Int. J. Nonlinear Anal. Appl. In Press, 1–11
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2026.40365.5630

LifeSage: Existence-aware temporal graphs with multi-interval Lifespans

Mohamad Ali Berahman, Amirhossein Sobhani*, Madjid Eshaghi Gordji

Department of Mathematics, Statistics and Computer Science, Semnan University, Semnan, Iran

(Communicated by Mohammad Bagher Ghaemi)

Abstract

Classical temporal graph models represent dynamics by allowing edges to appear and disappear over time while implicitly assuming that the underlying vertex set remains fixed. This assumption conflates fundamentally different phenomena, namely behavioral inactivity and structural non-existence of entities, and may lead to inaccurate representations in systems where vertices temporarily or permanently disappear and later reappear. In this work, we introduce *LifeSage*, an existence-aware temporal graph framework in which vertex existence is treated as a time-dependent quantity. Each vertex is associated with a multi-interval lifespan, enabling explicit modeling of intermittent, cyclic, and irreversible presence patterns. Temporal snapshots are interpreted as projections onto the set of existentially valid vertices, leading to masked adjacency representations in which non-existing entities are structurally excluded rather than treated as inactive. We further distinguish between continuous flows, which require simultaneous existence of both endpoints, and discrete (transit) flows, which may persist independently of endpoint existence. This distinction allows temporal snapshots to violate the classical incidence condition and motivates an existence-aware masking perspective. We also discuss snapshot equivalence, event times, and representative generator snapshots arising from structural transitions in the graph. Finally, we outline how the proposed framework may support existence-aware temporal representations and learning settings on dynamic systems with evolving entity sets.

Keywords: temporal graphs; dynamic networks; vertex existence; multi-interval lifespans; masked adjacency; existence-aware modeling; temporal snapshots
2020 MSC: Primary 05C90; Secondary 68R10, 05C82

1 Introduction

Temporal graphs provide a fundamental mathematical framework for modelling systems in which interactions evolve over time. Such models have been extensively studied across a wide range of domains, including communication networks, transportation systems, social dynamics, and biological processes [7, 11, 10, 1].

Comprehensive treatments of temporal network theory further emphasize that temporal dynamics arise not only from the ordering of interactions, but also from the evolving structural context in which these interactions are embedded [11].

*Corresponding author: a_sobhani@semnan.ac.ir
Email addresses: mohamadali_berahman@semnan.ac.ir (Mohamad Ali Berahman), a_sobhani@semnan.ac.ir (Amirhossein Sobhani), meshaghi@semnan.ac.ir (Madjid Eshaghi Gordji)

Received: January 2026 *Accepted:* April 2026

In most classical formulations, temporal variability is introduced by allowing edges to appear and disappear over time, while the underlying set of vertices is implicitly assumed to remain fixed [9, 12]. Within this paradigm, the absence of interactions involving a vertex is typically interpreted as a form of inactivity or reduced participation. However, this modelling assumption fails to capture a fundamental aspect of many real-world systems: entities themselves may temporarily or permanently cease to exist and later reappear. Existing approaches to dynamic graphs typically retain vertices throughout the entire temporal horizon, regardless of their actual presence in the system [13, 8, 15]. As a result, vertex absence is often conflated with inactivity, despite the increasing interest in learning-based models for temporal graph representation [13].

This conceptual limitation is illustrated by several motivating scenarios; see Examples 4.9, 4.10, and 5.3 in [4], where temporal graphs are analysed from an algebraic and topological perspective that emphasizes structural consistency across time. In a classroom setting, for instance, a teacher may temporarily leave the room during a break. Classical temporal models continue to represent the teacher as a vertex with deactivated edges, thereby permitting spurious inferences about potential interactions despite the teacher's physical absence. In transportation networks, small regional airports may close overnight and reopen cyclically, a behaviour that cannot be adequately represented by models assuming persistent vertex existence [5, 3]. Similarly, in communication networks, temporary device shutdowns and permanent device failures both manifest as missing signals, yet their structural and temporal implications are fundamentally different [14, 6].

Analogous distinctions arise in biological systems. In neuroscience, for example, a neuron in a resting state remains physically present and capable of future activation, whereas a neuron undergoing apoptosis has irreversibly ceased to exist. Network-based models of brain dynamics highlight the importance of such distinctions for accurately capturing neural structure and function [2]. Collectively, these examples demonstrate that temporal dynamics are governed not only by the activation and deactivation of interactions, but also by the explicit appearance, disappearance, and reappearance of entities themselves. Such disappearance may be temporary, permanent, or cyclic, and should be regarded as a first-order structural feature rather than a peripheral modelling detail.

Motivated by these observations, we introduce *LifeSage*, an existence-aware extension of the temporal graph framework developed in [4]. While the algebraic and temporal foundations of lifespan-aware graphs and temporal (f, g)-homomorphisms originate from [4], the present work focuses specifically on the distinction between existential absence and behavioral inactivity in temporal graph representations.

In the proposed framework, vertices may occupy three distinct states: *active* (present and interacting), *sleeping* (present but non-interacting), and *dead* (temporarily or permanently absent from the system). To represent these states, we introduce existence-aware masked snapshot representations in which non-existing vertices are structurally excluded from the effective graph representation rather than treated as isolated inactive nodes.

This masking perspective provides a formal distinction between inactivity and non-existence at the representation level and motivates existence-aware formulations for temporal graph learning and message propagation. In addition, we discuss how discrete or transit interactions may persist beyond the simultaneous existence of both endpoint vertices, leading to non-classical temporal snapshots in which the standard incidence condition may fail.

Rather than introducing a completely new temporal graph formalism, the purpose of LifeSage is to extend existing lifespan-aware temporal graph models with an explicit existential interpretation suitable for masking, lifecycle-aware reasoning, and existence-aware learning settings.

2 Existence-Aware Temporal Graph Framework

Definition 2.1 (Vertex Lifespan). Let $\mathbb{T}$ be an ordered time domain. The lifespan of a vertex $v \in V$ is defined as a subset

$$L_V(v) \subseteq \mathbb{T},$$

represented as a finite union of closed intervals. A vertex v is said to be *alive* at time $t \in \mathbb{T}$ if $t \in L_V(v)$.

Similarly, the edge lifespan function

$$L_E : E \to \mathcal{P}(\mathbb{T})$$

assigns to each edge $e \in E$ the set of times during which the edge exists, given by

$$L_E(e) = \bigcup_{j=1}^{K_e} [t^s_{e,j},\, t^e_{e,j}],$$

where K_e denotes the number of temporal existence intervals associated with e. An edge e is present at time $t \in \mathbb{T}$ if and only if $t \in L_E(e)$.

It is important to distinguish between *existence* and *activity*. A vertex is said to be *dead* at time t if $t \notin L_V(v)$, in which case the vertex is structurally absent from the snapshot representation. By contrast, if $t \in L_V(v)$, the vertex is alive and may be either active or sleeping depending on its participation in interactions at time t.

More precisely, an alive vertex $v \in V(t)$ is called:

- *active* if it is incident to at least one edge in the snapshot at time t;
- *sleeping* if it is alive but has degree zero in the snapshot at time t.

Thus, the distinction between active and sleeping vertices is graph-theoretic, whereas the distinction between alive and dead vertices is existential.

For each $t \in \mathbb{T}$, the temporal snapshot induced by $\mathcal{G}$ is defined as

$$G(t) = (V(t), E(t)),$$

where

$$V(t) = \{v \in V \mid t \in L_V(v)\}, \qquad E(t) = \{e \in E \mid t \in L_E(e)\}.$$

Unlike classical temporal graph models, the snapshot $G(t)$ is not merely a restriction of a fixed graph to a subset of active edges. Rather, it is interpreted as a projection onto the living subgraph at time t, in which dead vertices are structurally excluded rather than retained as isolated nodes.

In general, it is not required that

$$E(t) \subseteq V(t) \times V(t).$$

This situation arises naturally in the presence of discrete or transit processes, where an interaction may persist over a time interval even if one of its associated vertices ceases to exist during part of that interval. Consequently, temporal snapshots need not correspond to classical graphs.

From an algebraic perspective, the absence of an edge between two alive vertices is represented by a zero value, whereas the absence of a vertex corresponds to a structurally masked or undefined entry in the snapshot representation.

Remark 2.2. To model temporal existence, we employ time intervals of the form $[t^s, t^e]$, where t^s and t^e denote the start and end times, respectively, with $t^s \leq t^e$. An entity is said to exist at all times $t \in \mathbb{T}$ such that $t^s \leq t \leq t^e$.

A key feature of the framework is that temporal existence need not be continuous. An entity may exist over one interval, disappear, and later reappear. Accordingly, temporal existence is represented as a finite union of intervals rather than a single contiguous interval.

Formally, the lifespan of an entity is represented as an element of $\mathcal{P}(\mathbb{T})$. This representation allows intermittent, cyclic, and irreversible existence patterns to be modeled within a unified framework.

Existence at time t is characterized by the condition $t \in L$, where L denotes the lifespan of the entity. If $t \notin L$, the entity is regarded as dead and is structurally excluded from the corresponding snapshot.

The notions of active and sleeping vertices are defined separately at the snapshot level. In particular, sleeping vertices remain existentially present but have degree zero in the corresponding temporal snapshot.

Consequently, the effective structure of the system varies over time according to the subset of vertices that are existentially valid at each time instant. This observation motivates the projection-based interpretation of temporal snapshots and the use of existence-aware masking mechanisms.

2.1 Existence-Aware Temporal Graph

Edges are treated in an equally explicit manner. Rather than being instantaneous relations, edges may possess their own temporal extent and internal dynamics. This distinction becomes essential when differentiating between types of flow. In the case of *continuous flow*, the existence of an edge is constrained by the simultaneous existence of both endpoint vertices. By contrast, in the case of *discrete or transit flow*, an edge may represent a process that, once initiated, persists independently of the continued existence of its source vertex. Such edges may therefore remain temporally active even when one of their associated vertices has ceased to exist.

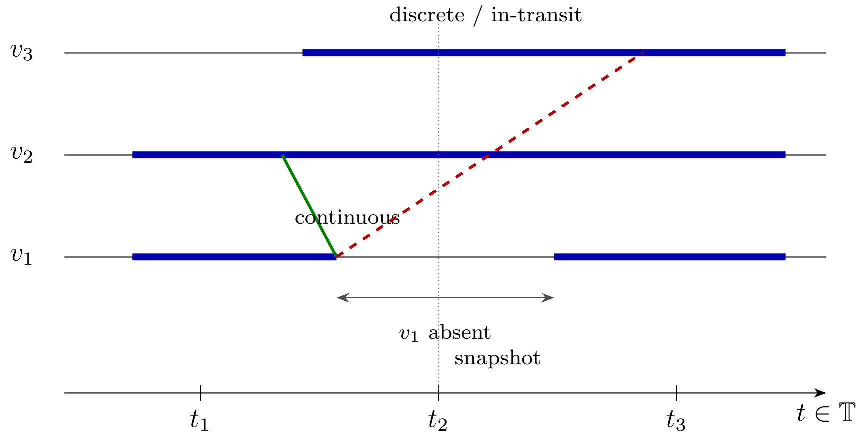

Figure 1: Existence-aware temporal graph with multi-interval vertex lifespans. Thick segments indicate alive intervals, while gaps correspond to absence. Solid edges represent continuous flows requiring both endpoints to be alive, whereas dashed edges represent discrete (in-transit) flows that may persist even when a vertex is absent. Consequently, a snapshot at t_2 need not correspond to a classical graph.

Consequently, an existence-aware temporal graph induces, at each time instant t, a snapshot consisting of all vertices that exist at t together with all edges that are active at t. Unlike classical temporal graph models, however, there is no requirement that every active edge in such a snapshot be incident to two existing vertices. In the presence of discrete flows, snapshots may contain edges whose source vertex is no longer present, implying that the resulting snapshot need not correspond to a classical graph in the standard graph-theoretic sense.

This behavior is illustrated conceptually in Figure 1. The figure demonstrates how vertices may exhibit multiple disjoint lifespan intervals and how edges with different flow characteristics interact with these lifespans over time. It also highlights that temporal snapshots may violate the classical incidence condition $E(t) \subseteq V(t) \times V(t)$, thereby motivating the need for an existence-aware formulation.

Overall, the existence-aware temporal graph generalizes conventional temporal graph models by explicitly allowing both vertex and edge existence to evolve over time. This formulation provides the structural foundation for distinguishing between different existential and behavioral states of vertices, enables a principled treatment of non-classical temporal snapshots, and prepares the ground for the algebraic and learning-based constructions developed in the subsequent sections.

2.2 Temporal Snapshots

Within the framework of existence-aware temporal graphs, the notion of a temporal snapshot arises naturally from the interaction between time and the lifespan intervals of vertices and edges. At a given time instant $t \in \mathbb{T}$, a snapshot is defined by restricting the graph to those entities whose lifespans contain t. In particular, a vertex or an edge is present at time t if and only if it exists at that time.

At first glance, this notion resembles the classical concept of temporal snapshots. However, a fundamental difference arises when vertex existence is treated explicitly. In standard temporal graph models, the vertex set is assumed to be fixed, and snapshots are obtained solely by activating or deactivating edges. In contrast, within the present framework, the vertex set itself is time-dependent. Consequently, snapshots at different times may involve different subsets of vertices and therefore operate on different effective structural supports.

Importantly, the absence of a vertex from a snapshot should not be conflated with inactivity. Rather, it reflects the structural non-existence of that vertex at the given time. This distinction plays a central role in the existence-aware formulation and underlies the projection-based interpretation of temporal snapshots developed throughout this work.

2.2.1 Flow-Dependent Snapshot Structure

This distinction becomes particularly significant when different types of flow along edges are taken into account. If all edges exhibit continuous flow, then edge existence requires the simultaneous presence of both endpoint vertices. In this case, each temporal snapshot coincides with a classical graph, since every active edge is incident to two existing vertices.

By contrast, in the presence of discrete or transit flows, this correspondence no longer holds. An edge may remain active over a time interval even though one of its associated vertices ceases to exist during part of that interval. Such behaviour naturally arises in applications such as transportation or communication systems, where a transfer process, once initiated, does not depend on the continued existence of its point of origin.

As a consequence, temporal snapshots induced by an existence-aware temporal graph need not correspond to classical graphs. There may exist time instants at which an edge is active while one of its associated vertices is absent, implying that the classical incidence condition

$$E(t) \subseteq V(t) \times V(t)$$

is violated. Temporal snapshots should therefore be interpreted as projections of the global graph structure onto a time-dependent existential subspace, rather than as ordinary induced subgraphs.

This perspective reveals that the collection of temporal snapshots generated by an existence-aware temporal graph possesses a richer algebraic structure than that assumed in conventional models. It also motivates the introduction of masking and projection mechanisms, which ensure that structural and computational operations are performed only on existentially valid components of the graph.

Example 1 (Classroom scenario: distinguishing existence, sleeping, and death).. Consider a classroom consisting of five entities: one teacher and four students, observed over the time window

$$\mathbb{T} = [08{:}00,\ 10{:}00].$$

Each entity is represented as a vertex in the temporal graph.

At time $t = 08{:}08$, the teacher interacts with one student; this interaction is modelled as a temporal edge active at that time. The remaining two students are physically present in the classroom but do not interact with anyone at $t = 08{:}08$. Hence, these vertices are *alive* but *sleeping* at that instant, as they exist but have zero degree in the corresponding snapshot.

Now suppose that a third student leaves the classroom during the interval

$$[08{:}16,\ 08{:}33].$$

Throughout this interval, the corresponding vertex does not exist and is therefore *dead* in the sense of LifeSage. Consequently, in the snapshot at $t = 08{:}20$ (within the absence interval), this vertex is not included in $V(t)$ and must be structurally excluded rather than treated as an isolated inactive node.

This example simultaneously illustrates the three fundamental vertex states in the LifeSage framework: (i) an *active* vertex (the interacting student), (ii) two *sleeping* vertices (present but non-interacting), and (iii) a *dead* vertex (structurally absent over a sub-interval of time). The key point is that LifeSage distinguishes between *lack of interaction* and *lack of existence*: sleeping vertices remain part of the snapshot vertex set, whereas dead vertices are removed from the structural support.

As illustrated in Fig. 2, this distinction is not made explicit in classical edge-centric temporal graph models. For additional practical and implementable examples, we refer the reader to the Appendix.

2.3 Continuous Flows

We begin by formalizing the notion of *continuous flows*, corresponding to interactions whose existence is inseparable from the simultaneous presence of their endpoint vertices. Continuous flows model instantaneous or persistent interactions that require both participating entities to exist throughout the duration of the interaction.

Let $e = (u, v)$ be an edge in an existence-aware temporal graph. The edge e is said to exhibit *continuous flow* if its activity at time t is constrained by the existential validity of both endpoints, that is,

$$\rho(e, t) = 1 \;\Rightarrow\; t \in L_V(u) \cap L_V(v),$$

where $\rho(e, t)$ denotes the activity status of e at time t.

Under continuous flow, an edge cannot be active unless both its source and target vertices are alive. Consequently, continuous flows preserve the classical incidence condition: whenever an edge is active, it connects two existing vertices. Temporal snapshots generated under purely continuous flows therefore satisfy the classical incidence condition, although the vertex set itself may vary over time.

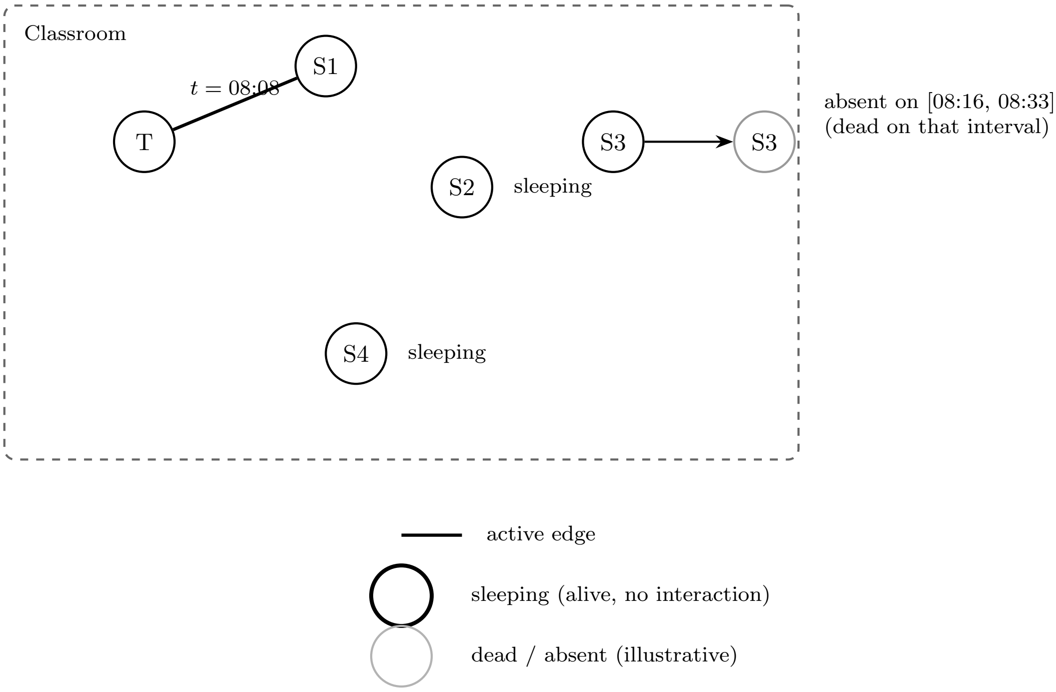

Figure 2: Illustration of active, sleeping, and dead vertex states in an existence-aware temporal graph. Sleeping vertices remain existentially present but non-interacting, whereas dead vertices are structurally absent from the snapshot.

Continuous flows arise naturally in settings where interactions are local, synchronous, or physically grounded, such as face-to-face communication, electrical connections, or instantaneous information exchange. In such cases, the disappearance of either endpoint immediately invalidates the interaction, reflecting a strict coupling between vertex existence and edge activity.

Thus, continuous flows represent the conservative extension of classical temporal graph models within the existence-aware framework. They serve as a baseline against which more general interaction mechanisms can be compared.

Lemma 2.3 (Edge–Vertex Lifespan Constraint for Continuous Flows). If an edge $e = (u, v)$ exhibits continuous flow, then

$$L_E(e) \subseteq L_V(u) \cap L_V(v).$$

2.4 Discrete Flows and Flying Edges

We now turn to *discrete flows*, which capture interaction processes whose temporal persistence is not strictly tied to the continued existence of their endpoint vertices. Discrete flows model transfers or processes that, once initiated, may evolve independently of the ongoing presence of their source or destination.

An edge $e = (u, v)$ is said to exhibit *discrete flow* if its activity over a time interval may extend beyond the lifespan of one or both of its endpoints. In particular, there may exist times $t \in \mathbb{T}$ such that

$$\rho(e, t) = 1 \quad \text{while} \quad t \notin L_V(u) \quad \text{or} \quad t \notin L_V(v).$$

Edges satisfying this property are referred to as *flying edges*, as they represent interactions that are temporally in transit rather than instantaneously anchored to both endpoints.

Lemma 2.4 (Initiation Constraint for Discrete-Flow Edges). Let $e = (u, v)$ be a discrete-flow edge. Then the initiation time of e must belong to the lifespan of its source vertex u. After initiation, the edge may remain active even if one or both endpoint vertices subsequently become non-existent.

The present framework models only the temporal persistence of such interactions at the structural level. Questions concerning buffering, completion, storage, or transmission failure are regarded as application-dependent semantics and lie outside the scope of the current work.

Discrete flows arise naturally in a wide range of real-world systems, including air transportation networks, communication systems, and supply chains. In such settings, once a transfer has been initiated, its continuation does not require the persistent existence of its point of origin.

The presence of discrete flows fundamentally alters the structure of temporal snapshots. In contrast to the continuous case, snapshots may contain active edges whose endpoints are not simultaneously present. As a result, the classical graph-theoretic incidence condition

$$E(t) \subseteq V(t) \times V(t)$$

may fail, and temporal snapshots need not correspond to classical graphs.

This phenomenon highlights a limitation of standard temporal graph models, which implicitly assume that all edges exhibit continuous flow. By accommodating discrete flows and flying edges, the existence-aware temporal graph framework can represent transit-type interactions that persist beyond simultaneous endpoint existence.

3 Generator Sets and Independence of Snapshots

A central consequence of existence-aware temporal graphs is that the sequence of temporal snapshots $\{G(t)\}_{t\in\mathbb{T}}$ should not be regarded as an arbitrary collection of independent observations. Rather, snapshots typically exhibit strong *structural dependencies* induced by (i) vertex existence constraints (Life Physics) and (ii) temporal transformations such as time-warping and time-compression, which are naturally captured by temporal (f, g)-homomorphisms.

This section formalizes two complementary notions: (i) a *generator set* of snapshots, representing a minimal collection of representative time instants whose snapshots can generate all others under admissible transformations; and (ii) *independence of snapshots*, which characterizes when two snapshots contain genuinely distinct information that cannot be derived from one another via such transformations.

3.1 Snapshot Space and Temporal Reparameterization

Let $\mathcal{G} = (V, E, L_V, L_E)$ be an existence-aware temporal graph defined over a time domain $\mathbb{T}$. For each $t \in \mathbb{T}$, the graph induces a temporal snapshot

$$G(t) = (V(t), E(t)).$$

In many situations, different snapshots may represent structurally similar configurations of the same evolving system. Such similarities may arise through vertex relabeling or through simple temporal reparameterizations such as time compression or dilation.

Consequently, temporal snapshots should not always be regarded as independent observations. Instead, many snapshots may correspond to structurally equivalent representations observed under different temporal parameterizations.

3.2 Generator Sets

In existence-aware temporal graphs, many temporal snapshots may remain structurally unchanged over extended time intervals. This observation motivates the notion of representative or generator snapshots.

Definition 3.1 (Generator Set of Snapshots). A subset of time instants $\mathcal{T}_0 \subseteq \mathbb{T}$ is called a generator set if every temporal snapshot of the graph is structurally equivalent to the snapshot at some time instant in $\mathcal{T}_0$.

Intuitively, a generator set contains representative time instants from which all distinct temporal configurations of the graph can be identified. Such representative snapshots typically arise near structural event times, including vertex appearance, disappearance, reappearance, or edge activation changes.

This perspective provides a compact description of temporal evolution by avoiding redundant repetitions of identical snapshot structures.

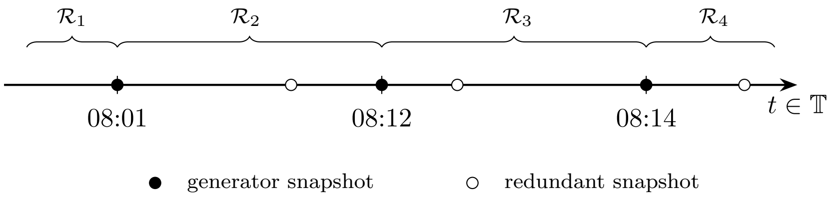

Figure 3: Generator snapshots in the classroom example. Event times partition the temporal domain into structural regimes $\mathcal{R}_i$ within which the induced snapshot remains unchanged. The times 08:01, 08:12, and 08:14 correspond to distinct regimes and therefore form a generator set.

3.3 Independence of Snapshots

While some temporal snapshots may be structurally repetitive, others represent genuinely distinct configurations of the system. This motivates the notion of snapshot independence.

Two snapshots are regarded as independent if they encode non-redundant structural information and cannot be obtained from one another through simple temporal reparameterizations or vertex relabelings.

Independence typically arises near structural event boundaries, including vertex appearance or disappearance, transitions between sleeping and active states, and changes in edge activity. Such events introduce qualitative modifications in the temporal graph structure and therefore require explicit representation in the snapshot stream.

3.4 Implications for Sampling and Learning

The preceding notions have direct implications for learning systems. First, uniform temporal sampling may be inefficient when extended intervals contain structurally repetitive snapshots. Second, the naive treatment of all missing interactions as inactivity becomes inconsistent in the presence of existential masking.

A more principled strategy is therefore to emphasize structurally distinct snapshots, particularly those occurring near existential transitions and discrete-flow events. This perspective motivates existence-aware sampling and masking strategies in downstream temporal learning settings.

4 Modeling Implications for Learning Systems

The LifeSage framework may provide a conceptual foundation for existence-aware temporal graph learning systems. Within this perspective, learning is influenced not only by temporal interaction dynamics, but also by the existential status of vertices over time. This viewpoint motivates modifications to several mechanisms commonly used in temporal graph learning architectures.

The first implication is the introduction of the *LifeMask* as an existence-aware filtering mechanism. At each time instant, the LifeMask identifies which vertices are existentially valid and therefore eligible to participate in message passing. In contrast to classical temporal graph models that implicitly assume persistent vertex existence, the LifeMask excludes dead vertices from interaction updates and message propagation.

A second implication concerns existence-aware asynchronous message passing. In this setting, message exchange depends not only on structural adjacency, but also on the existential validity of the participating vertices at the relevant time instant. Consequently, the distinction between inactivity and non-existence can be incorporated directly into the representation process.

A third implication concerns transit behavior and flying edges. In certain systems, an interaction may remain temporally active after one of its endpoint vertices has disappeared. The LifeSage framework provides a structural representation for such discrete-flow interactions, which may arise in transportation, communication, or transfer systems.

Collectively, these observations suggest that existence-aware temporal graph representations may be useful for modeling open and dynamically evolving systems in which entity appearance, disappearance, and inactivity play structurally distinct roles.

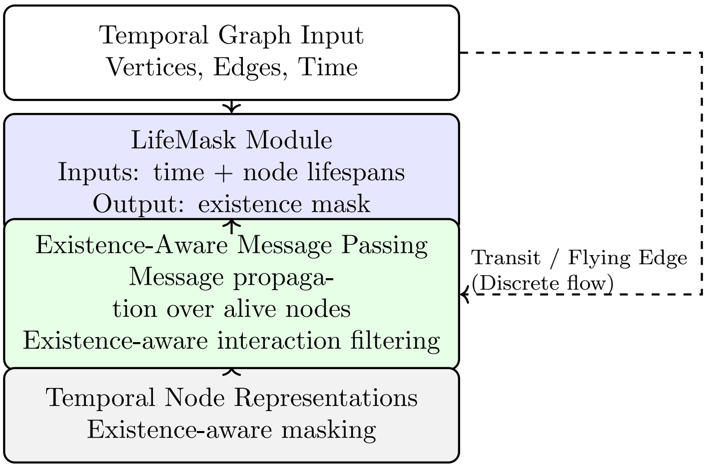

Figure 4: Conceptual overview of existence-aware masking and message propagation in the LifeSage framework. The LifeMask module filters vertices according to existential validity at each time step, while discrete or transit edges may persist independently of simultaneous endpoint existence.

4.1 Multi-Task Loss Design

Classical temporal graph learning objectives are often dominated by interaction prediction tasks and typically treat the absence of interactions as a homogeneous signal. Within an existence-aware framework, however, non-interaction may arise from different causes, including temporary inactivity or existential absence.

This observation motivates the consideration of multi-objective learning formulations in which interaction dynamics and existential states are modeled jointly. One possible component corresponds to temporal interaction prediction, while another may focus on predicting existential states such as active, sleeping, or absent vertices.

An additional optional objective may involve predicting future interaction times in continuous-time settings. Such formulations may encourage representations that capture both structural interaction patterns and entity life-cycle dynamics.

The discussion in this section is intended as a conceptual modeling perspective rather than a validated learning architecture. Experimental evaluation of such learning mechanisms remains an important direction for future work.

5 Discussion

Metrics such as AUC, average precision, and hit rate evaluate a model's ability to predict the presence or absence of edges at given time instants. However, they do not distinguish between cases in which an interaction fails to occur due to temporary inactivity and cases in which it is impossible due to the non-existence of one or more entities. As a result, a model may achieve strong link prediction performance while systematically misrepresenting the underlying structural dynamics of the system.

Existence-Aware Evaluation Criteria.. To address this limitation, we propose augmenting standard evaluation protocols with metrics that explicitly account for vertex life cycles. In particular, the following criteria are naturally supported by the LifeSage framework:

- **Survival Accuracy:** the accuracy with which the model predicts whether a vertex will transition into a permanently absent (dead) state within a future time horizon.
- **Rebirth Detection:** the ability of the model to correctly identify vertices that reappear after extended periods of absence, distinguishing temporary or cyclic disappearance from irreversible death.
- **State Classification Accuracy:** the performance in discriminating between existential states such as active, sleeping, and dead, particularly in regimes where interaction patterns alone are ambiguous.

These metrics directly assess whether the model has internalized the distinction between inactivity and non-existence, which constitutes the central conceptual contribution of existence-aware temporal graph learning.

Evaluation as Lifecycle Reasoning.. Under this perspective, model evaluation is no longer restricted to interaction prediction, but becomes a test of lifecycle reasoning. A successful model should not only predict which interactions are likely to occur, but also infer when entities are likely to disappear, reappear, or permanently exit the system. This shift aligns evaluation with real-world decision-making tasks such as failure prediction, churn analysis, and structural resilience assessment.

In this work, we focus on establishing the conceptual and architectural foundations required for such evaluation. A comprehensive empirical study, including benchmark datasets and quantitative comparisons with state-of-the-art temporal graph models, is left as an important direction for future research.

6 Conclusion

We introduced LifeSage, an existence-aware temporal graph framework that explicitly models the appearance, disappearance, and reappearance of entities over time. By relaxing the assumption of permanently existing vertices, LifeSage distinguishes structural non-existence from behavioral inactivity and provides an existence-aware perspective for representing dynamic systems with evolving entity sets.

The framework formalizes temporal lifespans as multi-interval subsets of time, leading to existence-masked snapshots that may deviate from classical graph structure, particularly in the presence of discrete or transit flows. We further discussed event times, snapshot equivalence, and representative generator snapshots arising from structural transitions in the graph.

Beyond its theoretical formulation, LifeSage motivates existence-aware learning settings in which masking, message passing, and supervision may account for entity life cycles. This perspective may support more faithful modeling of open and temporally evolving systems, and suggests several directions for future work, including algorithmic design, empirical evaluation, and integration with topological and spectral methods.

Appendix .1 Life Mask and Masked Snapshot Construction

Given an existence-aware temporal graph $\mathcal{G} = (V, E, L_V, L_E)$ and a time instant $t \in \mathbb{T}$, the instantaneous life mask and masked adjacency representation are constructed as follows.

Instantaneous Life Mask.. The life mask $\mathbf{M}^{(t)} \in \{0,1\}^{|V|}$ encodes existential validity:

$$M_i^{(t)} = \begin{cases} 1, & \text{if } t \in L_V(v_i), \\ 0, & \text{otherwise.} \end{cases}$$

Masked Adjacency Representation.. At the theoretical level, the temporal snapshot at time t is defined only on the set of alive vertices

$$V(t) = \{v_i \in V \mid M_i^{(t)} = 1\}.$$

Accordingly, dead vertices are structurally excluded from the snapshot graph.

For computational purposes, however, it is often convenient to work with a fixed-size matrix indexed by the global vertex set V. In this representation, entries associated with non-existing vertices are masked.

The masked adjacency matrix $\mathbf{A}^{(t)}$ is therefore defined entrywise by

$$A_{ij}^{(t)} = \begin{cases} w_{ij}(t), & \rho((i,j),t) = 1 \wedge M_i^{(t)} = 1 \wedge M_j^{(t)} = 1, \\ 0, & \rho((i,j),t) = 0 \wedge M_i^{(t)} = 1 \wedge M_j^{(t)} = 1, \\ \perp, & M_i^{(t)} = 0 \vee M_j^{(t)} = 0. \end{cases}$$

Here $\perp$ denotes a masked or undefined entry corresponding to the non-existence of at least one endpoint. In numerical implementations, $\perp$ may be represented by `NaN` or by an equivalent masking mechanism to prevent algebraic operations from propagating through non-existing vertices.

Computational Interpretation.. In practice, one first computes $\mathbf{M}^{(t)}$ from the vertex lifespans, then constructs the induced snapshot on the alive vertex set, and finally embeds this snapshot into a fixed-size masked representation when required for numerical or learning-based implementations.

Appendix A Computational Case Studies

We briefly describe three illustrative scenarios used to demonstrate the behavior of existence-aware temporal graphs under different life-cycle and flow conditions. These examples are not empirical evaluations; they are included solely to clarify the formal concepts introduced in the paper.

Appendix A.1 Classroom Scenario

A small interaction network represents a classroom with a teacher and several students. One student temporarily leaves the classroom and later re-enters, resulting in a vertex with a multi-interval lifespan. Snapshots taken during the absence correctly exclude the student from the masked representation, even though classical temporal models would treat the student as merely inactive.

This example illustrates the distinction between *sleeping* (alive but inactive) and *dead* (structurally absent) vertices.

Appendix A.2 Airport Logistics with Transit Edges

An airport network includes a flight that departs while both source and destination airports are alive, but whose destination ceases to exist before the flight completes. During the transit interval, the corresponding edge remains logically in progress while the destination vertex is dead.

In the masked snapshot representation, the destination vertex is structurally removed, and the edge is not incident to two existing vertices. This scenario illustrates discrete (or flying) flows and motivates the need for existential masking beyond classical graph incidence constraints.

References

[1] A. Barrat, M. Barthelemy, and A. Vespignani, *Dynamical Processes on Complex Networks*, Cambridge University Press, 2008.

[2] D.S. Bassett and O. Sporns, *Network neuroscience*, Nat. Neurosci. **20** (2017), 353–364.

[3] M. Batty, *The New Science of Cities*, MIT Press, 2013.

[4] Ma. Eshaghi and A. Jabbari, *On the algebraic and topological structure of temporal graphs: A unified framework for homomorphisms and spatio-temporal flows*, Int. J. Nonlinear Anal. Appl. In Press, doi: 10.22075/ijnaa.2025.10333.

[5] M.C. Gonzalez, C.A. Hidalgo, and A.L. Barabasi, *Understanding individual human mobility patterns*, Nature **453** (2008), 779–782.

[6] P. Holme, *Modern temporal network theory*, Eur. Phys. J. B **88** (2015), 234.

[7] P. Holme and J. Saramäki, *Temporal networks*, Phys. Rep. **519** (2012), 97–125.

[8] S.M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, *Representation learning for dynamic graphs*, J. Mach. Learn. Res. **21** (2020), 1–73.

[9] D. Kempe, J. Kleinberg, and A. Kumar, *Connectivity and inference problems for temporal networks*, J. Comput. Syst. Sci. **64** (2002), 820–842.

[10] Mikko Kivelä, A. Arenas, M. Barthelemy, J.P. Gleeson, Y. Moreno, M.A. Porter, *Multilayer networks*, J. Complex Netw. **2** (2014), 203–271.

[11] N. Masuda and P. Holme, *A Guide to Temporal Networks*, World Scientific, 2016.

[12] V. Nicosia, J. Tang, C. Mascolo, M. Musolesi, G. Russo, and V. Latora, *Graph metrics for temporal networks*, Temporal Networks, Springer, 2013, pp. 15–40.

[13] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, *Temporal graph networks for deep learning on dynamic graphs*, arXiv:2006.10637 (2020).

[14] M. Starnini, A. Baronchelli, A. Barrat, and R. Pastor-Satorras, *Random walks on temporal networks*, Phys. Rev. E **85** (2012), 056115.

[15] D.Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, *Inductive representation learning on temporal graphs*, Proc. Int. Conf. Learn. Represent. (ICLR), 2020.