

A Multi-Scale Process for Mining and Abstracting Environmental Spatio-Temporal Graphs

Assaad Zeghina ✉ 

Université de Strasbourg, CNRS, ENGEES, ICube UMR 7357, Strasbourg, France

Aurélié Leborgne ✉ 

Université de Strasbourg, CNRS, ENGEES, ICube UMR 7357, Strasbourg, France

Florence Le Ber ✉ 

Université de Strasbourg, CNRS, ENGEES, ICube UMR 7357, Strasbourg, France

Antoine Vacavant ✉ 

Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, France

Abstract

Understanding the continuous evolution of environmental systems increasingly relies on tracking large-scale, heterogeneous data across shifting timelines, commonly structured as temporal multigraphs. While such representations are highly expressive, their temporal density and structural heterogeneity make the automated analysis of long-term transitions particularly challenging. In this paper, we propose a unified process for extracting and abstracting recurrent evolutionary patterns within time-varying multigraphs. The proposed pipeline articulates four successive stages: (1) a spatio-temporal modeling stage based on RCC-8 relations to capture both topological structures and the chronological evolution of geographic entities across consecutive time steps; (2) a frequent pattern mining stage designed to discover recurring temporal sequences within multigraph representations; (3) a pattern matching stage that enables efficient, multi-scale localization of these evolutionary signatures; and (4) a hierarchical abstraction stage relying on super-nodes to produce a compact, condensed representation of the graph's historical progression. An interactive visual environment is additionally provided as a complementary exploration tool to inspect the timeline outputs of the pipeline. We report on the results obtained from historical graph analyses of CORINE Land Cover data across multiple European cities, demonstrating the process's capacity to reveal recurrent environmental behaviors and meaningful, time-dependent trajectories of change.

2012 ACM Subject Classification Information systems → Data mining; Computing methodologies → Neural networks; Information systems → Geographic information systems

Keywords and phrases Graph Deep Learning, Temporal Graphs, Pattern Mining, Temporal Analysis, Graph Abstraction, Subgraph Matching

Digital Object Identifier 10.4230/OASICS.TIME.2026.7

1 Introduction

Environmental monitoring has progressively evolved from isolated sensor measurements toward continuous spatio-temporal observation systems that capture dynamic environmental processes over time [14]. With the increasing availability of time-dependent, heterogeneous, and multidimensional data streams, the main challenge is no longer limited to data collection, but rather to understanding temporal evolutions, recurring patterns, and long-term dependencies within these complex datasets. Temporal (or spatio-temporal) graphs provide an expressive modeling framework capable of capturing temporal dynamics alongside multiple interconnected dimensions simultaneously. By representing entities as nodes and encoding temporal, spatial, and semantic relationships as edges, they offer a unified view of evolving environmental processes [30]. In particular, temporal intervals enable the modeling of the evolution of relationships over time, while the use of Region Connection Calculus (RCC-8) [20]



© Assaad Zeghina, Aurélié Leborgne, Florence Le Ber, and Antoine Vacavant;
licensed under Creative Commons License CC-BY 4.0

33rd International Symposium on Temporal Representation and Reasoning (TIME 2026).

Editors: AndreA Orlandini and Sophie Pinchinat; Article No. 7; pp. 7:1–7:15

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

provides a qualitative description of spatial topology. However, despite their expressive power, temporal graphs often suffer from structural complexity that hinders both algorithmic analysis and human interpretation.

Frequent pattern mining, which aims to identify recurring subgraph structures, has emerged as a powerful approach for uncovering meaningful relational configurations in complex networks [31]. Such patterns often reveal fundamental interaction schemes and provide insights into the underlying organization of the data [11]. However, traditional pattern mining approaches face severe computational limitations, as the number of candidate subgraphs grows exponentially, making subgraph enumeration and matching NP-hard problems. These challenges are further amplified in spatio-temporal graph settings, where multiple edge types and overlapping relations must be considered [18]. Consequently, recent advances increasingly advocate the use of neural-based approaches for frequent pattern mining, as they offer scalable alternatives capable of learning expressive representations of complex graph structures, including spatio-temporal and multigraph settings [26, 27]. In particular, approaches such as SPMiner or Multi-SPMiner exploit learned projection spaces and graph representation learning to guide the discovery of frequent graph patterns and spatio-temporal graphs respectively [29].

In this paper, we propose a complete framework for the analysis of environmental temporal graphs, with a particular focus on the temporal evolution of environmental structures and relationships. The process is articulated around four successive stages that together form a coherent analytical pipeline. The contributions can be summarized as follows:

- **Modeling stage:** environmental data are structured into temporal multigraphs using temporal intervals to capture dynamic evolutions over time and RCC-8 base relations to encode spatial topology.
- **Mining stage:** Multi-SPMiner [29] is employed to extract significant spatio-temporal patterns that characterize recurring temporal behaviors within the multigraph.
- **Matching stage:** the neural representation learned during mining is reused to efficiently localize pattern occurrences and their temporal manifestations within the graphs.
- **Abstraction stage:** super-nodes are introduced to represent frequent spatio-temporal patterns, while the semantic hierarchy of RCC-8 relations is exploited to aggregate edges, yielding a compact hierarchical representation of temporal graph evolution.

Unlike previous work, which is limited to pattern discovery, this paper extends the framework by introducing a neural pattern-matching strategy that reuses the representation learned during mining to localize pattern occurrences, a multi-scale analysis enabled by the RCC-8 semantic hierarchy, and a hierarchical graph formulation that integrates the discovered patterns as super-nodes together with semantic edge aggregation to produce a compact representation of temporal graph evolution. The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 details the four stages of the proposed process, with particular emphasis on the projection space of Multi-SPMiner, the pattern growth procedure, the structural constraints imposed on the subgraph matching module, and the hierarchical abstraction mechanism. Section 4 presents experimental results obtained on Corine Land Cover data, and Section 5 concludes the paper.

2 Related Work

Spatio-temporal Modeling. Extracting knowledge from massive heterogeneous datasets requires a precise understanding of the elements characterizing phenomena, their spatial distribution, and their temporal occurrence [19]. Early efforts in spatio-temporal modeling

largely relied on the Snapshot model, which organizes data into a set of independent layers where each layer describes spatial states at a fixed time [2]. However, such classical models are generally limited to representing sudden changes and often fail to capture continuous evolution effectively.

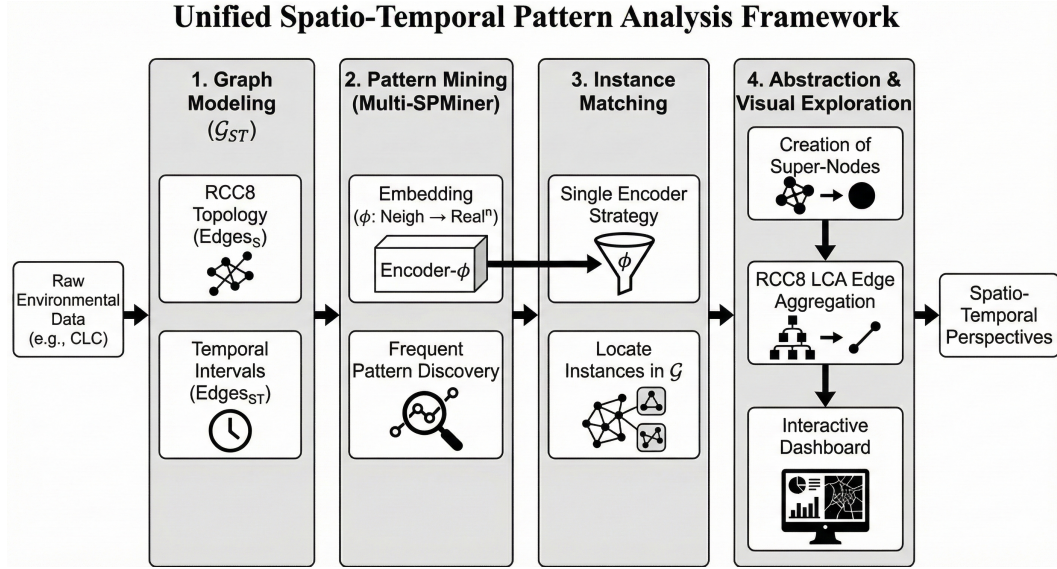
To address the complexities of identity change, graph theory has emerged as a powerful framework for modeling dynamic phenomena, where vertices represent entities and edges represent their relationships. Early graph-based research primarily focused on modeling spatial relations between entities [10]. Subsequent works introduced spatio-temporal graphs where edges link entities at consecutive time steps, thereby representing the connection between spatial footprints across temporal intervals [12]. Other variations include directed graphs that combine spatial relations with spatio-temporal transitions [24], as well as interaction graphs used to simulate landscape processes [4]. Among the existing spatio-temporal graph modeling approaches, Del Mondo et al. [5] proposed a representation framework for evolving geographic entities based on multiple types of relationships. Their model explicitly distinguishes between three categories of relations: spatial, temporal, and filiation. Such a structured representation enables the extraction of meaningful knowledge from spatio-temporal data, including recurrent patterns and interaction patterns.

Pattern Mining in Dynamic Graphs. Frequent pattern mining enables the identification of repetitive structures that reveal underlying interactions within complex systems [11]. However, this process is computationally intensive and known to be NP-hard, particularly when applied to multigraphs, which contain multiple edge types and directions [18]. Spatio-temporal graphs are a specific class of multigraphs that incorporate spatial and temporal dimensions to model dynamic phenomena. While traditional subgraph mining algorithms such as SIGRAM [13], GRAMI [7], and MuGram [9] have been adapted for multigraphs, they often suffer from scalability issues. Similarly, methods targeting dynamic or heterogeneous graphs [21] face limitations regarding genericity and processing speed at scale. To address these constraints, recent approaches in Graph Deep Learning (GDL) have emerged [26, 29], utilizing Graph Neural Networks (GNNs) to identify frequent patterns by offering reliable projections into embedding spaces, from which subgraph pattern frequency can be effectively estimated.

Hierarchical Abstraction of Large Graphs. Processing and representing large-scale temporal graphs raises a fundamental scalability issue known as the 'hairball' phenomenon, where excessive edge density renders the underlying structure illegible [8]. Abstraction techniques are therefore essential to reduce structural complexity while preserving topological information [25]. A standard approach to graph abstraction is the use of compound graphs, where subgraphs are collapsed into single derived vertices, commonly referred to as *super-nodes* [1]. In the context of pattern mining, this strategy enables repetitive structural patterns to be replaced by high-level aggregates, significantly reducing the number of explicit nodes [6].

3 A Multi-Stage Process for Spatio-temporal Graph Analysis

This section presents the proposed process for the analysis of environmental temporal graphs. The pipeline is structured into four successive stages: (1) graph modeling, (2) frequent pattern mining, (3) pattern matching, and (4) hierarchical abstraction, as illustrated in Figure 1. A lightweight visual environment, described at the end of this section, is additionally provided to inspect the outputs of the process.



■ **Figure 1** Overview of the proposed process for analyzing environmental spatio-temporal data.

3.1 Spatio-temporal Graph Modeling

Following the formal model introduced in [5] and its adaptation in [16], we define a spatio-temporal graph as $G = (V, E)$, composed of a spatial subgraph G_S and a spatio-temporal subgraph G_{ST} . Let $O(t)$ denote the set of geographic objects at time t . The vertex set is defined as $V = \bigcup_t O(t)$, and the edge set as $E = E_S \cup E_{ST}$, where E_S encodes spatial relations and E_{ST} encodes temporal relations.

Spatial Relation Modeling (G_S). The spatial subgraph G_S models topological relations between entities coexisting at the same time instant t . Each edge $(u, v) \in E_S$ is labeled with one of the eight exhaustive and mutually exclusive RCC-8 base relations [20]. As proposed in [15], the RCC-8 relation between two objects u and v can be derived from a dimensional intersection matrix evaluating intersections between interiors ($^\circ$), boundaries (∂), and exteriors ($^-$):

$$M(u, v) = \begin{pmatrix} \dim(u^\circ \cap v^\circ) & \dim(u^\circ \cap \partial v) & \dim(u^\circ \cap v^-) \\ \dim(\partial u \cap v^\circ) & \dim(\partial u \cap \partial v) & \dim(\partial u \cap v^-) \\ \dim(u^- \cap v^\circ) & \dim(u^- \cap \partial v) & \dim(u^- \cap v^-) \end{pmatrix}. \quad (1)$$

To limit graph density, the *Disconnected* (DC) relation is treated implicitly; only connectivity relations (EC , PO , TPP , $NTPP$, EQ) generate explicit edges. To handle spatial discontinuities common in environmental polygon data, we additionally model proximity relations following [17]:

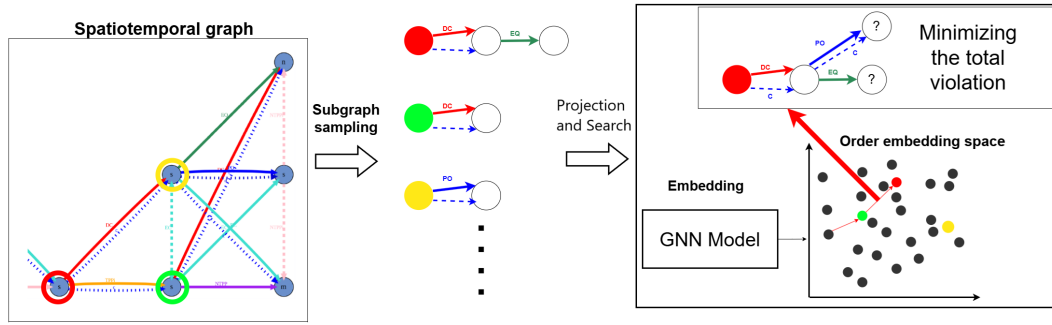
- **Absolute proximity** (p_a), defined using a fixed distance threshold b between object boundaries.
- **Relative proximity** (p_r), defined using a distance threshold proportional to the area of the neighboring object, capturing zones of spatial influence.

Spatio-temporal Relations and Indexing. The spatio-temporal subgraph G_{ST} captures temporal evolution by linking objects between consecutive time steps t and $t+1$. To account for digitization uncertainty in multi-annual datasets, we adopt the thick-boundary model [3]

as suggested by [17]. A temporal edge is created when the buffer zones of two geometries intersect, ensuring topological continuity under minor spatial shifts, using Fuzzy RCC-8 to encode temporal relations. All spatial relation computations are accelerated using an R-tree index, which reduces the number of candidate polygon comparisons through spatial indexing.

3.2 Frequent Pattern Mining with Multi-SPMiner

To identify recurrent structural patterns within the resulting multigraph, we rely on Multi-SPMiner, an extension of the SPMIner architecture [26] adapted for labeled, directed multigraphs [29]. Unlike traditional isomorphism-counting methods, Multi-SPMiner leverages graph deep learning to identify patterns anchored at nodes that exceed a support threshold s . This stage is itself divided into two phases: embedding and pattern search (see Figure 2).



■ **Figure 2** Multi-SPMiner is composed of two main phases: (1) encoding, which focuses on learning graph embeddings, and (2) decoding, which is dedicated to the search and extraction of frequent patterns [29].

3.2.1 Order-Preserving Embedding

The graph G is first decomposed into a set of neighborhoods $\mathcal{N} = \{N_v \mid v \in V\}$, where each N_v represents the k -hop subgraph anchored at node v . We set k to be larger than the maximum expected diameter of a pattern. A Multigraph Convolutional Network (MGCN) is then used to project these neighborhoods into an ordered embedding space $\mathbb{R}^n$. The embedding function $\phi : \mathcal{G} \mapsto \mathbb{R}^n$ is trained to preserve the subgraph isomorphism relation. Formally, for two graphs A and B , we enforce the order constraint

$$A \subseteq B \iff \phi(A) \leq \phi(B), \quad (2)$$

where $\leq$ denotes a coordinate-wise inequality. In the 2D plane, this implies that the embedding of a subgraph A must reside to the bottom-left of the embedding of its supergraph B . The MGCN is optimized by minimizing a max-margin loss function over positive pairs (subgraph relationships) and negative pairs:

$$\mathcal{L} = \sum_{(A,B) \in P} E(A,B) + \sum_{(A',B') \in N} \max(0, \alpha - E(A',B')), \quad (3)$$

where α is a margin hyperparameter and E is the order violation penalty defined as

$$E(A,B) = \|\max(0, \phi(A) - \phi(B))\|^2. \quad (4)$$

3.2.2 Pattern Search

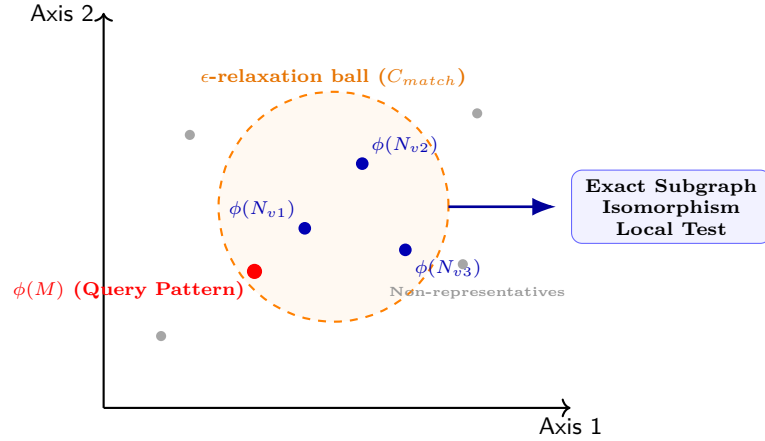
Once the neighborhoods are embedded, a heuristic search is performed to identify frequent patterns. The search is iterative, starting from a randomly selected single-node graph. At each step, the candidate pattern M is grown by adding adjacent nodes and edges found in the underlying multigraph. The growth strategy is guided by the embeddings: we seek to construct a pattern M that is a subgraph of as many neighborhoods $N \in \mathcal{N}$ as possible. This is achieved by minimizing the total order violation margin $m(M)$ across the dataset:

$$m(M) = \sum_{N \in \mathcal{N}} \|\max(0, \phi(M) - \phi(N))\|^2. \quad (5)$$

This optimization ensures that the discovered pattern M effectively fits into the embeddings of a large subset of the graph's neighborhoods.

3.3 Pattern Matching via Same-Encoder Strategy

Once an abstract pattern definition M has been identified, the next step consists in locating all its specific instances $\{I_1, I_2, \dots, I_k\}$ within the global graph G . To this end, we adopt a *Same-Encoder Strategy*. This approach leverages the property that the MGCN encoder ϕ was explicitly trained to preserve subgraph isomorphism relationships in the embedding space. Accordingly, no separate global matching algorithm is required before candidate filtering: the pre-trained encoder ϕ is directly reused as a high-efficiency filter (see Figure 3).



■ **Figure 3** Subgraph matching principle. The model projects local neighborhoods into a latent space using the function ϕ learned during the mining stage, translating topological inclusion into a geometric ordering constraint. By applying the same encoder to the query pattern, occurrences are identified directly through their relative positions in the embedding space. The circle denotes the neighborhood defined by an epsilon margin, used to apply an isomorphism test for more precise pattern matching.

For a target pattern $M \in \mathcal{M}$ treated as a query, a node $v \in V$ is considered a candidate anchor for an instance if the order constraint is satisfied in the embedding space, i.e., $\phi(M) \leq \phi(N_v)$. This condition implies that M is likely a subgraph of the neighborhood anchored at v . This strategy drastically prunes the search space. The candidate retrieval stage effectively amounts to an approximate range query in the embedding space, followed by a local isomorphism verification to confirm the exact RCC-8 and temporal edge configurations.

The encoder ϕ enforces a partial order on the vector space, as expressed in Equation (2). Consequently, for a node v to be a candidate anchor for pattern M , its neighborhood embedding $\phi(N_v)$ must structurally “contain” the pattern embedding $\phi(M)$. We define the *order violation* term $V(M, N_v)$ as the magnitude of the violation of this inequality:

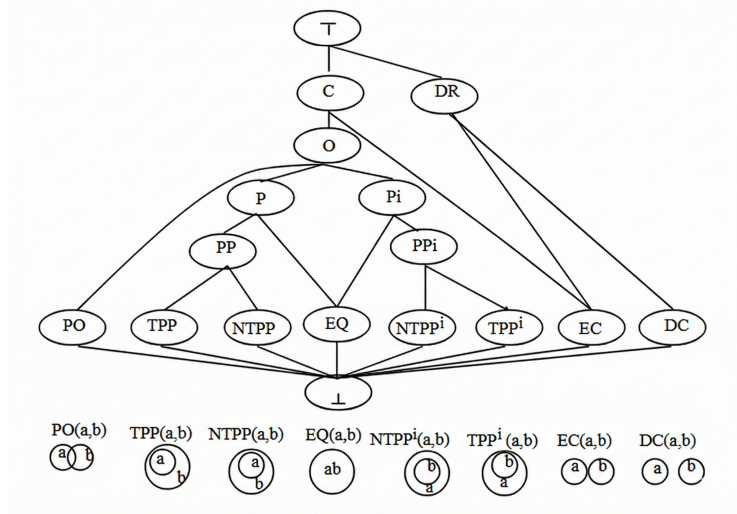
$$V(M, N_v) = \|\max(0, \phi(M) - \phi(N_v))\|^2. \quad (6)$$

Ideally, for a perfect matching, $V(M, N_v) = 0$. However, given the approximate nature of graph neural networks and the continuous nature of the latent space, we introduce a relaxation threshold ϵ . The set of candidate anchor nodes C_{match} for pattern M is defined as

$$C_{match} = \{v \in V \mid V(M, N_v) \leq \epsilon\}. \quad (7)$$

For each candidate anchor $v \in C_{match}$, the corresponding k -hop neighborhood N_v is extracted from the original graph, and an exact subgraph isomorphism test is applied between M and N_v . By restricting exact matching to neighborhoods selected through proximity in the embedding space, the combinatorial search space is significantly reduced while structural correctness is preserved.

3.4 Hierarchical Abstraction via Partitioned LCA



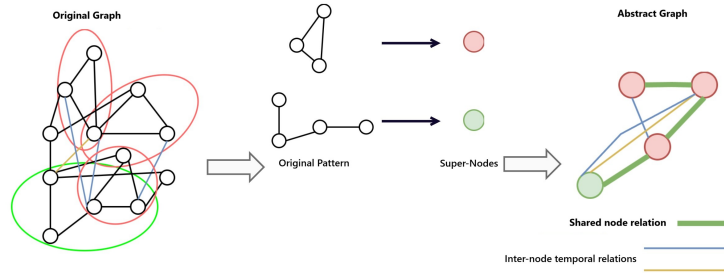
■ **Figure 4** Hierarchical representation of RCC relations [20]. The RCC-8 relations form the lower level of the hierarchy.

The outputs produced by the mining and matching stages are numerous and still interleaved with the original graph structure. To produce a compact representation of the graph that remains faithful to its semantics, we aggregate detected pattern instances into super-nodes. However, collapsing all interactions between two super-nodes S_i and S_j into a single averaged or majority relation hides essential semantic structure. Environmental spatio-temporal graphs frequently exhibit *mixed* RCC-8 relations between the same regions, reflecting genuinely complex dynamics that should not be oversimplified into a single generic link. To faithfully capture this variability, we introduce a flexible multi-scale abstraction mechanism driven by the hierarchical structure of the RCC relations [20] (see Figure 4).

Partitioned LCA with User-Controlled Granularity. The RCC hierarchy $\mathcal{H}$ forms a fixed-depth lattice of four conceptual levels (from the root relation T to the atomic leaves). A discrete granularity parameter $\lambda \in \{0, 1, 2, 3, 4\}$ specifies the desired level of abstraction. Increasing λ by one moves the abstraction one level upward in the RCC hierarchy:

- $\lambda = 0$: maximum precision, where relations remain at the atomic level;
- $\lambda = 1, 2, 3$: intermediate abstraction levels, each merging relations according to the corresponding level of the RCC hierarchy;
- $\lambda = 4$: maximum abstraction, where all relations collapse to the root relation T .

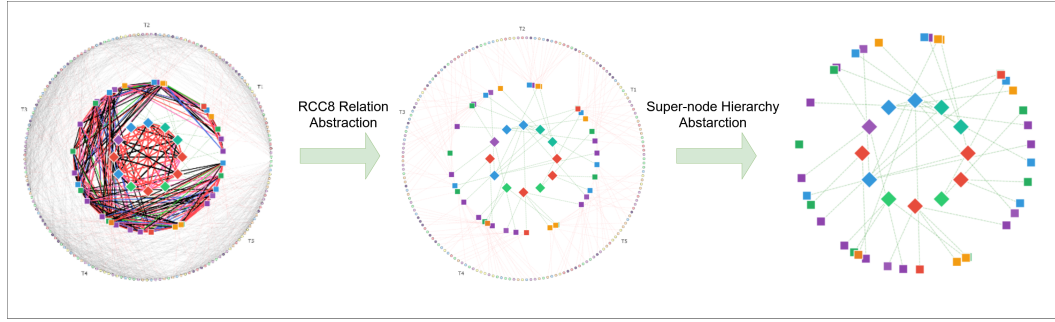
Let Ω_{ij} denote the set of RCC relations observed between the nodes composing super-nodes S_i and S_j . Instead of merging all these relations into a single abstract label, we group them according to their proximity in the RCC-8 hierarchy. Two relations are assigned to the same group if their lowest common ancestor in the hierarchy is sufficiently specific with respect to the chosen granularity level λ . Each group then produces a single abstract edge labeled by this common ancestor. This grouping strategy ensures that semantically similar relations are merged while distinct spatial relations are preserved. As a result, the abstraction between two super-nodes may consist of multiple edges, each representing a different type of spatial interaction. An illustrative example is presented in Figure 5.



■ **Figure 5** Following the pattern search and matching stages on the original graph, identified pattern instances are aggregated into super-nodes. At this higher abstraction level, these super-nodes are retained while independent nodes are filtered out, preserving only the most critical spatio-temporal dynamics.

3.5 A Complementary Visual Environment for Result Exploration

The four-stage process described above produces a hierarchical, compact, and semantically-aware representation of the input temporal graph. To facilitate the inspection of these outputs, we additionally provide a lightweight visual environment. This environment is not a contribution of the process itself; it is a companion tool whose role is to let the analyst browse mining and matching results, examine super-nodes, and compare configurations obtained on different geographic areas. Concretely, the environment offers a coordinated multi-view layout combining an abstract graph representation of the hierarchical output with a geospatial anchor view, so that patterns produced by the pipeline can be related to their real-world geographic counterparts. Selecting a super-node highlights its corresponding geographic footprint, while selecting a geographic region reveals the associated patterns, and semantic and temporal filters are used to reduce visual load when exploring dense results (see Figure 6). While the proposed framework enables the discovery of a variety of spatio-temporal patterns, a comprehensive analysis of these patterns and their relevance from a domain perspective is outside the scope of this work. Such an analysis is presented in [28].



■ **Figure 6** Example of hierarchical abstraction of a spatio-temporal graph. Left: dense original graph. Middle: aggregation of RCC8 relations. Right: super-node representation based on frequent patterns.

4 Experiments and Results

To evaluate the proposed process, we conducted an extensive case study on real-world temporal data. The experiments aim to assess the ability of the pipeline to (i) extract meaningful temporal patterns, (ii) efficiently localize their occurrences with high accuracy, and (iii) produce a compact yet interpretable hierarchical abstraction of large graphs through super-node representations. We additionally report on the implementation details and execution times that characterize the practical behavior of the pipeline.

Data Description. The CORINE Land Cover (CLC)¹ dataset is a large-scale geospatial dataset produced at the European Union level from satellite imagery. The dataset follows a hierarchical nomenclature with three levels of semantic granularity, covering categories such as artificial surfaces, agricultural areas, forests, wetlands, and water bodies. In our experiments, we rely on the second level of the classification, resulting in 15 distinct land-use types. The minimum mapping unit is 25 hectares. The dataset is available as a long-term time series spanning five temporal snapshots: 1990, 2000, 2006, 2012, and 2018. Each land polygon is represented as a node labeled according to its land-use class.

4.1 Impact of the Encoder Architecture on the Mining Stage

To analyze the robustness of the mining stage, we evaluate the impact of different GNN-based encoders on the identification of frequent spatio-temporal patterns. The objective of this experiment is twofold: (i) to assess the stability of the embedding-based pattern representation across architectures, and (ii) to evaluate the generalization capability of the learned encoders across synthetic and real-world spatio-temporal graphs. We define motif detection accuracy (equivalent to precision in this context) as the percentage of discovered motifs that are frequent motifs.

Two complementary training scenarios are considered. In the first setting, models are trained exclusively on synthetically generated spatio-temporal graphs containing patterns of predefined sizes, produced with a random ST-graph generator [16]. These synthetic graphs are designed to closely resemble real-world conditions by incorporating five temporal snapshots and high-frequency pattern insertions. The trained models are then evaluated on

¹ <https://land.copernicus.eu/en/products/corine-land-cover>

real spatio-temporal graphs generated from the CLC dataset, constructed from 20 randomly selected European cities. This scenario evaluates the ability of the model to generalize across domains after a single training phase. In the second setting, both training and evaluation graphs are derived from distinct geographic regions within the CLC dataset. For each scenario, 200 graphs are used for training and an additional set of 20 graphs is reserved for testing. The evaluation focuses on patterns of sizes 6, 8, and 10, which represent a practical compromise between expressiveness and computational complexity.

■ **Table 1** Pattern identification accuracy on the CLC dataset for different GNN-based encoders, evaluated on patterns of sizes 6, 8, and 10, under synthetic and real-data training scenarios. Bold values denote the best result per row.

Target pattern size	Training Data	MGCNN (Baseline)	GAT	R-GCN	GGCN	HAN	MGAT	MMAGAT
6	Synthetic	0.67 ± 0.01	0.69 ± 0.02	0.70 ± 0.02	0.71 ± 0.02	0.72 ± 0.02	0.72 ± 0.01	0.73 ± 0.02
	CLC	0.68 ± 0.01	0.69 ± 0.02	0.71 ± 0.01	0.72 ± 0.02	0.73 ± 0.02	0.73 ± 0.02	0.74 ± 0.01
8	Synthetic	0.66 ± 0.03	0.70 ± 0.02	0.69 ± 0.03	0.69 ± 0.02	0.70 ± 0.02	0.68 ± 0.03	0.71 ± 0.02
	CLC	0.66 ± 0.02	0.70 ± 0.03	0.71 ± 0.01	0.71 ± 0.02	0.72 ± 0.02	0.73 ± 0.02	0.73 ± 0.03
10	Synthetic	0.65 ± 0.01	0.68 ± 0.03	0.70 ± 0.03	0.68 ± 0.02	0.68 ± 0.03	0.68 ± 0.03	0.69 ± 0.02
	CLC	0.64 ± 0.03	0.67 ± 0.02	0.68 ± 0.03	0.68 ± 0.02	0.69 ± 0.03	0.69 ± 0.03	0.69 ± 0.01

Table 1 reports the pattern identification accuracies obtained for the seven encoders considered: the MGCN baseline, Graph Attention Network (GAT), Relational Graph Convolutional Network (R-GCN) [23], Gated GCN (GGCN), Heterogeneous Attention Network (HAN), Multi-view GAT (MGAT), and Multi-Mask Aggregators for GAT (MMAGAT) [22]. Three complementary observations can be drawn from these results. First, all GNN encoders surpass the MGCN baseline by a margin ranging from 2 to 6 percentage points across pattern sizes, confirming that enriched relational representations consistently benefit pattern discovery within the proposed mining stage. Second, the gap between models trained on synthetic data and those trained directly on real CLC graphs remains marginal (less than 1% on average), which suggests that the learned order embeddings capture transferable structural and temporal regularities that are largely independent of the data source. This is a practically valuable property since the cost of generating ground truth on synthetic graphs is significantly lower than on real graphs. Third, attention-based architectures (HAN, MGAT, MMAGAT) exhibit a consistent advantage, particularly for smaller patterns (size 6), with MMAGAT achieving the best overall performance. This behavior can be attributed to their ability to selectively emphasize salient spatial and temporal relations, which is especially beneficial in multigraph settings involving heterogeneous edge semantics such as RCC-8 and temporal links. The relative gap between encoders narrows as pattern size increases, reflecting the intrinsic difficulty of mining larger structures that span multiple temporal snapshots.

■ **Table 2** Average mining-stage execution time per CLC graph (seconds) and number of trainable parameters for each encoder. Times include both pattern search and per-pattern decoding.

Metric	MGCNN	GAT	R-GCN	GGCN	HAN	MGAT	MMAGAT
Avg. mining time (s)	19.4	27.1	25.8	29.2	28.5	27.6	28.7
Trainable params (M)	0.18	0.31	0.42	0.54	0.49	0.38	0.61

Table 2 reports the average mining-stage execution time per CLC graph, alongside the number of trainable parameters for each encoder. All experiments were conducted on a machine equipped with an NVIDIA GeForce RTX 4080 SUPER GPU, 64 GB of RAM, and an AMD Ryzen 7 9800X3D processor. All neural encoders complete the mining stage in under 30 seconds per graph, which contrasts sharply with the ground-truth MuGram

algorithm [9] that requires on average 14 minutes (840 seconds) per CLC graph of comparable size. The MGCN baseline is the lightest both in execution time and in parameter count, while MMAGAT, despite reaching the highest accuracy, is the most parameter-intensive encoder due to its per-neighbor relational masks. R-GCN offers a favorable trade-off between accuracy and footprint and may be preferred in resource-constrained settings.

4.2 Effectiveness of Pattern Matching

The effectiveness of the matching stage relies on the same neural encoder used during the mining stage. The objective of this experiment is twofold: (i) to assess the precision and recall of embedding-based pattern localization using the Same-Encoder strategy alone, and (ii) to quantify the benefits of integrating an exact subgraph isomorphism verification step within the local neighborhoods retrieved in the embedding space. Two matching configurations are considered:

- **Same-Encoder (Neural)**: pattern occurrences are identified solely on the basis of the order-embedding constraint in the latent space, controlled by the relaxation threshold ϵ .
- **Same-Encoder + Subgraph Isomorphism (Neural + Iso.)**: candidate occurrences retrieved in the embedding space are further validated using an exact subgraph isomorphism test within the corresponding local graph neighborhoods.

To rigorously characterize the matching stage, we report both precision and recall against the exhaustive ground truth produced by MuGram. Recall measures the fraction of true pattern occurrences that are recovered, whereas precision measures the fraction of returned occurrences that correspond to genuine matches. This distinction is essential because the embedding-based retrieval, governed by the relaxation threshold ϵ , is intentionally tuned to avoid missing true positives, which inflates the candidate set and inevitably introduces false positives. The threshold ϵ is estimated at the end of the training phase by analyzing the average distance between different embedding representations of identical patterns and averaging these distances across the training set. More specifically, ϵ is defined as a vector having the same dimensionality as the embedding space, where each component captures the average admissible variation observed for the corresponding embedding dimension.

■ **Table 3** Pattern matching precision and recall (%) on the CLC dataset for the Same-Encoder (Neural) configuration. Recall is by design near-saturated, while precision is limited by false positives introduced by the approximate nature of continuous embeddings.

Metric	MGCNN (Baseline)	GAT	R-GCN	MGAT	MMAGAT
Recall	95.62 $\pm$ 0.39	95.78 $\pm$ 0.36	95.81 $\pm$ 0.34	95.93 $\pm$ 0.28	96.04 $\pm$ 0.26
Precision	86.74 $\pm$ 0.62	87.41 $\pm$ 0.58	87.86 $\pm$ 0.55	88.92 $\pm$ 0.49	89.37 $\pm$ 0.46

■ **Table 4** Average matching execution time per CLC graph for each encoder.

Matching Strategy	MGCNN (Baseline)	GAT	R-GCN	MGAT	MMAGAT
Neural	1m18s	1m21s	1m19s	1m23s	1m24s
Neural + Iso.	3m33s	3m35s	3m36s	3m41s	3m43s

Tables 3 and 4 jointly summarize the matching performance of the pipeline, from which three observations can be drawn. First, recall remains consistently high and stable around 95.9% across all encoders, and is essentially unaffected by the verification step since the latter only filters previously retrieved candidates. The relaxation threshold ϵ is intentionally

calibrated to exhaustively cover the candidate set, ensuring that the embedding-based retrieval rarely misses a true occurrence. The small inter-encoder differences reflect the fidelity with which each encoder preserves subgraph containment in the latent space: more expressive encoders such as MGAT and MMAGAT produce embeddings that leave fewer true matches outside the ϵ -ball. Precision differs markedly between the two configurations. The Same-Encoder strategy alone yields precision values ranging from 86.7% (MGCN baseline) to 89.4% (MMAGAT). Once the exact subgraph isomorphism verification is applied within the retrieved neighborhoods, precision is perfect for all encoders, which justifies the omission of a dedicated table for the post-verification values, as the step operates as a high-precision verification step that removes false positives almost entirely without affecting true positives. The additional verification cost remains modest. Pure neural matching is completed in approximately 1m20s per CLC graph, whereas the Neural + Iso. configuration takes around 3m40s. This overhead is contained because exact matching is restricted to the small set of candidates retrieved in the embedding space rather than applied globally over the full graph. By comparison, a purely algorithmic subgraph isomorphism approach applied directly to the entire graph requires up to 22 minutes per graph. Overall, these results validate the design of the matching stage: the neural retrieval component ensures high recall and fast candidate generation, while the isomorphism step delivers high precision. Decoupling the two roles allows each component to be tuned independently and yields a matching stage whose final precision-recall profile is significantly more favorable than what either component could achieve in isolation.

4.3 Hierarchical Abstraction

The abstraction stage transforms the rich but dense output of the matching stage into a compact, semantically-aware hierarchical representation of the graph. Frequent patterns capture recurrent structural and semantic configurations that characterize the dominant spatio-temporal behaviors observed in the data, while rare or noisy interactions are naturally filtered out. By relying on frequent patterns as building blocks, the abstraction stage produces a synthetic summary of the underlying graph structure: instead of manipulating thousands of individual nodes and heterogeneous relations, the graph is summarized through a reduced set of pattern instances and their interactions. We now evaluate this abstraction quantitatively, by measuring the compression achieved at successive hierarchical levels, and qualitatively, by inspecting the structural dependencies that emerge between super-nodes.

We apply the abstraction to spatio-temporal graphs of approximately 2000 nodes extracted from different European regions, and measure the structural compression achieved at three successive levels of the hierarchy: (L0) the original raw graph; (L1) the first abstraction layer obtained by replacing detected pattern instances with first-order super-nodes; (L2) the second abstraction layer obtained by re-applying the mining and matching stages on the L1 graph and aggregating the resulting pattern co-occurrences into second-order super-nodes.

Table 5 shows that the first abstraction layer reduces the size of the graph by approximately 71.5% in nodes and 74.8% in edges on average, while the second layer drives this compression to an average reduction of 91.4% in nodes and 93.6% in edges with respect to the original graph. Although the absolute graph sizes are comparable across the five cities considered, the compression rates exhibit meaningful city-level variability: more structurally homogeneous cities such as Helsinki and Madrid yield stronger compression (around 73% at L1 and 92% at L2), reflecting the presence of more redundant structural patterns that the abstraction can collapse, whereas more heterogeneous land-use mosaics such as Berlin and Vienna yield slightly weaker compression (around 70% at L1 and 90% at L2). This pattern suggests that

■ **Table 5** Quantitative evaluation of the hierarchical abstraction on five representative European cities. All graphs use the second-level CLC nomenclature (15 land-use classes). Mining and matching are recursively applied at each level using the same MMAGAT encoder, with an estimated frequency threshold of approximately 25 occurrences per pattern. $|V|$ and $|E|$ denote the number of nodes and edges at each level. Reduction percentages are computed with respect to the original graph (L0).

City	L0 (Raw)		L1 (pattern super-nodes)				L2 (Co-occurrence super-nodes)			
	$ V $	$ E $	$ V $	$ E $	$\Delta V $	$\Delta E $	$ V $	$ E $	$\Delta V $	$\Delta E $
Paris	2148	8316	612	2104	−71.5%	−74.7%	187	542	−91.3%	−93.5%
Berlin	2103	8194	638	2189	−69.7%	−73.3%	204	591	−90.3%	−92.8%
Madrid	2167	8358	581	2002	−73.2%	−76.0%	168	487	−92.2%	−94.2%
Vienna	2089	8087	624	2126	−70.1%	−73.7%	195	562	−90.7%	−93.0%
Helsinki	2118	8205	568	1949	−73.2%	−76.2%	162	471	−92.4%	−94.3%
Mean	2125	8232	605	2074	−71.5%	−74.8%	183	531	−91.4%	−93.6%

the abstraction mechanism captures regularities that are intrinsic to the underlying spatio-temporal process while remaining sensitive to the specific structural diversity of each region. The relative reduction in edges is consistently higher than the reduction in nodes across all cities because the partitioned LCA mechanism merges semantically similar RCC-8 relations between the same pair of super-nodes into a single abstract edge, while still preserving genuinely distinct relations. By contrast, applying a higher granularity λ on the original graph aggressively merges RCC-8 relations along their hierarchy, which mechanically increases the number of detectable structural patterns but yields patterns of limited significance, since they collapse semantically distinct configurations into a single coarse label.

4.4 Discussion

Taken together, the experiments characterize the proposed process along three complementary axes. The mining stage identifies frequent patterns with an accuracy of approximately 72% on real CLC data using the best encoder (MMAGAT), while running roughly an order of magnitude faster than the exhaustive MuGram baseline. The matching stage achieves a stable recall of approximately 95.9% and a perfect precision once the isomorphism verification step is added, with an overall execution time that remains within a few minutes per graph. Finally, the abstraction stage compresses the graph by a factor of three to ten across two hierarchical levels while preserving multi-relational structure through a tunable granularity parameter. The clear separation of roles between stages, the use of a single encoder for both mining and matching, and the recursive applicability of the pipeline to its own outputs together yield a process that is interpretable, scalable, and faithful to the multi-relational nature of environmental spatio-temporal graphs.

5 Conclusion and Perspectives

This paper has presented a complete process for mining, matching, and abstracting frequent spatio-temporal patterns in large environmental graphs. The pipeline addresses both the scalability and the interpretability challenges inherent to real-world spatio-temporal data. Experiments on the Corine Land Cover dataset have shown that the learned embeddings robustly capture recurrent spatio-temporal structures and generalize across synthetic and real data. The Same-Encoder matching strategy enables efficient pattern localization, while the integration of a constrained subgraph isomorphism verification step improves accuracy

with a controlled computational overhead, offering a favorable trade-off compared to purely algorithmic approaches. Finally, the hierarchical abstraction based on frequent patterns and partitioned aggregation produces a compact and interpretable summary of complex graphs, and enables the identification of higher-order pattern co-occurrences.

Future work will focus on extending the process to dynamic pattern evolution analysis and to other classes of spatio-temporal graphs, as well as on refining the companion visual environment to better support the exploratory analysis of pipeline outputs.

References

- 1 James Abello, Frank Van Ham, and Neeraj Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, 2006. doi:10.1109/TVCG.2006.120.
- 2 J. Chen, H. Wu, S. Li, A. Liao, C. He, and S. Peng. Temporal logic and operation relations based knowledge representation for land cover change web services. *ISPRS Journal of Photogrammetry and Remote Sensing*, 83:140–150, 2013.
- 3 Eliseo Clementini and P. Di Felice. Approximate topological relations. *International Journal of Approximate Reasoning*, 16(2):173–204, 1997. doi:10.1016/S0888-613X(96)00127-2.
- 4 P. Degenne and D. Lo Seen. Ocelet: Simulating processes of landscape changes using interaction graphs. *SoftwareX*, 5:89–95, 2016. doi:10.1016/J.SOFTX.2016.05.002.
- 5 G. Del Mondo, J. G. Stell, C. Claramunt, and R. Thibaud. A graph model for spatio-temporal evolution. *Journal of Universal Computer Science*, 16:1452–1477, 2010. doi:10.3217/JUCS-016-11-1452.
- 6 Cody Dunne and Ben Shneiderman. Motif simplification: improving network visualization readability with pattern replacement. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1923–1932, 2013.
- 7 M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis. Grami: Frequent subgraph and pattern mining in a single large graph. *Proceedings of the VLDB Endowment*, 7(7):517–528, 2014. doi:10.14778/2732286.2732289.
- 8 Ivan Herman, Guy Melançon, and M. Scott Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000. doi:10.1109/2945.841119.
- 9 V. Ingalalli, D. Ienco, and P. Poncelet. Mining frequent subgraphs in multigraphs. *Information Sciences*, 451:50–66, 2018.
- 10 Bin Jiang and Christophe Claramunt. Topological analysis of urban street networks. *Environment and Planning B: Planning and design*, 31(1):151–162, 2004.
- 11 C. Jiang, F. Coenen, and M. Zito. A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28(1):75–105, 2013. doi:10.1017/S0269888912000331.
- 12 L. Khiali, D. Ienco, and M. Teisseire. Object-oriented satellite image time series analysis using a graph-based representation. *Ecological Informatics*, 43:52–64, 2018. doi:10.1016/J.ECOINF.2017.11.003.
- 13 M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph. *Data mining and knowledge discovery*, 11(3):243–271, 2005. doi:10.1007/S10618-005-0003-9.
- 14 Suprava Ranjan Laha, Binod Kumar Pattanayak, and Saumendra Pattnaik. Advancement of environmental monitoring system using IoT and sensor: A comprehensive analysis. *AIMS Environmental Science*, 9(6):771–800, 2022.
- 15 F. Le Ber and A. Napoli. Design and comparison of lattices of topological relations for spatial representation and reasoning. *Journal of Experimental & Theoretical Artificial Intelligence*, 15(3):331–371, 2003. doi:10.1080/0952813031000119700.
- 16 Aurélie Leborgne, Marija Kirandjiska, and Florence Le Ber. Random generation of a locally consistent spatio-temporal graph. In *International Conference on Conceptual Structures*, pages 155–169. Springer, 2021. doi:10.1007/978-3-030-86982-3_12.

- 17 Aurélie Leborgne, Adrien Meyer, Henri Giraud, Florence Le Ber, and Stella Marc-Zwecker. Un graphe spatio-temporel pour modéliser l'évolution de parcelles agricoles. In *SAGEO 2019*, Clermont-Ferrand, France, 2019.
- 18 Pei Li, Jianyong Yu, Jianxun Liu, Dong Zhou, and Buqing Cao. Generating weighted social networks using multigraph. *Physica A: Statistical Mechanics and its Applications*, 539:122894, 2020.
- 19 Donna J. Peuquet and Elizabeth Wentz. An approach for time-based analysis of spatiotemporal data. In *Advances in GIS research. Proc. 6th symposium*, pages 489–504, 1994.
- 20 David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 165–176, 1992.
- 21 Abhik Ray, Lawrence B Holder, and Albert Bifet. Efficient frequent subgraph mining on large streaming graphs. *Intelligent Data Analysis*, 23(1):103–132, 2019. doi:10.3233/IDA-173705.
- 22 Ahmet Sarigün and Ahmet Sureyya Rifaioglu. Multi-mask aggregators for graph neural networks. In *The First Learning on Graphs Conference*, 2022. URL: <https://openreview.net/forum?id=hZ3b8CskgC>.
- 23 Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks, 2017. arXiv:1703.06103.
- 24 M. Sriti, R. Thibaud, and C. Claramunt. A fuzzy identity-based temporal gis for the analysis of geomorphometry changes. In *Journal on Data Semantics III*, pages 81–99, 2005.
- 25 Tatiana Von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J. Van Wijk, Jean-Daniel Fekete, and Dieter W. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011. doi:10.1111/J.1467-8659.2011.01898.X.
- 26 R. Ying, A. Wang, J. You, and J. Leskovec. Frequent subgraph mining by walking in order embedding space. *The International Conference on Machine Learning (ICML)*, 2020.
- 27 Assaad Zeghina, Aurélie Leborgne, Florence Le Ber, and Antoine Vacavant. Deep qminer: Towards a generalized deep q-learning approach for graph pattern mining. In Luc Brun, Vincenzo Carletti, Sébastien Bogleux, and Benoît Gaüzère, editors, *Graph-Based Representations in Pattern Recognition*, pages 134–144, Cham, 2025. Springer Nature Switzerland.
- 28 Assaad Zeghina, Aurélie Leborgne, Florence Le Ber, and Antoine Vacavant. Neural pattern mining in spatiotemporal multigraphs: A pan-european land-use study. In *International Conference on Pattern Recognition Workshops GREEN-PR*, 2026.
- 29 Assaad Zeghina, Aurélie Leborgne, Florence Le Ber, and Antoine Vacavant. Multi-SPMiner: A Deep Learning Framework for Multi-Graph Frequent Pattern Mining with Application to Spatiotemporal Graphs. *Procedia Computer Science*, 225:1094–1103, 2023. doi:10.1016/J.PROCS.2023.10.097.
- 30 Assaad Zeghina, Aurélie Leborgne, Florence Le Ber, and Antoine Vacavant. Deep learning on spatiotemporal graphs: A systematic review, methodological landscape, and research opportunities. *Neurocomputing*, 594:127861, 2024. doi:10.1016/j.neucom.2024.127861.
- 31 Z. Zhang, P. Cui, and W. Zhu. Deep learning on graphs: A survey. *IEEE Trans. on Knowledge and Data Engineering*, 34(1):249–270, 2020.