

Cognitive Complexity for Auditable Evaluation of Knowledge Graph Change Benchmarks

Sara Todorovikj^{1,*}, Michael Martin¹

¹Chemnitz University of Technology, Germany

Abstract

Benchmarks for knowledge graph change evaluate systems with a single recall-oriented score that reports how often a system was right, but not what the task required to produce that outcome. For governance and explainability this gap is significant, since a scalar cannot reveal whether a system failed due to lack of reasoning a task needs or succeeded by exploiting a regularity in the data. That means it is not able to support auditing or accountable deployment. We address this with a system-agnostic, cognitively grounded characterization of task demand along three dimensions (cognitive process, knowledge type and relational complexity), assigned a priori from an operation's definition and serving as an explanatory layer over the score. We apply it to text-driven knowledge graph updating and temporal forecasting benchmarks and show how operations differ in demand, along with how a documented shortcut exploits this blindness. It is designed to support demand-resolved auditing and yields a criterion for when a benchmark can be governed at all.

Keywords

knowledge graph benchmarks, evaluation beyond accuracy, auditable evaluation, cognitive complexity, task characterization

1. Introduction

Benchmarks are how the field decides what systems can do, and those judgments increasingly carry into deployment, including regulated and high-stakes settings, where the decision to rely on a system must be defensible to an external party. This places a demand on evaluation that the dominant practice does not meet, namely, evaluation beyond accuracy. Knowledge-graph systems are typically scored by how often their outputs are correct and such a number reports an outcome without saying what the task required to produce it.

In this article we focus on knowledge-graph change, from graph manipulating operations, such as updating or revising, to forecasting facts as the world moves on. In these scenarios, that gap is especially consequential and a system can score well without performing the operation a task names.

What is missing is a statement, independent of any system, of what a task demands. Towards that, we propose a characterization framework for knowledge-graph operations along three cognitively grounded dimensions that together form a compact demand profile. Because the profile is fixed a priori from an operation's definition and describes the task rather than the solver, it is system-agnostic and serves as an explanatory layer over the score. In the following we will present the characterization and illustrate how it is applied to two settings: text-driven knowledge graph updating and temporal forecasting.

2. Governance Gap

Deploying a knowledge graph system in a regulated or high-stakes setting is a governance problem. An external party must be able to examine what a system was tested on and confirm that the testing was meaningful for the use at hand. This requires capability claims expressed in terms of what tasks demand

The International Workshop on Semantic Architectures for Governance and Explainability (SAGE 2026), co-located with SEMANTiCS 2026: 22nd International Conference on Semantic Systems, September 15, 2026, Ghent, Belgium

*Corresponding author.

✉ sara.todorovikj@informatik.tu-chemnitz.de (S. Todorovikj)

ORCID 0000-0002-2418-1358 (S. Todorovikj); 0000-0003-0762-8688 (M. Martin)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

of a system, in contrast to aggregate measures of how often it was right. Internal-audit frameworks for AI make the same point from the developer’s side, that each stage of evaluation should yield evidence that a reviewer can later interrogate [1].

Yet a bare figure is essentially all that most benchmarks for knowledge graph change report. For example, the knowledge graph updating benchmark EMERGE uses completeness and a recall-oriented similarity measure [2]. Temporal forecasting benchmarks report Mean Reciprocal Rank and Hits@ k [3]. OWL reasoner competitions measure performance with the number of problems solved within a timeout [4]. Ultimately, capability is summarized by one recall- or count-based scalar, providing no account of what makes one operation more demanding than another.

This type of metric measures an outcome and not the operation that produced it. That gap carries a significant governance cost. We call a benchmark *gameable* when a system can earn high scores without performing the operation the benchmark was meant to test, by exploiting a statistical regularity in the data, i.e. a surface shortcut, in place of the genuine operation. In temporal forecasting, a heuristic that counts how often facts have co-occurred, while ignoring time entirely, comes close to state-of-the-art [5]. Because a scalar cannot separate the genuine operation from its shortcut, the evaluation is *ungovernable*, as there is no principled basis for setting requirements or comparing systems. Moreover, it is also not *auditable*, since an external party cannot establish what a system can or cannot do beyond the aggregate it is handed.

What governance requires here and what these benchmarks lack is a shared vocabulary for what tasks demand. A characterization of each operation that is fixed independently of any system, against which a score can be interpreted. Ultimately, this is a question of construct validity. A benchmark score is at best a measurement relative to a dataset and a learning problem. Any inference from it to a capability rests on assumptions that must be made explicit [6].

3. Cognitive Complexity Frameworks

Governing an evaluation requires a statement of what its task demands. We provide one by characterizing each operation along three dimensions drawn from established accounts of human cognition. These dimensions and their possible values are presented in Table 1. Each dimension isolates a different facet of demand and together they form a compact profile that describes a task independently of any system that might attempt it.

The first dimension originates in *Bloom’s taxonomy* of educational objectives [7]. It was originally developed to classify goals based on the required level of cognitive complexity for completion. It is developed based on behavioral observations of learning processes and classifies various cognitive processes from a simple recall to higher-level reasoning and creative generation. Almost 50 years later, a revision was made in order to adapt to modern views of cognitive psychology [8]. This revision introduces a new, complementary dimension – *Knowledge Dimension (KD)* – that separates what is known from what is done with it, by distinguishing between types of knowledge required for different tasks. Finally, *Relational Complexity Theory (RCT)* [9] is inspired by the notion of relational arity from formal systems in logic and computer science, including relational database theory. It models task difficulty as a function of the number of simultaneous relations that need to be processed. The three are orthogonal, meaning that an operation can be procedurally routine yet relationally heavy, or conceptually rich yet lightly coordinated. A profile therefore separates facets of demand that a single accuracy figure necessarily collapses.

We assign a profile to an operation a priori, from its definition. For each operation we identify the characteristic act that distinguishes it from the others and read each dimension off that act. Please note that while we adopt terminology from cognitive science, we do not claim that the tested systems engage in these processes in a human sense. We assess the extent to which their outputs reflect behavior consistent with such operations.

The unit of analysis is the task, rather than the solver, so one profile applies whether the operation is carried out by an OWL reasoner, an embedding model, a question-answering system or a language

Table 1

The characterization vocabulary with three dimensions of task demand and their values.

<i>Cognitive process</i> (Bloom [7])	
Remember	Recalling facts or definitions without further processing.
Understand	Interpreting given information, structures, or queries without transforming them or generating new representations.
Apply	Correctly executing a known procedure or pattern, such as retrieval or following syntactic rules.
Analyze	Recognizing or decomposing relationships between data, especially when multiple elements or steps must be coordinated.
Evaluate	Judging the correctness, relevance, or quality of a result.
Create	Generating new content, such as queries or data structures.
<i>Knowledge dimension</i> (Anderson and Krathwohl [8])	
Factual	Recalling or recognizing specific terminology, syntax, or concrete information.
Conceptual	Structural or relational understanding, such as schema structure, data models, or logical organization.
Procedural	Correctly applying known methods, routines, or transformation steps.
Metacognitive	Awareness and control over one’s own strategies, such as selecting approaches, planning, or monitoring correctness.
<i>Relational complexity</i> (Halford et al. [9])	
Low	Interpreting or manipulating individual binary relations or isolated, simple structures with minimal dependencies.
Medium	Processing multiple relations and entities simultaneously, such as coordinating several triples.
High	Considering multiple interrelated entities or nested dependencies simultaneously, often requiring abstract or hierarchical reasoning.

model. That makes this characterization system-agnostic. Because the profile is fixed prior to and independently of performance, it supplies an explicit account of what a task requires, against which a score becomes interpretable.

The closest approach is ADeLe [10], which places task instances on general ability-demand scales to explain what benchmarks measure and to predict system performance. It likewise finds that benchmarks do not always measure what they claim. It differs in a few ways. First, the dimensions here are few and interpretable, drawn from widely used accounts of cognitive demand, which keeps a profile legible to the non-specialists who must act on it. Second, the characterization takes the task as its unit and is deliberately solver-independent. One profile applies to an OWL reasoner, an embedding model or a language model alike. ADeLe’s scales are instead calibrated to system abilities and validated by predicting performance. That is a strength when the goal is forecasting how a given system will score, but one that is tied to a class of solver, which a system-agnostic characterization cannot assume. Third, the purpose here is explanatory and audit-oriented rather than predictive, though prediction is not precluded. While we take change benchmarks as a focus point here, the framework is not specific to them. Prior work applied it to querying and interaction tasks [11, 12].

4. Illustrative Characterization

In order to illustrate how task characterization works within our framework, we take into consideration two benchmarking settings: text-driven knowledge graph updating, exemplified by EMERGE [2] and temporal knowledge graph forecasting, exemplified by the ICEWS [13] and GDELT [14] benchmarks as standardized in TGB 2.0 [3]. In each, recall-oriented scalars are used to score a system’s capability with a product that quantifies the overlap between a predicted triple and a reference one. However, the operations that are being evaluated are distinct processes that each happen to emit facts. By construction, product-match metrics are unable to separate such processes, so disaggregating the score by operation type does not supply an account of what any operation demands. That is what a characterization layer

adds, and it can be stated before any system is run.

The change operations these benchmarks evaluate are instances of a small, long-studied family. Belief-revision theory [15] distinguishes expansion (incorporating new, consistent information), contraction (relinquishing information) and revision (accommodating information that conflicts with what is currently held). Temporal settings, on the other hand, add the gradual change in a fact’s relevance over time. The operations can be located in this family, giving each one of them an identity independent of any particular benchmark.

4.1. Text-driven Knowledge Graph Updating

EMERGE [2] is a benchmark for text-driven knowledge graph updating (TKGU). Over a KG snapshot at a time t , $G_t = (V_t, R_t, T_t)$, where V_t is a set of entities, R_t is a set of relation types and T_t is a set of triples, and an incoming textual passage, EMERGE defines five update operations: *Exists* – confirm that a stored triple is supported by the text; *Add* – insert a triple between two entities already in G_t ; *Mint+Add* – introduce a new entity and then attach a triple to it; *Infer* – add a triple linking an emerging entity to the graph where the relation is not stated in the text; and *Deprecate* – invalidate a stored triple that the new evidence supersedes. Each operation is scored independently using Completeness [16] and a recall-oriented G-BERTScore [17].

Table 2

Characterization of EMERGE’s update operations [2] along the three dimensions, derived a priori from the operation definitions. All five are reported on the same scales (Completeness, G-BERTScore-R).

Task	Bloom	Knowledge Dimension	Relational Complexity
Exists	Remember, Understand	Factual	Low
Add	Apply	Factual, Procedural	Low
Mint + Add	Apply	Factual, Procedural	Low
Deprecate	Analyze	Conceptual, Procedural	Medium
Infer	Analyze, Create	Conceptual, Procedural	High

Table 2 characterizes each operation along the three dimensions. The annotations are derived a priori from the operation definitions. *Exists* demands the least: a stored fact is recognized as present and the passage is interpreted to confirm it is entailed (Bloom: Remember, Understand), with no change to the graph. The unit is one concrete fact (KD: Factual), checked as a single binary relation against one passage (RC: Low). *Add* and *Mint+Add* are both placement operations. A relation stated in the text is recognized (KD: Factual) and inserted by a defined routine (Bloom: Apply, Procedural). They differ only in a procedural sub-step. *Mint+Add* first creates the new node before attaching the edge, without additional relational load, since each still coordinates a single binary relation between one subject and one object (RC: Low). The fact that two operations the benchmark separates collapse to a single demand profile informs us that the distinction *Mint+Add* draws is procedural and not a difference in what the task fundamentally requires.

The demand rises at the two operations that must relate new information to existing graph structure. *Deprecate* turns on recognizing that the incoming evidence supersedes a stored triple by relating new information to an existing belief (Bloom: Analyze). Recognizing supersession requires schema-level understanding of what excludes what (KD: Conceptual, Procedural) and it coordinates the stored triple, the incoming evidence and the relation between them (RC: Medium). *Infer* is the most demanding operation EMERGE scores. It produces a triple present in neither the text nor the graph, deriving an unstated link and thereby originating new structure (Bloom: Analyze, Create). The derivation rests on ontological category structure (KD: Conceptual, Procedural) and must hold the emerging entity, its inferred type, the target entity and the licensing schema constraints together (RC: High).

The characterization shows that the operations span recognition to inference, i.e. from Low-RC retrieval to High-RC derivation, so a strong performance on *Add* does not grant any inference about *Infer*. We can easily see the switch from the three placement operations that are Factual to Conceptual

for *Deprecate* and *Infer*, where reasoning about graph structure becomes necessary. A recall-oriented score renders that boundary invisible.

4.2. Temporal Forecasting

A Temporal Knowledge Graph is a set of quadruples (s, r, o, t) up to time t , with entities s, o , relation r and time stamp t [3, 18]. Temporal KG forecasting asks a system to predict future facts. Given a temporal KG, it ranks candidate entities completing a query such as $(s, r, ?, t+1)$, namely entity forecasting, also termed future link prediction. Long-standing benchmarks such as ICEWS14 [13] and GDELT [14], consolidated and re-standardized by TGB 2.0 [3], score systems by Mean Reciprocal Rank (MRR) and Hits@ k .

Forecasting is defined as predicting facts at a future timestep conditioned on the temporal history of the graph, so performing the task as defined requires coordinating an entity’s interaction history across many timestamps, weighting recency and recurring patterns to anticipate the next fact. Decomposing those temporal relationships among multiple interacting entities is an Analyze-level process over Conceptual and Procedural knowledge. Because it requires holding many interrelated facts and their ordering simultaneously, it sits at High relational complexity. This is the demand profile describing a system genuinely modeling temporal evolution would exhibit. A process that ranks candidates without conditioning on temporal history does not perform the task as defined, however high it scores.

The temporal case differs, however, from EMERGE because it specifies only an input/output relationship. In contrast, EMERGE specifies operations like *deprecate* one triple or *infer* a link. The I/O relationship fixes what goes in and what comes out, but not the operation that must connect them. A demand profile can be stated for the intended solution, yet nothing in the benchmark compels a system to realize it. That has been also empirically shown. A non-learnable co-occurrence heuristic that ignores temporal order entirely, but merely counts how often entity-relation pairs have co-occurred, comes within roughly a quarter of the state-of-the-art on average, and closer still on individual benchmarks such as GDELT. This artifact holds across multiple benchmarks, including ICEWS, GDELT, YAGO and Wikidata [5]. That work traces the cause to co-occurrence biases baked into the datasets and an evaluation formulation that lets those biases be exploited. An independent recurrency baseline predicting facts that have recurred in an entity’s past tells the same story from the other direction. Compared against eleven methods on five datasets it ranks among the top for three out of five datasets, which its authors present precisely as a way to separate actual from fictitious progress in the field [19]. TGB 2.0’s own maintainers likewise report that simple heuristic baselines are often competitive with elaborate models [3]. The pattern is not confined to the established benchmarks. On a newly constructed temporal KG of global conflicts and trade, the same recurrency baseline again outscores every learned model tested, while even static, non-temporal baselines rank highly owing to the frequent recurrence of facts [18]. Counting co-occurrences or recurrences is the execution of a fixed procedure (Bloom: Apply) over concrete frequency statistics (KD: Factual), using isolated counts with no coordination of interrelated entities or temporal dependencies (RC: Low). These shortcuts were detected by prior work [5, 18, 19]. Our contribution is a single, system-agnostic vocabulary in which the gap they exploit can be named a priori and a benchmark’s susceptibility to such a gap can be anticipated. A summary of the characterization of both the intended and shortcut temporal modeling is presented in Table 3.

Table 3
Characterization of temporal modeling as intended and its shortcut.

Task	Bloom	Knowledge Dimension	Relational Complexity
Temporal modeling (intended)	Analyze	Conceptual, Procedural	High
Co-occurrence shortcut	Apply	Factual	Low

The consequence is that the metric fails to order systems by the reasoning they engage. A Low-RC counting process is scored as equal to, or even better than, a system performing genuine temporal modeling, so a high MRR cannot certify that the High-RC reasoning the task nominally requires was

performed at all [20].

Across both settings the characterization measures the demand the operation imposes. Stated before any system runs, it gives an auditor the expectation needed to ask whether a score reflects a genuine solution or a shortcut beneath it. With that it turns the later obtained score number into a capability claim that an external party can examine and verify.

5. Implications for Governance

The characterization does not replace accuracy. It rather aims to make it interpretable while addressing each governance failure identified in Section 2. Because a demand profile is fixed independently of any system and states what an operation requires, it supplies the vocabulary that lets an external party set and verify capability requirements.

First, evaluation can be made auditable at the level of demand. Instead of a single figure, a system's results can be read against the profiles of the operations it was tested on. For example, do failures concentrate in high-relational-complexity or in Analyze- or Create-level operations, while low-demand placement operations succeed. Localizing successes and failures in demand space lets an auditor require evidence at a stated level of demand rather than accept an average that conceals where competence ends. In fact, it does so in the stage-wise spirit that internal-audit frameworks prescribe [1]. Failure on a low-demand operation and failure on a high-demand one are different findings and only a demand-resolved report tells them apart.

Second, gameability becomes expressible as a testable hypothesis. The profile states the demand a genuine solution must meet, setting an expectation against which observed behavior can be checked. When a score attainable only by a high-relational-complexity operation is in fact reached by a process of far lower demand, like the co-occurrence count matches in temporal forecasting systems (Section 4), the divergence between nominal and exhibited demand points to a surface shortcut. This now becomes a testable hypothesis, meaning that an evaluator can ask whether performance survives once the exploitable regularity is removed and read a system's demand profile as a claim to be falsified rather than assumed.

Third, the analysis yields a general criterion for whether a benchmark can be governed at all. A benchmark that fixes only an I/O contract constrains what a system receives and returns, but not the operation that connects them. Therefore it cannot enforce any demand profile, whatever the solver. In these benchmarks, time enters only as an ordinal index on each quadruple, recording when a fact holds. However, it does not explain what its temporal extent or expiry mean. A representation that carries no temporal semantics cannot demand temporal reasoning of a system. Additionally, to be governable, an evaluation must either constrain the operation itself or pair its I/O contract with a demand characterization and tests that discriminate between demand levels. Concretely, such tests isolate the demand in question. For temporal forecasting, a benchmark could report performance separately on recurring and non-recurring queries, so that a purely recurrence-based process cannot succeed and only a system using temporal structure can. The gap between the two subsets then measures how much of a headline score rests on the low-demand shortcut. Governability requires that the demand a benchmark targets be enforceable.

Finally, the same profiles inform deployment in regulated settings. A demand-resolved profile lets a decision-maker match demonstrated capability to the demand profile of the target application and set thresholds tied to specific capabilities rather than to an aggregate whose meaning is unspecified. A compliance regime can then require evidence of competence at the demand levels a use case actually involves and an auditor can confirm that the evidence exists. Ultimately the process is the same: the characterization converts an aggregate score into a capability claim stated in the vocabulary of task demand, making it interpretable.

6. Limitations and Future Work

The characterization has limitations, each of which marks a direction for future work. The demand profiles are assigned a priori, by expert reading of an operation’s definition. This keeps the assignment transparent and open to inspection, but it is, for now, a matter of judgment. The next step is to explicitly and clearly define the annotation procedure in order for it to be reliable across annotators and validate the assigned profiles against independent measures of difficulty. Second, the characterization needs to be applied across a wider landscape of knowledge graph benchmarks. Another direction this work opens is the temporal representation itself. Where time is modelled as a bare index, the characterization can state that a task technically demands temporal reasoning, but, the representation does not allow for it. Richer encodings of time, such as validity intervals, durations or versioned change may be needed before a benchmark can genuinely require temporal reasoning. Characterizing how a temporal representation’s expressivity bears on task demand is then a natural extension of the framework. That leads to the final point that the three dimensions are not presented as a closed set. They capture facets of demand that a single score conflates and are sufficient for the characterizations given here. However, as the framework is applied to new task types, further dimensions may prove useful. We treat this as an opening and extensions can be motivated as the need arises. Taken together these mark the path from a characterization as an explanatory instrument to a validated basis for governable evaluation.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

- [1] I. D. Raji, A. Smart, R. N. White, M. Mitchell, T. Gebru, B. Hutchinson, et al., Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAccT ’20), 2020, pp. 33–44.
- [2] K. Zaporjets, D. Daza, E. Barba, I. Assent, R. Navigli, P. Groth, EMERGE: A benchmark for updating knowledge graphs with emerging textual knowledge, 2025. [arXiv:2507.03617](#).
- [3] J. Gastinger, S. Huang, M. Galkin, E. Loghmani, A. Parviz, F. Poursafaei, et al., TGB 2.0: A benchmark for learning on temporal knowledge graphs and heterogeneous graphs, in: Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024), Datasets and Benchmarks Track, 2024.
- [4] B. Parsia, N. Matentzoglou, R. S. Gonçalves, B. Glimm, A. Steigmiller, The OWL reasoner evaluation (ORE) 2015 competition report, *Journal of Automated Reasoning* 59 (2017) 455–482.
- [5] J. Zhang, Z. Li, M. Wang, J. Shao, J. Cui, H. Li, Towards better evolution modeling for temporal knowledge graphs, 2026. [doi:10.48550/arXiv.2602.08353](#). [arXiv:2602.08353](#).
- [6] T. Freiesleben, S. Zezulka, The benchmarking epistemology: Construct validity for evaluating machine learning models, 2025. [arXiv:2510.23191](#).
- [7] B. S. Bloom, Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook 1: Cognitive Domain, Longman, New York, 1956.
- [8] L. W. Anderson, D. R. Krathwohl, A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives, Addison Wesley Longman, New York, 2001.
- [9] G. S. Halford, W. H. Wilson, S. Phillips, Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology, *Behavioral and Brain Sciences* 21 (1998) 803–831.
- [10] L. Zhou, L. Pacchiardi, F. Martínez-Plumed, K. M. Collins, Y. Moros-Daval, S. Zhang, et al., General scales unlock AI evaluation with explanatory and predictive power, *Nature* 652 (2026) 58–67.
- [11] S. Todorovikj, L.-P. Meyer, M. Martin, Characterizing knowledge graph tasks in LLM benchmarks

- using cognitive complexity frameworks, in: Posters and Demo Track of the International Conference on Semantic Systems (SEMANTiCS 2025), volume 4064 of *CEUR Workshop Proceedings*, 2025. URL: <https://ceur-ws.org/Vol-4064/PD-paper5.pdf>.
- [12] L.-P. Meyer, J. Frey, F. Brei, D. Heim, S. Gründer-Fahrer, S. Todorovikj, et al., Evaluating large language models for RDF knowledge graph related tasks: The LLM-KG-Bench framework 3, *Semantic Web* (2026). In press.
 - [13] E. Boschee, J. Lautenschlager, S. O’Brien, S. Shellman, J. Starz, M. Ward, ICEWS coded event data, Harvard Dataverse, 2015.
 - [14] K. Leetaru, P. A. Schrod, GDELT: Global data on events, location, and tone, 1979–2012, in: *Proceedings of the International Studies Association (ISA) Annual Convention*, 2013, pp. 1–49.
 - [15] C. E. Alchourrón, P. Gärdenfors, D. Makinson, On the logic of theory change: Partial meet contraction and revision functions, *The Journal of Symbolic Logic* 50 (1985) 510–530.
 - [16] P. Jiang, J. Lin, Z. Wang, J. Sun, J. Han, GenRES: Rethinking evaluation for generative relation extraction in the era of large language models, in: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2024)*, Volume 1: Long Papers, 2024, pp. 2820–2837. doi:10.18653/v1/2024.naacl-long.155.
 - [17] S. Saha, P. Yadav, L. Bauer, M. Bansal, ExplaGraphs: An explanation graph generation task for structured commonsense reasoning, in: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, 2021, pp. 7716–7740.
 - [18] J. Gastinger, T. Szttyler, N. Steinert, S. Gründer-Fahrer, M. Martin, A. Schuelke, et al., Dynamic representations of global crises: A temporal knowledge graph for conflicts, trade and value networks, in: *Proceedings of the Third Learning on Graphs Conference (LoG 2024)*, volume 269 of *Proceedings of Machine Learning Research*, 2024.
 - [19] J. Gastinger, C. Meilicke, F. Errica, T. Szttyler, A. Schuelke, H. Stuckenschmidt, History repeats itself: A baseline for temporal knowledge graph forecasting, in: *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*, 2024, pp. 4016–4024.
 - [20] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, et al., Shortcut learning in deep neural networks, *Nature Machine Intelligence* 2 (2020) 665–673.