Report on the International Workshop on Transforming Graph Data (TGD@EDBT/ICDT 2025-2026)

Anna Bernasconi
Politecnico di Milano
anna.bernasconi@polimi.it

Stefania Dumbrava
SAMOVAR & ENSIIE
stefania.dumbrava@ensiie.fr

Riccardo Tommasini
LIRIS & INSA Lyon
riccardo.tommasini@insa-lyon.fr

ABSTRACT

The International Workshop on Transforming Graph Data (TGD) ran for two editions, co-located with EDBT/ICDT in Barcelona in March 2025 and in Tampere in March 2026. It was motivated by an asymmetry in the graph data management ecosystem. Storage, indexing, and querying have matured considerably, while the mechanisms needed to evolve, integrate and reason over graph data remain comparatively ad hoc. The program comprised 12 papers, four keynotes and one panel, spanning declarative and logic-based transformation languages, schema evolution and discovery, interoperability between property graphs and RDF, hybrid graph and time-series models, and transformation-aware summarization and lineage. This report summarizes the contributions, the recurring themes, and the agenda that emerged.

1. INTRODUCTION

Graph models are a standard means of capturing relationships in modern data [2], from social networks to knowledge graphs to legislative systems, and graph database systems have advanced substantially in storage, indexing and query processing. The mechanisms by which graph data is *transformed* have received comparatively little attention. Transformation tasks are frequently addressed with custom pipelines rather than declarative specifications, and they often lack a formal semantics. Temporal properties such as recency, timeliness, and validity are increasingly relevant, and interest in combining symbolic and neural methods is growing.

The TGD workshop was established to address this gap. The first edition was held on March 25, 2025 in Barcelona, Spain, and the second on March 24, 2026 in Tampere, Finland, both full-day events colocated with the 28th and 29th EDBT/ICDT Joint Conference, with proceedings in CEUR-WS Vol-3946 and Vol-4192. The two editions accepted 12 papers, five in 2025 and seven in 2026. Two keynotes were delivered per edition, the first edition closed with a panel, and the workshop drew roughly 30 to 50 participants. Both editions are documented at `https://transforming-graph-data.github.io/`, where the program committees are listed in full.

Section 2 organizes the papers around three recurring themes. Section 3 summarizes the keynotes. Section 4 reports the panel and the agenda it set.

2. KEY TOPICS

We present papers from both workshop editions below, grouped into three main categories.

2.1 Graph Transformation Applications

Colombo et al. [6] considered legislative knowledge management with property graphs, a domain in which transformations are applied implicitly to manage law versioning and temporal retrieval, and so to monitor evolving legal systems.

Dachselt et al. [7] addressed explainability for Datalog with stratified aggregation, given its growing use as a graph transformation language. Their *summary proof tree* merges structurally similar proof trees into one, labeling nodes by sets of inferences so that derivations stay inspectable at graph scale. From a set of facts, one query returns the largest proof structure they share. From a proof pattern, the other returns every fact derived that way, by compiling the pattern into Datalog rules run in Nemo. On standard benchmarks most queries return within a second, while a baseline built on Nemo's tracing mostly times out.

Gu and Kim [10] introduced ScaDANN, a disk-based graph indexing method for approximate nearest neighbor search under constrained memory. Vectors are partitioned into grid blocks, and construction is interleaved with merging through overlapping block-level insertion, producing a unified index. Against DiskANN, ScaDANN achieves 1.3 to 1.5 times faster construction, 1.2 to 1.4 times higher throughput at comparable recall, and a single index can be built over a dataset up to ten times larger under the same memory budget.

2.2 Temporality in Graph Transformations

Ammar et al. [1] argued for extending the property graph model to treat time series as first-class citizens, observing that time-series data in graph applications is often stored separately, either in side systems or as attributes in graph databases. HyGraph responds with a unified model where vertices and edges may be of property-graph or time-series kind, in which subgraph membership is itself a time series, and with two transformation operators over it. The first ingests time series and produces a graph capturing relationships among them based on correlation values. The second generates a time series from the evolution of temporal graph metrics, and can augment the hybrid structure with it for self-enrichment. A micro-mobility case study on bike-sharing data applies both transformation operators to identify rebalancing partners among stations.

Skvortsov et al. [14] approached transformation through Logica, an open-source Datalog variant with aggregation that compiles to SQL and runs on parallel engines such as DuckDB and BigQuery. The paper argues by construction that a general-purpose logic language with the right compilation target already handles what classical graph transformation approaches find difficult, notably temporal transformations and scalability. Examples include message passing, transitive reduction, graph condensation, and a well-founded formulation of the Win-Move game. Most relevant here is time-aware pathfinding, where edges carry the interval over which they exist and earliest arrival times follow from a very succinct program. A taxonomic tree over a Wikidata dump of 806M facts across 89M objects, 13 GB in DuckDB, was computed in under seven seconds on one cloud instance with no custom indexing.

Budi and Amagasa [5] treated adversarial perturbation of discrete-time dynamic graphs explicitly as graph transformation, in which modifications to edge sets across snapshots propagate through a sequential architecture. Their TIDGA framework weights perturbations by a temporal position term, on the intuition that early perturbations have more subsequent computation to influence, and by an edge persistence factor measured through Jaccard similarity between consecutive edge sets. The weighting helps only if the victim model updates its parameters across snapshots and gradient magnitudes are spread rather than concentrated. Otherwise, an untargeted baseline does at least as well.

Both 2025 keynotes approached the aspect of temporality through learned components. Once a model is trained over a graph that continues to evolve, model freshness becomes a concern in its own right.

2.3 Graph Data Interoperability

Elhalawati et al. [9] addressed the mapping of property graphs to RDF through a flexible RML-based mechanism. The structural and semantic mismatch between the two models is the difficulty, and a declarative mapping language is offered as the means of keeping transformations consistent and comparable across approaches, and RML proves more expressive than the approaches it captures. Marketakis and Tzitzikas [13] set out a vision for schema mappings assisted by large language models, in which heterogeneous sources become RDF knowledge graphs aligned with shared ontologies. Explicit mappings are retained as the artifact that keeps the transformation scalable and explainable, in contrast to a single opaque step from prompt to RDF. Dutkiewicz et al. [8] framed lineage discovery as a knowledge graph link prediction problem, using the transformation into a knowledge graph to unify database semantics and reason over dependencies across sources.

Hausler and Klettke [11] addressed schema evolution, which derives a target schema from a source schema and a sequence of operations, from schema transformation, which derives the operations from a source and target schema pair. The mechanism is the Graph Data Profile, which makes heterogeneity explicit by extracting statistical metadata together with uniqueness, existence and type constraints, missing edges and valid inclusion dependencies. A profile diff previews the effect of an operation on heterogeneity, while inclusion dependencies detect renames, moves, and the redundancy left by copies. Costs combine a formalization of each operation with metrics drawn from the profiles, so that a user sees the cost and the structural consequence of a change before committing. Limnaios et al. [12] presented Quantum-PG-HIVE, a schema discovery tool for property graphs based on quantum computing.

Zervos et al. [16] presented GSS, which summarizes graphs through answer-centered explanatory subgraphs, organized around what explains an answer rather than around frequency.

3. KEYNOTE HIGHLIGHTS

The 2025 keynotes, reported by the organizers, were by Juan Reutter (Pontificia Universidad Católica de Chile) and Yuanyuan Tian (Microsoft Gray Systems Lab). The 2026 ones, with abstracts in the proceedings, were by Angela Bonifati (Lyon 1, CNRS LIRIS, IUF) and Efthymia Tsamoura (Huawei Labs).

3.1 Transformations in Graph Learning

Prof. Reutter presented a neuro-symbolic approach to query answering that combines graph structure

with natural language input, so that a query can be interpreted against information inferred from learned patterns and not only against the current state of a graph. The difficulty lies in linking entities mentioned in the graph to entities mentioned in natural language. The talk spanned fully neural approaches, which train graph neural networks to answer queries directly from materialized data and symbolic constraints, and hybrid models that embed graphs and queries while permitting soft logical reasoning. Graph neural networks generalize poorly over unseen relations, so relations must be contextually encoded, often through a metagraph capturing the essential semantics. The talk also set out constraints on these systems. Queries should avoid cycles, stay contextually relevant, and avoid excessive generality.

3.2 Graph Databases in Industry

Dr. Tian took an industrial view and opened on a practical obstacle, namely that customer data rarely arrives in graph form, and that building usable graph models from heterogeneous and often tabular sources is labor-intensive and error-prone. She asked whether large language models can serve as general-purpose assistants for graph modeling, easing the path from tables to entity-relationship models and then to graphs, and raised self-evolving graphs, networks that change over time by integrating new data and relationships, particularly in the life sciences. The talk was explicit about the limits. Natural language remains ambiguous; perfect matches are neither always possible nor always necessary, and expert knowledge has to stay in the data management lifecycle. She closed with thoughts on the distance between academic research and industrial practice, noting the absence of surveys capturing graph technologies from an industrial standpoint and calling for closer alignment on usability and scalability.

3.3 Graph Transformations in Action

Prof. Bonifati [4] examined property graphs as a unifying model for data management and analytics, in which multi-labeled nodes and edges carrying key-value properties are expressive enough to represent data from diverse sources. The central theme was the emergence of declarative paradigms for property graph transformation designed for efficiency and scalability, which let users specify what a transformation should achieve rather than how it should execute. The talk connected this to the ISO/IEC standardization effort behind GQL and SQL/PGQ, and to the schema and constraint languages emerging as part of future data definition languages. Beyond integration and cleaning, it extended the same machinery to causal inference and path-based causal analysis.

3.4 Trigger Graphs

Dr. Tsamoura [15] challenged the belief that symbolic reasoning does not scale, introducing trigger graphs, which support exact Datalog reasoning in the order of seconds over stores with billions of edges and so reframe scalability as a practical objective rather than an inherent limit. Trigger graphs organize the reasoning itself as a graph computation, which lets them avoid the redundant work characteristic of most commercial and open-source engines and support probabilistic reasoning more efficiently than approximate techniques. The talk then introduced equivalence semantics for probabilistic logic programming. Where the standard approach induces a distribution over subsets of probabilistic facts, equivalence semantics induces a distribution over the possible equivalence relations between symbols, and was reported to improve on state-of-the-art neurosymbolic techniques for link prediction, rule mining and symbolic grounding by up to 42%.

4. LOOKING TO THE FUTURE

The 2025 panel, moderated by Riccardo Tommasini, brought together the two keynote speakers with Angela Bonifati, Leonid Libkin and Stefania Dumbrava, who opened with a position statement.

Angela Bonifati advocated for human-in-the-loop methods as a way to reconcile expressiveness with tractability, a central trade-off in graph transformations. Leonid Libkin argued for compositionality of graph query languages and noted that GQL and SQL/PGQ do not yet support it, a gap addressed in subsequent work [3]. Stefania Dumbrava argued for a composable graph transformation ecosystem whose reliability rests on a small trust kernel with a formal semantics, rather than on agreed APIs.

The panel returned to the absence of formalism in transformation languages, interoperability across graph data models, and the still-underexplored role of generative AI. It converged on the view that graph construction should start from user needs and proceed iteratively, rather than follow a fixed pipeline from tables to an entity-relationship model to a graph. Enabling end users, whether researchers, analysts or domain experts, to participate meaningfully in graph modeling should thus be a precondition for adoption, and not only a usability concern. This position complements the industrial keynote of Section 3.2, which proposed large language models as assistants, leaving open whether generative AI

should accelerate this path or enable a new one.

4.1 Perspectives

Languages. The program offered two answers to what a graph transformation language should be. One reuses a general-purpose declarative language and invests in the compilation target, as in the Logica work [14]. The other one designs paradigms specific to property graphs, as in Section 3.3 and in the schema and constraint languages emerging alongside GQL and SQL/PGQ. The panel's point about composability bears on the choice, since a host language confers composability on the formalism embedded in it, while a dedicated language must establish it independently. A third strand concerns what a language owes its user once the program runs, treated as part of language design, through explainability of derivations [7] and cost transparency [11].

Systems. Exact Datalog reasoning now runs in seconds over stores with billions of edges [15], a recursive query over an 806M-fact Wikidata dump completed in under seven seconds on a single cloud instance [14], and billion-scale index construction fits inside a fixed memory budget [10]. Indexing structures accommodating both topological and temporal changes remain open, as no query language natively integrates time-series operations with graph traversals. Both are prerequisites for the unified treatment of time series and graph data that HyGraph [1] envisions.

Applications. Graph transformation supports the integration and evolution of complex, interconnected data. In biomedicine, it allows genomic and clinical sources to be combined for drug discovery and disease analysis. It matters as well in law, mobility and social networks, where evolving data, interoperability and analytics meet, increasingly in combination with machine learning. Reliability and efficiency, the panel's closing concern, have to hold across all of them at once. Engaging these communities remains part of the workshop's purpose.

Acknowledgements

The first edition was chaired by Stefania Dumbrava and Riccardo Tommasini, joined by Anna Bernasconi for the second. We thank the program committees, the participants, and the EDBT/ICDT workshop chairs. Both editions were supported by ANR (ANR-24-CE25-1109), and the second also by the UNICA project, under the EU4Health Programme (No. 101218966).

References

[1] Mouna Ammar et al. 2025. Transforming Time Series into Graphs and Vice Versa with HyGraph. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[2] Renzo Angles et al. 2017. Foundations of Modern Query Languages for Graph Databases. *ACM Comput. Surv.* 50, 5 (2017), 68:1–68:40.

[3] Marcelo Arenas, Leonid Libkin, and Wim Martens. 2026. A Compositional Language for Property Graphs. *CoRR* abs/2606.23399 (2026).

[4] Angela Bonifati. 2026. Property Graph Transformations in Action: From Data Integration to Causal Analysis. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[5] Brilian Surya Budi and Toshiyuki Amagasa. 2026. Exploiting Temporal Importance for Adversarial Attacks on Dynamic Graphs. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[6] Andrea Colombo et al. 2025. Legislative Knowledge Management with Property Graphs. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[7] Raimund Dachselt et al. 2026. Declarative Debugging for Datalog with Aggregation. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[8] Jakub Dutkiewicz et al. 2026. Data Lineage Discovery in Databases based on Knowledge Graph Link Prediction. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[9] Ali Elhalawati et al. 2025. Flexible RML-Based Mapping of Property Graphs to RDF. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[10] Hyein Gu and Min-soo Kim. 2025. ScaDANN: A Scalable Disk-based Graph Indexing Method for ANN. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[11] Dominique Hausler and Meike Klettke. 2026. Heterogeneity Detection and Comparison for Cost-Aware Graph Schema Evolution and Transformation. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[12] Emmanouil Limnaios et al. 2026. Quantum-PG-HIVE: Schema Discovery for Property Graphs using Quantum Computing. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[13] Yannis Marketakis and Yannis Tzitzikas. 2026. Beyond Prompt-to-RDF: A Vision for Scalable and Explainable Graph Transformations via LLM-assisted Schema Mappings. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[14] Evgeny S. Skvortsov et al. 2025. Logica-TGD: Transforming Graph Databases Logically. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[15] Efthymia Tsamoura. 2026. Trigger Graphs and Probabilistic Equivalence: Towards Scalable and Efficient Inference and Neurosymbolic Learning. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.

[16] Valantis Zervos et al. 2026. GSS: Graph Semantic Summarization using answer-centered explanatory subgraphs. In *EDBT/ICDT Workshops (CEUR WS Proc.)*.