goddard et al. [26] empirically characterize, through a phase diagram built from simulated regression tasks, a phase transition in task diversity: below a critical level, in-context learning remains tied to the pretraining task distribution; above it, the learned procedure generalizes to a broader task space. Azizian and Hasan [6] prove that the choice of pretraining-task distribution carries a formal trade-off between robustness to distribution shift and sample efficiency, so widening the pretraining distribution is not a free way to acquire more structure. It substitutes one cost, task-specific data, for another, a less sample-efficient prior. Universal-approximation results establish that transformer-based in-context learning is expressive enough, in principle, to represent a broad class of functions [49], but expressivity is a different property from the sample and compute efficiency the results above address, and this specific gap (between what transformers can represent and what they can efficiently learn) is exactly where the current theoretical literature remains incomplete. No result surveyed here characterizes the sample complexity of learning a permutation-invariant or graph-structured function through pretraining-distribution diversity alone, which is the missing connective theory Section 8 treats as this review's central open question.

6.3 Scaling is real, sub-linear, and does not by itself close the gap

This subsection's title makes three separable claims, and each should be checked against what the referenced studies measure. Whether scaling improves performance at all; that any such improvement is sub-linear, with respect to a specified quantity; and that scaling closes the gap to a matched-bias specialized architecture rather than merely improving in isolation. The literature supports the first claim directly, the second precisely once the scaled quantity is specified, and only a substantially weaker version of the third.

Ma et al. [54] fit a power-law exponent of approximately 0.4 for tabular in-context learning with respect to both model parameter count and pretraining data size, considered separately, an exponent

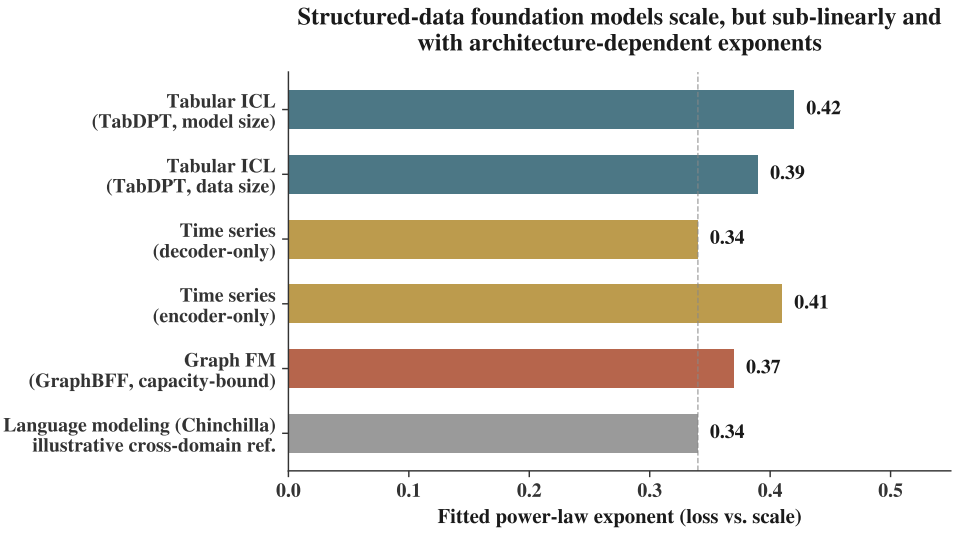

Figure 4: Reported scaling exponents for structured-data foundation models, with the Chinchilla language-modeling exponent shown as an illustrative reference [83]. The reported exponents are positive, indicating that scale improves performance, but remain sub-linear. Existing studies do not directly measure how the performance gap relative to a matched-bias architecture changes with scale and structural complexity.

below 1 in the sense of $L(N) \propto N^{-\alpha}$, so loss falls more slowly than scale grows, which is the only sense in which this review uses "sub-linear." For time series, Yao et al. [107] report that encoder-only architectures scale more favorably than decoder-only architectures, and that changes improving in-distribution performance can reduce out-of-distribution scalability, a finding about which architecture scales best, not only whether scaling helps. In graph learning, Liu et al. [51] find that model depth, rather than parameter count, is the dominant determinant of scaling behavior, so the relevant scaled quantity is not even the same across studies. Tay et al. [83] show more broadly that the architecture with the best scaling behavior changes as scale increases, so an exponent measured at one scale range need not extrapolate to another. None of these four studies measures the same quantity, over the same scale range, against the same baseline, which is why Figure 4 presents them side by side rather than pooled into a single comparison; pooling them would not be a valid inference from what any one of them reports.

None of the four measures captures the quantity that this review's central question actually requires. The relevant quantity is not $L(N)$, the loss of the language-mediated model as scale N increases, but $\Delta(N) = L_{\text{general}}(N) - L_{\text{specialized}}(N)$, the performance gap to a matched-bias architecture as a function of scale. A positive, shrinking $L(N)$ is compatible with $\Delta(N)$ shrinking toward zero, staying constant, or even growing, depending on how the specialized baseline itself scales over the same range; Tay et al.'s finding that the best-scaling architecture changes with size is a direct reason to expect $\Delta(N)$ need not behave monotonically even where each individual $L(N)$ does.

The evidence located in this review supports a conservative reading that scaling reduces loss within a representational regime, at a rate that is real but sub-linear in the specific senses reported above, and no study identified here measures whether that reduction eliminates $\Delta(N)$ asymptotically, because no study measures $\Delta(N)$ directly. This claim rests on a dedicated search of the wider scaling-law literature conducted specifically for this review, looking for any controlled comparison that scales a language-mediated and a matched-bias specialized architecture together and reports the gap between them as a function of scale. No genuine $\Delta(N)$-measuring study was located in either direction

(neither one showing scaling closing a matched-bias gap for structured data specifically, nor one showing scaling failing to close one under a controlled comparison.

7 Comparative Synthesis

Section 4 establishes the patterns that recur across nine modalities; this section asks what those recurrences imply, answers the motivating questions posed in Section 1, and states explicitly what the evidence does and does not establish.

7.1 Why the cross-modality pattern is mechanistic?

Section 4.8 identifies the same corrective move in at least three research communities with no shared authorship, benchmarks, or citation practice. The recurrence itself is initially only phenomenological. Three communities arriving at similar remedies could reflect a shared underlying constraint, unrelated domain-specific failures that happen to admit similar repairs, diffusion of ideas across neighboring fields, or simply the cases this review was able to document most clearly.

The mechanistic interpretation is nevertheless the most economical explanation of the evidence. First, the correction is not attributable to scale, pretraining data, or compute alone. In each case, the decisive intervention is architectural or representational (an attention scaffold, a vocabulary extension, or a reattached structural tool) rather than simply a larger model or longer training. Second, the effect is not explained by prompting or superficial engineering. None of the three studies characterizes its corrective mechanism as a prompting intervention, while Yoon et al.'s attempt to induce order-invariant behavior through prompting and formatting alone, discussed in Section 6.2, failed until the positional mechanism itself was modified. Third, the convergence is not an artifact of trivially weak baselines. MoLFormer's SMILES-only representation already outperformed graph neural networks at 1.1 billion molecules of pretraining scale [73]; the subsequent structural correction therefore addresses a specific failure (structural blindness to rings and branching) within an otherwise strong sequence model. Finally, the fact that the successful systems are hybrid rather than purely linguistic does not weaken the argument. Hybridization is precisely the recurring response that once a sequence or serialization representation is shown to omit information required by the task, the missing structural channel is restored.

The mechanistic interpretation is further supported by Section 6.2, where two relevant constraints are established independently of the application domains. Kozachinskiy et al. [46]'s VC-dimension ceiling concerns one-layer softmax attention as a function class. The limitation applies to tasks requiring the specific compositional operations considered in the proof, regardless of whether the underlying inputs represent molecules, proteins, tables, or other objects. Likewise, Egressy and Stühmer [19]'s non-permutation-invariance result follows from positional encoding and causal ordering, properties that arise whenever a decoder-only transformer is applied to set- or graph-structured inputs. These results provide a domain-agnostic mechanism that predicts why structurally different applications can encounter the same class of failure. The independently observed convergence of corrective strategies is therefore consistent with a common architectural constraint and is more explained by it.

The argument also does not exclude methodological diffusion between adjacent fields. CARTE [45] and SaProt [77], for example, necessarily draw on ideas from broader graph-neural-network and structural-learning literatures. The narrower claim is that such diffusion is not sufficient to explain why the same type of correction is effective. A domain-agnostic architectural mechanism

independently provides a reason for these communities to require an explicit structural channel. Thus, the convergence documented in Section 4.8 is best interpreted as evidence for a shared mechanism affecting at least two of the four failure modes in Table 6, establishing a common cause for all four. Section 7.4 returns to this distinction.

A dedicated search for a counterexample to this interpretation (a pure, architecturally unmodified language model solving a permutation-sensitive or compositional task at scale without an external structural channel or modification to its own attention or positional mechanism) did not identify one. Instead, it revealed a refinement already contained in the reviewed evidence. Egressy and Stühmer [19]'s Set-LLM, introduced above as evidence that standard decoder-only transformers are not permutation-invariant, also provides a direct architectural repair that it replaces the model's positional encoding and causal attention mask with set-specific alternatives, proves invariance from the resulting construction, and validates it empirically. This is an internal modification of the language model rather than the addition of an external non-linguistic encoder of the kind used by CARTE, SaProt, and the chemistry approach. The mechanistic claim is therefore more precise than the statement that language-mediated systems must always attach a separate structural component. The evidence instead identifies at least two ways of restoring the missing inductive bias: modify the attention mechanism itself, as in Kozachinskiy et al.'s Strassen attention and Egressy and Stühmer's Set-LLM, or reintroduce a separate structural pathway, as in CARTE, SaProt, and the chemistry correction. What the reviewed evidence does not provide is a case in which scale, additional data, or prompting alone resolves the underlying structural limitation while the architecture and representation remain otherwise unchanged.

7.2 The claim map

Table 5 consolidates the evidence from Sections 4 and 6 into ten claims that recur, in different formulations, throughout the reviewed literature. Each claim is paired with its strongest supporting and counter-evidence. **The strongest claim is that specialized inductive biases remain necessary**. Its evidence is consistently supportive across the modalities reviewed, although the demonstrations differ in form: TabPFN achieves a 230× inference speedup over tuned boosting ensembles without using language [37]; PatchTST reaches state-of-the-art long-horizon forecasting using a language-free patch-based architecture [62]; Set Transformer explicitly encodes permutation-invariant attention for set-structured inputs [47]; and GraphBFF demonstrates predictable scaling in a billion-parameter, non-linguistic graph architecture [9]. This is the claim most closely aligned with the theoretical distinction in Section 6.2 that specialized structure is not merely an implementation preference but can provide measurable computational or statistical advantages.

A second tier consists of the claims that language provides a genuine interface and orchestration capability, and that predictive competitiveness is real in specific regimes. These claims are well supported but conditional. The evidence in Section 4 establishes concrete settings in which language-mediated systems are competitive, including extreme few-shot prediction, symbolic or discretized representations, and orchestration of specialized tools. It does not establish that these advantages extend across data regimes, structural complexity, or matched computational budgets.

A third tier contains claims about generalization to unseen synthetic structural tasks and statistical efficiency. Here the evidence is genuinely contested. Positive results exist, but so do controlled failures and evidence of information asymmetry or contamination. Once those factors are accounted for (Section 4.7), the evidence in this review weighs against broad interpretations of these results, but it does not justify treating every such claim as definitively disproven.

Table 5: Claim map: the ten claims in the literature, with the strongest evidence and counter evidence located on each side (Section 7.2). "Status" is this review's synthesis judgment.

Claim	Supporting evidence	Counter-evidence	Status
LLMs learn tabular functions from text	TabLLM, few-shot	Grinsztajn et al. [29]; Gorla and Puduppully [28]	Narrow yes
LLMs learn graph functions from text	GraphText (uncorroborated)	Egressy and Stühmer [19]; Thushalika et al. [86]	Mostly no
Language preserves graph structure	—	61.8-point encoding swing [21]	No
LLMs learn temporal functions from text	LLMTime, zero-shot	Tan et al. [81]	Mostly no
Language can express / install inductive bias	Decomposition-prompting narrows compositionality gap	Yoon et al. [108]; Webson and Pavlick [98]	Largely no
LLMs replace specialized architectures generally	—	Entire method-comparison table (Table 2)	No
Specialized inductive biases remain necessary	TabPFN family [37], PatchTST [62], DLinear [110], Set Transformer [47]	—	Yes
LLMs generalize to unseen synthetic structural tasks	Garg et al. [24] (simple function classes)	Dziri et al. [18]; NP-complete graph tasks	Bounded
LLMs are statistically efficient learners	TabLLM extreme few-shot	Garnelo and Czarnecki [25]; Grinsztajn et al. [29]	No
LLMs are computationally efficient learners	—	Tan et al. [81]: 1000× compute for no gain	No

The weakest claims are those asserting general replacement of specialized architectures. No located result supporting such a broad claim survives direct scrutiny across the relevant dimensions of representation, computation, information availability, and structural invariance. These claims are therefore better characterized as refuted within this review's evidence base.

The minimum defensible thesis that explains this hierarchy is narrower than either extreme in the field's public discourse. Language-mediated systems can reliably achieve behavioral competitiveness in identifiable regimes (including extreme few-shot prediction, discretizable spatial layouts, textually annotated knowledge graphs, and large-scale single-modality pretraining), but that competitiveness does not generalize into representational, computational, or architectural replacement in the stricter senses distinguished in Table 1. Notably, the direction of recent research already reflects this distinction more clearly that the strongest systems increasingly use language as an interface to, or orchestrator of, specialized structural computation rather than treating language itself as a universal substitute for that computation (Section 2.4).

7.3 Answers to the central questions

The confidence labels below are qualitative judgments about this review's evidence base. *High* indicates that multiple independent studies, or a formal proof, directly address the specific claim. *Moderate* indicates that the evidence directly supports a closely related question but does not fully establish the exact claim. *Low* indicates that the conclusion depends primarily on a single unreplicated result or on the absence of counterevidence. This convention is consistent with the qualitative interpretation used in this work (Figures 2 and 3).

Is language a general representation for structured data, or primarily an interface to systems that perform the underlying computation? Primarily an interface, with high confidence within the structural regimes directly tested. Every domain in Section 4 can be described, queried, or orchestrated through language (Section 4.6), but interface universality is not equivalent to representational sufficiency. The failures documented in Section 4 concentrate precisely at this boundary that language-mediated representations do not reliably preserve permutation invariance (Section 4.2), temporal structure (Section 4.3), or continuous spatial geometry (Section 4.4) where these properties have been directly tested.

Can an inductive bias that a specialized architecture enforces by construction instead be recovered by stating or demonstrating it in a prompt? Rarely, and not at the level of exact computational invariance directly measured in the reviewed cases. Confidence is high for the cases actually tested, without extending the conclusion beyond them. Yoon et al.'s attempt to induce order-invariant behavior through prompting and formatting alone failed on real listwise tasks and required modification of the positional-encoding mechanism (Section 6.2). Kozachinskiy et al. [46] provide a theoretical reason for expecting such failures in broader settings that their structural VC-dimension ceiling for one-layer attention cannot be removed by additional instructions or demonstrations because it follows from the computational form of the attention mechanism itself.

If a language-based model matches a specialized model's accuracy, what exactly has been replaced? At most functional replacement in the narrow sense of Table 1, with moderate-to-high confidence. This conclusion is partly definitional but has important empirical consequences. A predictive match establishes equivalence only under the data regime, compute budget, and information conditions of the comparison (Section 4.7). Structural or computational replacement requires additional evidence, such as a direct invariance test or a component-level ablation. Where this review locates such tests (including Tan et al.'s ablation and Egressy and Stühmer's architectural analysis), the evidence does not establish structural replacement.

Can existing evidence distinguish genuine structural generalization from benchmark familiarity, encoding artifacts, or external computation? Sometimes, with confidence that is high where a direct test exists and low where it does not. The distinction is not uniformly available across the corpus. Controlled ablation, independent contamination audits, and information-matched comparisons can separate these explanations when they are performed: Tan et al.'s ablation isolates the contribution of the language-model component; Gorla and Puduppully and Silvestri et al. provide evidence distinguishing genuine generalization from benchmark familiarity; and Fatemi et al. directly tests the sensitivity of performance to the textual encoding of an otherwise fixed graph. But Section 5 shows that such protocols remain exceptional. For most reported results, the correct conclusion is therefore not that structural generalization has been demonstrated or ruled out, but that the evaluation was never designed to distinguish it from these alternatives.

7.4 Limitations and failure modes

The conclusions of this review are bounded by two distinct classes of limitation. The first concerns the evidence base itself. Coverage is uneven across modalities: the four core modalities contain approximately fifteen to twenty-five papers each, whereas the five adjacent modalities contain only five to seven each (Section 4.5). More importantly, Section 5 shows that most benchmarks in this literature were not designed to test representation preservation or replacement directly. Consequently, an absence of positive evidence has two possible interpretations: in some cases, a claim has been directly tested and failed; in others, the relevant test was simply never performed. Section 7.3 makes

this distinction explicit. This is a limitation of what the field has measured.

The second limitation concerns the failure modes of language-mediated learning themselves. Table 6 organizes these failures by mechanism rather than symptom because a mechanism-first organization allows a more informative question: can the failure plausibly be resolved by scale, by changing the representation, or only by restoring specialized computation? The evidence does not support the same answer for all four mechanisms.

For two mechanisms, the claim that scale alone does not resolve the failure is supported by formal results rather than by an empirical failure to find a sufficiently large model. Non-permutation-invariance (Section 4.2) follows from positional encoding and causal ordering, and Kozachinskiy et al.'s ceiling holds for arbitrarily large width by construction. The compositional-reasoning ceiling has the same status: the limitation follows from the computational form of the attention mechanism rather than from insufficient parameter count. These are therefore high-confidence claims under the confidence convention of Section 7.3.

The remaining two mechanisms require more cautious interpretation. Benchmark contamination (Sections 4.1 and 4.3) is an evaluation-mismatch problem: additional data, scale, or architectural changes cannot retrospectively make a contaminated benchmark measure clean generalization. But this conclusion follows from the definition and consequences of contamination rather than from a theorem about attention or representation learning. The vision-to-language information bottleneck (Section 4.4) is weaker still. It appears representation-solvable in the specific sense that discretized symbolic spatial encodings substantially improve performance on the tested tasks, while Alam et al. [3] find no resolution through the encoder and training-objective variations they examine. That negative result is informative, but it is a single controlled study rather than a proof that scaling or other architectural changes could never close the gap.

Section 7.1 further argues that two of these four mechanisms (non-permutation-invariance and the compositional ceiling) have a domain-agnostic mechanistic explanation rooted in properties of attention. This should not be generalized to all four mechanisms. The available proofs establish why two failure modes recur; they do not constitute a formal cross-domain meta-analysis demonstrating that every failure in Table 6 has the same cause. The recurrence is therefore strong evidence for a shared mechanism in these two cases, while the remaining mechanisms should be treated with the weaker confidence warranted by their empirical evidence.

The conceptual framework introduced by this review carries a corresponding limitation. The four-way distinction in Section 3.3 is a synthesis proposed here. Its components nevertheless correspond to distinctions already present in the theoretical literature reviewed in Section 6.2: describing a structure and computationally implementing it are different properties, and the contribution of this review is to apply that distinction systematically across structured non-linguistic data modalities. What the framework does not establish is a quantitative relationship between structural complexity and the resulting performance or efficiency gap. Section 8 specifies the controlled experiment required to measure that relationship.

Table 6: Recurring limitations of language-mediated learning, organized by underlying mechanism rather than surface symptom (Section 7.4). Across the four core modalities, the same mechanisms recur, suggesting that these limitations arise from the representational regime rather than from any single domain.

Mechanism	Underlying cause	Primary impact	Evidence
Order sensitivity	Positional encoding and causal ordering introduce dependence on the presentation order of the input.	Set- and graph-structured inputs	Provably non-permutation-invariant [19]; encoding choices can cause 61.8-point performance swings [21], with failures on isomorphic graphs [86].
Compositional ceiling	A single softmax-attention layer has a width- and precision-independent VC-dimension ceiling for specific relational computations.	Multi-hop reasoning, execution semantics, and deep feature interactions	Theoretical limitation proven in relational tasks [46]; empirically reflected in accuracy collapse with reasoning depth [18] and near-chance two-hop composition [7].
Benchmark contamination	Pretraining can expose models to popular benchmarks; instruction tuning may therefore reproduce apparent structural competence without learning the underlying structure.	Few-shot tabular and time-series benchmarks	Two independent 2026 tabular audits report substantial contamination effects [28, 75]; analogous leakage is documented for time-series benchmarks [58].
Vision–language bottleneck	Information retained by the visual encoder is not necessarily expressed or exploited by the language decoder.	Spatial and quantitative VLM reasoning	The information bottleneck is directly localized [113] and persists across encoder, positional-encoding, and objective variants [3].

8 Open Problems and Future Work

8.1 The precise gap

Existing empirical work evaluates language-based representations within individual structural modalities and individual inductive biases, almost always on real-world benchmarks confounded by semantic priors, memorization, and contamination (Section 5). Learning theory has independently supplied several of the relevant ingredients (sample-complexity bounds for transformers [105], universal-approximation costs of permutation-invariant architectures [79], a VC-dimension ceiling on compositional tasks [46], and a phase transition governing when in-context learning generalizes structurally [26]), but no study has connected these ingredients to a controlled, cross-regime empirical measurement. A dedicated search for such a study, conducted specifically for this work and did not locate one. The resulting gap is therefore an untested empirical relationship.

The missing experiment can be stated compactly, and is best understood as one cross-section of a larger surface that this review has already partially identified. Let L_{general} denote the loss of a language-mediated model on a task and $L_{\text{specialized}}$ the loss of an architecture designed to encode that task's structure explicitly, with training and evaluation otherwise matched. Section 6.3 defines

$\Delta(N) = L_{\text{general}}(N) - L_{\text{specialized}}(N)$ as the performance gap as a function of model or data scale N, and shows that no existing study measures it. The complementary quantity is the same gap as a function of structural complexity c at fixed scale: $\Delta(c) = L_{\text{general}}(c) - L_{\text{specialized}}(c)$, with information content and sample budget held fixed while the source of structural bias is varied (an architectural constraint, an explicit natural-language instruction, or an in-context demonstration) across multiple structural regimes. The evaluation should use synthetic tasks with known ground truth and the direct invariance-violation metric of Table 4. Together, $\Delta(N)$ and $\Delta(c)$ define a single unmeasured surface. The contribution proposed here is therefore to specify the $\Delta(c)$ cross-section precisely enough to make it experimentally testable.

The gap is falsifiable. A flat or non-monotonic $\Delta(c)$ curve under the controls above would refute the central empirical hypothesis developed in Section 6. It is also experimentally tractable using the metric already defined in Table 4. More importantly, it provides a common axis for several apparently conflicting results in Section 4: SimKGC and MoLFormer appear to support substitution, whereas Tan et al.'s and Kozachinskiy et al. [46]'s results appear to challenge it. These results occupy different, previously unconnected points on the structural-complexity axis. No existing paper, however, measures that axis directly.

8.2 Research questions and hypotheses

Each question below sets competing hypotheses against one another that make distinct, observable predictions.

RQ1. For a fixed synthetic function class, how does the sample efficiency of in-context learning compare with that of a matched specialized architecture as structural complexity (permutation-group size, locality radius, interaction sparsity, or compositional depth) increases? Kozachinskiy et al. [46] prove the one-layer case for standard softmax attention: none of three compositional tasks (three-way matching, function composition, binary-relation composition) is solvable, regardless of width or precision, whereas their proposed Strassen attention solves all three at one layer. Their proof does not establish what standard, unmodified multi-layer attention can do on the same tasks; their own discussion identifies this as future work. **H1 (architectural ceiling):** the impossibility persists at any constant depth for standard attention, so the barrier reflects what softmax attention computes rather than the width of a single layer. **H2 (depth suffices):** additional layers, without changing the attention mechanism itself, recover the missing compositional operations, making the one-layer result a depth limitation rather than a limitation of the mechanism. These hypotheses make different predictions as depth increases at fixed parameter count: H1 predicts no improvement on any of the three tasks regardless of depth; H2 predicts improvement attributable to depth before parameter count alone would explain it. **Falsification of H1:** a many-layer standard-attention model closing the gap on any of the three tasks without a Strassen-style architectural change. **Confound:** an apparent depth effect caused by the additional parameters introduced by increasing depth controllable by matching parameter count across depths.

RQ2. When a structural invariance is stated explicitly in language, does the resulting output distribution become measurably closer to invariant under the direct metric of Section 6.1? **H1 (bias is inert):** the residual gap remains flat or increases with dimensionality, consistent with Yoon et al.'s prompting-only attempt before positional-encoding surgery was required. **H2 (bias is dilute but real):** instruction narrows the gap gradually, at a rate too small to have been detected in Yoon et al.'s particular tasks but detectable in aggregate. **Critical experiment:** measure the invariance metric directly across a dimensionality sweep rather than infer invariance from downstream task

accuracy, which cannot distinguish H1 from H2 at the required resolution. **Falsification of H1:** a measurable narrowing that remains stable across dimensionality.

RQ3. Do demonstrations close more of the gap than explicit instruction alone, and can their effect be distinguished from memorization? **H1 (in-support generalization):** Wakayama and Suzuki's exponential-decay result predicts that demonstrations provide rapidly diminishing risk when the target structure lies within the support of the pretraining task mixture. **H2 (out-of-support failure):** Goddard et al.'s phase transition instead predicts a qualitative break once the target structure leaves that support, instead of a smooth continuation of H1's decay. In their linear-function setting, a transformer pretrained on tasks spanning less than approximately 120° of task-angle diversity fails on unseen tasks outside that span, whereas diversity beyond roughly 120° (shifting to approximately 135° under added label noise) yields a solution that generalizes across the full test range. This is a measured threshold, not merely a qualitative claim of "sharpness." **Critical experiment:** vary the target task's distance from the pretraining mixture directly in a setting with a matched structural-complexity axis rather than Goddard et al.'s task-angle axis, and test whether the empirical curve follows H1's smooth decay or instead exhibits a comparable transition. **Confound:** apparent in-support generalization that is actually retrieval of a memorized near-duplicate, controllable through the contamination canary in Section 8.3.

RQ4. Which measurable task properties (symmetry-group order, minimum description length, or discretizability) predict in advance whether the gap will be small or catastrophic? The discrete-versus-continuous split reported in Section 4.4, where text-symbol grids outperform pixel grids on identical tasks while both fail on continuous geometry, provides the strongest existing empirical indication that discretizability may be an important moderator. Tabaghi and Wang's $2DN$ parameter-cost formula provides a natural, citable complexity variable against which this hypothesis can be tested. **H1** is that discretizability alone predicts the gap's magnitude. **H2** is that discretizability is one of several independent moderators, including symmetry-group order and minimum description length, that cannot be reduced to a single variable. **Critical experiment:** hold discretizability fixed while varying symmetry-group order and MDL independently, and test whether the gap changes.

Table 7 restates these questions as five independently addressable sub-problems, each paired with the reason it remains open and a concrete experimental direction.

8.3 Minimal experimental design

The design follows directly from the gap defined in Section 8.1 and is intended to resolve RQ1–RQ4 through a common protocol. For each of a small set of task families with known generating functions (permutation invariance; a direct replication of all three of Kozachinskiy et al.'s compositional impossibility tasks, three-way matching, function composition, and binary-relation composition, tested separately because their one-layer proof treats them as distinct and their multi-layer behavior remains unknown for all three; locality; sparse tabular interaction; and graph neighborhood aggregation), five representation arms are compared:

1. a specialized architecture matched to the target bias;
2. naïve text serialization with no structural hint;
3. serialization plus an explicit statement of the relevant invariance;

Table 7: Open problems this review identifies for future directions (Section 8). The first row is the paper's central gap statement; the remaining rows are narrower, independently addressable sub-problems.

Problem	Why unresolved	Promising direction
No cross-regime, information-controlled measurement of the architecture-vs-language gap	Every existing study varies modality *and* representation *and* real-world confounds simultaneously	Synthetic task families with known ground truth, five matched representation arms, and a direct invariance-violation metric (Section 8.3)
No closed-form connection between proven complexity-theoretic ceilings and an empirical structural-complexity variable	Kozachinskiy et al.'s VC-dimension result and Goddard et al.'s phase transition are each demonstrated in isolation, on different task families	Replicate the compositional-task impossibility result directly, extended with instruction and demonstration arms not covered by the original proof
Knowledge-graph completion's positive result is unexplained mechanistically	SimKGC's win is attributed to textual side-information post hoc, not tested directly	An ablation removing entity/relation text descriptions while preserving triples, to isolate information availability from structural inference
Scaling-law exponents are reported in absolute terms, never relative to a matched-bias baseline	No paper fits a joint curve of (architecture accuracy − language accuracy) against scale	Report scaling curves as a gap-vs-scale plot, not two separate loss-vs-scale plots
Level-6 orchestration systems are evaluated on task success, not on what fraction of the computation the LLM itself performs	Production AutoML-agent papers report end-task metrics, not a decomposition of LLM vs. tool contribution	An ablation analogous to Tan et al. [81], applied to orchestration agents rather than forecasters

4. serialization plus demonstrations only (no explicit statement), isolating whether demonstrations alone can convey the bias;
5. a lossless canonical serialization, providing the best-case upper bound attainable if representation loss were the sole source of any observed gap.

Within the taxonomy of Section 3.1, arms 2–4 occupy Levels 1–4 and arm 1 occupies Level 8. The design deliberately excludes Levels 5–7 because the question is whether the language-mediated computation itself acquires the relevant bias, rather than whether an orchestrated tool or separate encoder can supply it, a question Section 4.6 and Table 1's orchestration row already answer affirmatively for a different reason. Contamination is controlled by sampling fresh generating functions on every run, contrasting randomized-symbol feature names with real-sounding names to isolate world-knowledge leakage, and holding out one structural regime never referenced during development as a contamination canary. The primary dependent variable is the direct invariance-violation metric defined in Section 6.1, not accuracy. Compute cost per prediction is reported alongside it so that any apparent advantage remains interpretable against Tan et al.'s demonstration that accuracy parity can coexist with a 1000× compute disadvantage. Together, these measurements populate the dimensions of Table 1 that end-task accuracy alone cannot. Functional replacement (arm comparison at fixed accuracy), computational replacement (cost per prediction), and structural replacement (the direct invariance metric) are each measured in this case.

8.4 Concrete future directions

Beyond the primary experiment in Section 8.3, four narrower and independently valuable directions follow directly from gaps identified in Sections 4 and 6.

1. Report structured-data scaling laws as $\Delta(N)$-versus-scale curves relative to a matched-bias baseline instead of isolated loss-versus-scale curves. Section 6.3 shows that this connective analysis is absent from every scaling-law paper located in the review, even though it requires no new training run and it can be obtained by re-analyzing quantities that most of these papers already collected and reported.

2. Test empirically whether depth alone, while keeping the attention mechanism otherwise standard, closes the gap on Kozachinskiy et al.'s three compositional tasks (three-way matching, function composition, and binary-relation composition), treating them separately because their one-layer proof treats them as distinct and their paper identifies the multi-layer case as open for all three. This is feasible as an empirical study on the small synthetic tasks already specified by their proof and requires modest compute, well before the harder theoretical question of proving a corresponding multi-layer impossibility result, if H1 in RQ1 is correct, is resolved. The experimental design in Section 8.3 is constructed to run precisely this test.

3. Perform an ablation on SimKGC analogous to Tan et al.'s time-series ablation by removing textual entity and relation descriptions while keeping the underlying triples fixed. This would determine how much of SimKGC's advantage over embedding-based methods arises from information availability rather than genuine exploitation of structure, a distinction currently inferred instead of directly tested in Section 4.2. This review searched specifically for a verified prior instance of this exact test and did not locate one meeting the evidentiary standard applied elsewhere in the review; the direction is therefore treated as fully open. It is highly feasible because SimKGC's codebase and the WN18RR/FB15k-237 benchmarks are public and small, making the ablation a relatively inexpensive.

4. Apply a Tan et al.-style component ablation to Level-6 orchestration agents (Section 4.6), decomposing end-task success into the contribution attributable to the language model's search and planning versus that of the specialized tool being orchestrated. Current evaluations report primarily end-task metrics and therefore cannot distinguish genuine orchestration capability from cases in which the tool performs the substantive computation largely independently of the orchestrator's quality.

9 Conclusion

Across nine modalities and four methodological lenses, this review finds that behavioral competitiveness should not be conflated with replacement. Language-mediated systems can match specialized architectures on predictive performance, but where representation, computation, or architecture has been tested directly, such parity rarely extends beyond the narrower senses of replacement distinguished in Table 1. Language can substitute for explicit inductive bias in identifiable regimes, including extreme few-shot prediction, discretizable spatial layouts, textually annotated knowledge graphs, and settings where sufficiently large pretraining compensates for limited task-specific structure. Beyond these regimes, apparent success is often associated with contamination, format

familiarity, or information supplied through channels that are not purely linguistic. The central methodological lesson is therefore simple: accuracy, statistical efficiency, computational cost, and structural equivalence are different claims, and benchmark performance alone cannot establish the latter.

The evidence also points to a recurring cross-modal pattern. When language-only representations prove insufficient, successful systems frequently reintroduce non-linguistic structure through tools, specialized tokens, or architectural scaffolds. What appears to be the disappearance of specialization is therefore often its relocation. This pattern recurs across independent research communities, although the available evidence does not establish it as a universal law. Similarly, the literature provides clear evidence that language can serve as an effective interface for directing and composing specialized components, but this should be understood as a distinct form of integration rather than as architectural replacement. Scaling is also real, but the relevant unresolved quantity is not whether $L(N)$ decreases with scale; it is whether the gap to a matched-bias architecture, $\Delta(N)$, decreases and eventually closes (Section 6.3).

The central open question is consequently more precise than whether language can replace specialized machine learning. Existing theoretical results, including the compositional-task VC-dimension ceiling of Kozachinskiy et al. [46] and the pretraining-diversity phase transition of Goddard et al. [26], provide important constraints in their respective settings, but neither directly predicts the performance gap as a function of scale or structural complexity (Sections 6.3 and 8.1). The next step is therefore to identify what predicts, quantitatively and in advance, where language-mediated systems cease to substitute for specialized structure. Section 8 outlines a falsifiable framework for answering this question and for moving the field from demonstrations of benchmark parity toward a theory of when, why, and at what structural cost such parity is achievable.

References

[1] Nikhil Abhyankar, Parshin Shojaee, and Chandan K. Reddy. LLM-FE: Automated feature engineering for tabular data with LLMs as evolutionary optimizers. *Transactions on Machine Learning Research*, 2025.

[2] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *International Conference on Learning Representations (ICLR)*, 2023.

[3] Nahid Alam, Leema Krishna Murali, Siddhant Bharadwaj, Patrick Liu, Timothy Chung, Drishti Sharma, Akshata A., Kranthi Kiran, Wesley Tam, and Bala Krishna S Vegesna. Spatial reasoning is not a free lunch: A controlled study on LLaVA, 2026.

[4] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, et al. Chronos: Learning the language of time series, 2024.

[5] Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning, 2025. NeurIPS 2025.

[6] Waiss Azizian and Ali Hasan. How does the pretraining distribution shape in-context learning? a fundamental trade-off, 2025.

[7] Mikita Balesni, Tomek Korbak, and Owain Evans. The two-hop curse: LLMs trained on A→B, B→C fail to learn A→C, 2024.

[8] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, et al. Relational inductive biases, deep learning, and graph networks, 2018.

[9] Maya Bechler-Speicher, Yoel Gottlieb, Andrey Isakov, David Abensur, Ami Tavory, Daniel Haimovich, Ido Guy, and Udi Weinsberg. Billion-scale graph foundation models, 2026.

[10] Sebastian Bordt, Harsha Nori, Vanessa Rodrigues, Besmira Nushi, and Rich Caruana. Elephants never forget: Memorization and learning of tabular data in large language models. In *Conference on Language Modeling (COLM)*, 2024.

[11] Konstantinos Bougiatiotis, Dimitrios Kelesis, and Georgios Paliouras. Improving molecular property prediction in small language models using graph-based tools, 2026.

[12] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. 2021.

[13] Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large language and graph assistant. In *International Conference on Machine Learning (ICML)*, 2024.

[14] Shao-Ting Chiu, Junyuan Hong, and Ulisses Braga-Neto. Deeposets: Non-autoregressive in-context learning with permutation-invariance inductive bias, 2024.

[15] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning (ICML)*, 2016.

[16] Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. LIFT: Language-interfaced fine-tuning for non-language machine learning tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[17] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

[18] Nouha Dziri, Ximing Lu, Melanie Sclar, et al. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems (NeurIPS), Spotlight*, 2023.

[19] Beni Egressy and Jan Stühmer. Set-LLM: A permutation-invariant LLM, 2025.

[20] Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. Large language models (llms) on tabular data: Prediction, generation, and understanding – a survey, 2024. Transactions on Machine Learning Research.

[21] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[22] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision (ECCV)*, 2024.

[23] Josh Gardner, Juan C. Perdomo, and Ludwig Schmidt. Large scale transfer learning for tabular data via language modeling, 2024. NeurIPS 2024; central claim independently overturned by Gorla and Puduppully [28].

[24] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[25] Marta Garnelo and Wojciech M. Czarnecki. Why large language models fail at tabular prediction, 2026. Unverified: no independent replication found at time of review.

[26] Ben Goddard, Kai Smith, Vudtiwat Ngampruetikorn, and David J. Schwab. When can in-context learning generalize out of task distribution? In *International Conference on Machine Learning (ICML)*, 2025.

[27] Shahriar Golchin and Mihai Surdeanu. Time travel in LLMs: Tracing data contamination in large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[28] Aditya Gorla and Ratish Puduppully. The illusion of generalization in tabular language models. In *International Conference on Machine Learning (ICML)*, 2026.

[29] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2022.

[30] Léo Grinsztajn, Klemens Flöge, Oscar Key, et al. TabPFN-2.5: Advancing the state of the art in tabular foundation models, 2025. Prior Labs.

[31] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. LLMTime.

[32] Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I. Wang. CRUXEval: A benchmark for code reasoning, understanding and execution. In *International Conference on Machine Learning (ICML)*, 2024.

[33] Yang Han, Ziping Wan, Lu Chen, Kai Yu, and Xin Chen. From generalist to specialist: A survey of large language models for chemistry. In *International Conference on Computational Linguistics (COLING)*, 2025.

[34] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. TabLLM: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

[35] Daniel Herbst, Lea Karbevska, Divyanshu Kumar, Akanksha Ahuja, Fatemeh Gholamzadeh Nasrabadi, and Fabrizio Frasca. Lost in serialization: Invariance and generalization of LLM graph reasoners, 2025. Verified via arXiv abstract page (title, authors, and finding cross-checked directly; not a full-text read of the PDF). Finds larger non-fine-tuned models more robust to serialization changes than fine-tuned variants, and that fine-tuning trades reduced node-relabeling sensitivity for increased sensitivity to structure/format changes – extends Fatemi et al. [21] rather than merely repeating it.

[36] Joshua Hill, Benjamin Eyre, and Elliot Creager. Transformers don't in-context learn least squares regression, 2025. ICML 2025 Workshop on Reliable and Responsible Foundation Models – workshop tier, not full peer review. Verified via direct fetch of the arXiv abstract. Empirical counterevidence to the claim that transformers implement OLS/gradient-descent-like algorithms in-context: shows ICL transformers fail to generalize under prompt-distribution shift and instead appear to memorize training-distribution spectral signatures, located during a dedicated adversarial search for counterevidence to the positive in-context-learning results cited in Section 6.2.

[37] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023.

[38] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025.

[39] Carlos Huertas. Gradient boosting trees and large language models for tabular data few-shot learning, 2024. FedCSIS 2024 Data Mining Competition.

[40] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, et al. Perceiver IO: A general architecture for structured inputs & outputs, 2021.

[41] Alireza Jafari, Judy Fox, Geoffrey C. Fox, Madhav Marathe, and Aniruddha Adiga. Understanding key features of time series foundation models from epidemic forecasting, 2026.

[42] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey, 2023.

[43] Ming Jin, Shiyu Wang, Lintao Ma, et al. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*, 2024. Falsified by ablation in Tan et al. [81].

[44] Juno Kim, Tai Nakamaki, and Taiji Suzuki. Transformers are minimax optimal nonparametric in-context learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[45] Myung Jun Kim, Léo Grinsztajn, and Gaël Varoquaux. CARTE: Pretraining and transfer for tabular learning. In *International Conference on Machine Learning (ICML)*, 2024.

[46] Alexander Kozachinskiy, Felipe Urrutia, Hector Jimenez, Tomasz Steifer, Guido Pizarro, Ivan Fuentes, Diego Meza, Francisco Calderon, and Cristóbal Rojas. Strassen attention, split VC dimension and compositionality in transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[47] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

[48] Young-Jun Lee, Byungsoo Ko, Han-Gyu Kim, Yechan Hwang, and Ho-Jin Choi. Intriguing properties of large language and vision models. In *arXiv preprint*, 2024.

[49] Xingwu Li, Yuling Jiao, Jian Huang, Yiming Wei, and Xiliang Chen. Transformers meet in-context learning: A universal approximation theory, 2025.

[50] Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, and Chuan Shi. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. Cited as reference [14] within Sun et al. [78]; located via citation-chaining from that survey's primary text rather than an independent search pass.

[51] Jingzhe Liu, Haitao Mao, Zhikai Chen, Tong Zhao, Neil Shah, and Jiliang Tang. Towards neural scaling laws on graphs, 2024.

[52] Peiyuan Liu, Hang Guo, Tao Dai, et al. CALF: Aligning LLMs for time series forecasting via cross-modal fine-tuning, 2024. Falsified by ablation in Tan et al. [81].

[53] Do Xuan Long, Hai Nguyen Ngoc, Tiviatis Sim, Hieu Dao, Shafiq Joty, Kenji Kawaguchi, Nancy F. Chen, and Min-Yen Kan. LLMs are biased towards output formats! systematically evaluating and mitigating output format bias of LLMs. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 299–330, 2025. Verified via ACL Anthology (2025.naacl-long.15); located during a dedicated search of the wider evaluation-methodology literature, outside this review's core 195-paper corpus, for evidence of benchmark bias running against rather than toward language-mediated methods – see Section 5.

[54] Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Hamidreza Kamkari, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[55] Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 2024. Feature Review.

[56] Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. In *International Conference on Machine Learning (ICML), Position Paper Track*, 2024.

[57] Haitao Mao, Guangliang Liu, Yao Ma, Rongrong Wang, Kristen Johnson, and Jiliang Tang. A survey to recent progress towards understanding in-context learning. In *Findings of the Association for Computational Linguistics: NAACL*, 2025.

[58] Marcel Meyer, Sascha Kaltenpoth, Kevin Zalipski, and Oliver Müller. Rethinking evaluation in the era of time series foundation models: (un)known information leakage challenges, 2025.

[59] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Édouard Grave, Yann LeCun, and Thomas Scialom. Augmented language models: A survey. *Transactions on Machine Learning Research*, 2023. Verified via dblp (journals/tmlr/MialonDLNPRRSDC23); distinguishes reasoning-augmentation from tool/action-augmentation along a different organizing axis than Section 3.1's location-of-computation axis – see Section 3.1.

[60] Sarthak Mittal, Yang Li, I-Cheng Yen, Deniz Guetta, and Hongseok Namkoong. Architectural and inferential inductive biases for exchangeable sequence modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[61] Jaehyun Nam, Jinsung Yoon, Jiefeng Chen, Jinwoo Shin, Sercan Ö. Arık, and Tomas Pfister. MLE-STAR: Machine learning engineering agent via search and targeted refinement, 2025. Google.

[62] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations (ICLR)*, 2023. PatchTST.

[63] Pascal Notin, Aaron W. Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora S. Marks. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. Verified via the NeurIPS proceedings record; Su et al. [77] adopts this benchmark, alongside a ten-task suite including Thermostability, Metal Ion Binding, DeepLoc, EC/GO annotation, and HumanPPI, for downstream evaluation.

[64] Catherine Olsson, Nelson Elhage, Neel Nanda, et al. In-context learning and induction heads, 2022. Transformer Circuits Thread.

[65] Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjørn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, and David Moher. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372:n71, 2021. doi: 10.1136/bmj.n71.

[66] Zhuoyi Peng, Jingzhou Jiang, Hanlin Gu, Lixin Fan, and Yi Yang. GraphInfer-Bench: Benchmarking LLM's inference capability on graphs, 2026.

[67] Lennart Purucker, Andrej Tschalzev, Nick Erickson, et al. Beyond IID: How general are tabular foundation models, really?, 2026.

[68] Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model, 2026.

[69] Paul Quinlan, Jeremy Levasseur, Qingguo Li, and Xiaodan Zhu. Chronicle: A multimodal foundation model for joint language and time series understanding, 2026.

[70] Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind: Failing to translate detailed visual features into words. In *Asian Conference on Computer Vision (ACCV)*, 2024.

[71] Scott Reed, Konrad Zolna, Emilio Parisotto, et al. A generalist agent. *Transactions on Machine Learning Research*, 2022. Gato.

[72] Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh Chawla, and Chao Huang. A survey of large language models for graphs, 2024.

[73] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022. MoLFormer.

[74] Rohan Saxena, Alessandro Suglia, and Pasquale Minervini. VLM-RobustBench: A comprehensive benchmark for robustness of vision-language models, 2026.

[75] Matteo Silvestri, Fabiano Veglianti, Flavio Giorgi, Fabrizio Silvestri, and Gabriele Tolomei. When large language models know the table: A framework for assessing data contamination in tabular datasets, 2026.

[76] Xingyou Song et al. OmniPred: Language models as universal regressors. *Transactions on Machine Learning Research*, 2024. Google DeepMind.

[77] Jin Su, Chenchen Han, et al. SaProt: Protein language modeling with structure-aware vocabulary. In *International Conference on Learning Representations (ICLR)*, 2024.

[78] Qingyun Sun, Haonan Yuan, Yi Huang, Ziwei Zhang, Xingcheng Fu, Ruijie Wang, Haoyi Zhou, Jia Wu, Jianxin Li, and Philip S. Yu. A survey on foundation models for structured data: Tabular, time series, and graphs. Preprints.org, 2026. doi:10.20944/preprints202605.1532.v2. Posted 28 May 2026; explicitly labeled "Not peer-reviewed version" by the platform (Preprints.org is an MDPI-affiliated preprint server, not IEEE TKDE – an earlier draft of this manuscript misattributed the venue based on an ambiguous secondary source and has been corrected after obtaining and reading the primary PDF directly; see Section 2.2). Full text read in full for this review.

[79] Puoya Tabaghi and Yusu Wang. Universal representation of permutation-invariant functions on vectors and tensors. In *International Conference on Machine Learning (ICML)*, 2024.

[80] Jiarui Tan, Zhongjian Zhang, YaBo Guo, Jiawei Liu, Yujie Xing, Muhan Zhang, Cheng Yang, and Chuan Shi. GABench: A comprehensive benchmark for evaluating LLM agents on graph analysis tasks, 2026.

[81] Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems (NeurIPS), Spotlight*, 2024.

[82] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. GraphGPT: Graph instruction tuning for large language models. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.

[83] Yi Tay, Mostafa Dehghani, et al. Scaling laws vs model architectures: How does inductive bias influence scaling? In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

[84] Himanshu Thakur, Anusha Kamath, Anurag Muthyala, Dhwani Sanmukhani, Smruthi Mukund, and Jay Katukuri. Towards reliable ML feature engineering via planning in constrained-topology of LLM agents, 2026.

[85] Vishal Thengane, Xiatian Zhu, Salim Bouzerdoum, Son Lam Phung, and Yunpeng Li. Foundational models for 3d point clouds: A survey and outlook, 2025.

[86] Y. Thushalika, S. Kishanthan, and Asela Hevapathige. Detecting differences is not understanding structure: Large language models fail at graph isomorphism, 2026. Unverified: very recent preprint, not yet independently replicated.

[87] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. MMVP.

[88] Keyon Vafa, Justin Y. Chang, Ashesh Rambachan, and Sendhil Mullainathan. What has a foundation model found? using inductive bias to probe for world models. In *International Conference on Machine Learning (ICML)*, 2025.

[89] Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning (ICML)*, 2023.

[90] Tomoya Wakayama and Taiji Suzuki. In-context learning is provably bayesian inference: A generalization theory for meta-learning, 2025.

[91] Duolin Wang et al. S-PLM: Structure-aware protein language model via contrastive learning between sequence and structure. *Advanced Science*, 2025.

[92] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[93] Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. SpatialEval.

[94] Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. 2022.

[95] Ruiyu Wang, Zifeng Wang, and Jimeng Sun. UniPredict: Large language models are universal tabular classifiers, 2023.

[96] Zehong Wang, Zheyuan Liu, Tianyi Ma, et al. Graph foundation models: A comprehensive survey, 2025. Author list corrected from an earlier draft's Wang, Xin and Liu, Sike and Ma, Yushun, which did not match the arXiv record; verified against arxiv.org/abs/2505.15116.

[97] Azmine Toushik Wasi, Wahid Faisal, Abdur Rahman, Mahfuz Ahmed Anik, Munem Shahriar, Mohsin Mahmud Topu, Sadia Tasnim Meem, Rahatun Nesa Priti, Sabrina Afroz Mitu, Md. Iqramul Hoque, Shahriyar Zaman Ridoy, Mohammed Eunus Ali, Majd Hawasly, Mohammad Raza, and Md Rizwan Parvez. SpatiaLab: Can vision-language models perform spatial reasoning in the wild? In *International Conference on Learning Representations (ICLR)*, 2026. Verified via arXiv abstract page; ICLR 2026 venue confirmed. Note: an earlier internal draft of this review's notes mistakenly conflated this paper with a differently-titled benchmark ("Spatial-DISE," arXiv:2510.13394) found in the same search pass – that conflation has been corrected; this entry cites only the paper actually verified at arXiv:2602.03916.

[98] Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their prompts? 2022.

[99] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[100] Xiaofeng Wu, Alan Ritter, and Wei Xu. Tabular data understanding with large language models: A survey of recent advances and challenges, 2025.

[101] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018. doi: 10.1039/c7sc02664a. Verified via the publisher record (Royal Society of Chemistry); this is the benchmark suite Ross et al. [73]'s "ten molecular-property benchmarks" draws on (BBBP, Tox21, ClinTox, HIV, BACE, SIDER, QM8, QM9, ESOL, FreeSolv, lipophilicity).

[102] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. Verified via the CVPR open-access proceedings; introduces the ModelNet benchmark, whose ModelNet40 test split Xu et al. [104] uses for generative 3D object classification.

[103] Yijia Xiao, Wanjia Zhao, et al. Protein large language models: A comprehensive survey, 2025. EMNLP 2025 Findings.

[104] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. PointLLM: Empowering large language models to understand point clouds. In *European Conference on Computer Vision (ECCV)*, 2024.

[105] Yiwei Yang, Nathan Srebro, and Yuanzhi Li. Tight sample complexity of transformers. In *Conference on Learning Theory (COLT)*, 2026.

[106] Liang Yao, Jiazhen Peng, Chengsheng Mao, and Yuan Luo. Exploring large language models for knowledge graph completion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[107] Qingren Yao, Chao-Han Huck Yang, Renhe Jiang, Yuxuan Liang, Ming Jin, and Shirui Pan. Towards neural scaling laws for time series foundation models. In *International Conference on Learning Representations (ICLR)*, 2025.

[108] Soyoung Yoon et al. Towards more reliable responses for order-invariant inputs. In *Association for Computational Linguistics (ACL)*, 2025. RoToR.

[109] Zhuowen Yuan, Tao Liu, Yang Yang, Yang Wang, Feng Qi, Kaushik Rangadurai, Bo Li, and Shuang Yang. ArchPilot: A proxy-guided multi-agent approach for machine learning engineering, 2025.

[110] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence*, 2023. DLinear.

[111] Weichen Zhang et al. The point, the vision and the text: Does point cloud boost spatial reasoning of large language models? a bias-controlled study, 2025.

[112] Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K. Gupta, and Jingbo Shang. Large language models for time series: A survey, 2024.

[113] Yunkai Zhang, Linda Li, Yingxin Cui, Xiyuan Ruan, Zeyu Zheng, Kezhen Chen, Yi Zhang, and Diji Yang. Grid2Matrix: Revealing digital agnosia in vision-language models, 2026.

[114] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems (NeurIPS), Spotlight*, 2023. GPT4TS; falsified by ablation in Tan et al. [81].

[115] Yuxiang Zhou, Jiazheng Li, Yanzheng Xiang, Hanqi Yan, Lin Gui, and Yulan He. The mystery of in-context learning: A comprehensive survey on interpretation and analysis. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2024.