

ContextGraph: Lifelog Intelligence Framework for Contextual Subgraph Evolution

Anil Sharma, Gunturi Venkata Sai Phani Kiran, Jayesh Rajkumar Vachhani, Sourabh Vasant Gothe, Ayon Chattopadhyay, Yashwant Saini, Parameswaranath Vadackupurath Mani, Barath Raj Kandur Raja

Samsung R&D Institute India-Bangalore (SRI-B)

{anil.sh, g.kiran, jay.vachhani, sourab.gothe, a.chattopadh, yash.saini, nath.vm, barathraj.kr}@samsung.com

Abstract

Lifelogging involves the continuous and comprehensive recording of a user’s daily activities, behaviors, and interactions, offering valuable insights for personalized healthcare, event retrieval, and lifestyle analysis. However, extracting meaningful patterns from lifelog data requires models to capture deeper temporal contexts beyond simple retrieval. To address this, we introduce ContextGraph, a lifelog intelligence framework that models lifelogs as a Temporal Knowledge Graph (TKG) to reason about the user’s evolving life patterns over time. ContextGraph computes Day Context Embeddings (DCE) to encode the temporal spread and social scene context of user’s daily behavior. Then a novel Lens module extracts semantically meaningful subgraph snapshots around an anchor node in the TKG, representing specific personal contexts in the user’s life. The Lens module also computes an evolution signature for each subgraph, indicating whether it is growing, decaying, or remaining static. By analyzing these evolution signatures, ContextGraph provides actionable insights into the user’s lifelogs such as stable routines, behavioral drifts, or lifestyle changes. Our experiments showcase DCE’s versatility, outperforming baselines in graph/node classification and reasoning on the Enzyme and DBLP datasets.

Project — <https://contextgraph-aaai26.github.io>

Introduction

Smartphones have evolved into powerful multimodal sensing platforms that continuously record a wide range of personal data, including location traces, activities, interactions, app-usage, photos, and more. This constant flow of multimodal data, often referred to as lifelogging (Ali et al. 2022), offers enormous potential for building intelligent personal assistants to provide deeper insights into our daily lives and behaviors.

The multimodal nature of smartphone data introduces several challenges. The data is inherently fragmented and heterogeneous, with sparse and structured elements that complicate unified analysis. Additionally, user preferences, which evolve dynamically through interactions and daily routines, further hinder consistent modeling and interpretation. Existing lifelogging systems (Tysse Hordvik et al.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

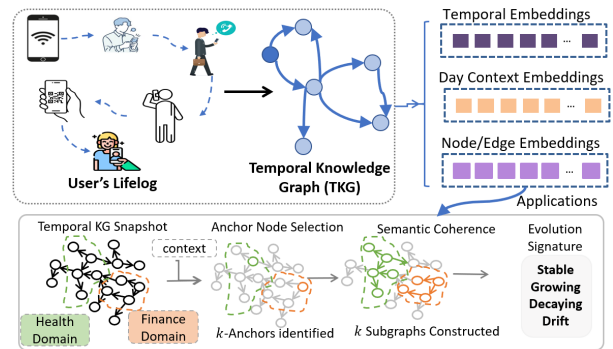


Figure 1: Illustration of user lifelogs modeled as a Temporal Knowledge Graph, where the proposed method learns day-contextual-embeddings by integrating both temporal and contextual information. These embeddings enable the detection of new routines, travel patterns, and emerging interests.

2024; Ho-Le et al. 2024; Spiess, Schuldt, and Rossetto 2024; Vuong et al. 2024) primarily emphasize data retrieval, allowing users to search past experiences using queries like “photos from the park” or “calls from last Monday”. While effective for memory recall, these systems lack the ability to reason about behavior evolution. For example, detecting gradual lifestyle changes, forming habits, or identifying emerging routines over time. Recent research has also proposed the construction of Personal Knowledge Graphs that model a user’s life through entities and relations derived from personal data (Balog and Kenter 2019; Skjæveland et al. 2024; Montoya et al. 2018). They rely heavily on third-party data sources such as email, cloud storage, shopping history, etc. which are not always accessible or available on-device. Moreover, they generally construct static graphs, failing to model the temporal dynamics essential for understanding user behavior over time.

We posit that true lifelog intelligence extends beyond data retrieval and necessitates the ability to infer, track, and explain changes in a user’s behavior by reasoning over structured and evolving data. To achieve this, we introduce ContextGraph, a framework based on Temporal Knowledge Graphs (TKGs). ContextGraph represents smartphone data as an evolving RDF graph. This TKG dynamically inte-

grates multimodal data, enabling inference across diverse data granularities and facilitating comprehensive contextual understanding.

For contextual understanding of the user’s daily dynamics, we propose DCE (Day Context Embeddings) leveraging the temporal dynamics and the contextual dynamics of the user’s interaction for the whole day. DCE excels in capturing the multifaceted dynamics of daily events, such as family gatherings, travel, or health-related activities. By integrating diverse data modalities, ranging from location check-ins and calendar entries to health metrics and media interactions, DCE constructs a holistic representation of each day. It models dynamics via a dual Variational Auto-Encoder (VAE) network over which is trained in self-supervised manner. However, overlapping events and ambiguous boundaries complicate isolating distinct life aspects. ContextGraph comprises a core module ‘Lens’, leveraging DCE to analyze daily TKG snapshots. Lens identifies anchor nodes signaling behavioral changes as sudden contextual shifts (e.g., new locations, payments, step spikes) i.e., the anchor node denotes the volatile information in the TKG snapshot. For each anchor, Lens extracts contextual subgraphs and monitors their evolution. The subgraph is meant to semantically capture a life-aspect and tracked sequentially. The Subgraph trajectories (growing, decaying, stable) reveal behavioral patterns. For example: Fig. 1 shows the user’s daily lifelog data, which is transformed into a TKG to learn DCE by integrating both temporal and contextual information. Then by focusing on evolving subgraph semantics, Lens module disambiguates overlapping events or life-aspects, providing actionable insights into daily life dynamics.

Our specific contributions are as follows:

1. We model smartphone data as a TKG, a time-aware graph structure that dynamically represents evolving entity relationships. We introduce DCE, novel Day Context Embeddings over TKG, to encode both temporal dynamics and contextual nuances using a dual VAE network.
2. We introduce Lens, a core module of our method designed to extract life-aspects from TKGs. It identifies anchor nodes and extracts subgraphs around them which are analyzed over time to find the evolution signature.
3. We define a novel dual similarity-based evolution signature of subgraphs to classify the changes as growth, decay, drift, or stable evolution.
4. We generate user behavior data using temporal point process to train and validated our framework. We also conducted a user trial to collect real smartphone data from 25 users across 4 countries. Our results demonstrate DCE’s and Lens’s ability to track evolving life-aspects.

Related Works

Personal Knowledge Graphs

Recent studies (Balog and Kenter 2019; Chakraborty and Sanyal 2023a; Skjæveland et al. 2024; Chakraborty and Sanyal 2023b; Montoya et al. 2018; Rastogi and Zaki 2020; Gyrard et al. 2018) focus on PKGs to represent users’ daily

life data in structured, time-independent forms. PKGs enhance personalization in applications like digital assistants, recommender systems, and semantic search. For example, Majumder et al. (Balog and Kenter 2019) integrate multimodal user data with open-domain knowledge graphs to improve entity understanding and query answering. Systems like (Montoya et al. 2018) model events and locations from tweets or mobile traces, while (Rastogi and Zaki 2020; Gyrard et al. 2018) infer health events from social or medical data. However, these works often rely on non-on-device data or fail to capture evolving user behavior. PKGs offer flexibility in modeling incomplete, evolving, and semantically linked information without rigid schemas. Most PKG works neglect temporal dynamics and device-level deployment due to reliance on external or cloud data. In contrast, our approach emphasizes time-aware modeling and on-device data, ensuring privacy, deployability, and interpretability. Our TKG reasons about temporal behavior and enables inferences like habit evolution, interest emergence, and lifestyle transitions.

Lifelog Search and Intelligence

The lifelog search challenge (Gurrin et al. 2024) has advanced systems for multimodal search and retrieval over personal data (Tysse Hordvik et al. 2024; Ho-Le et al. 2024; Spiess, Schuldt, and Rossetto 2024; Vuong et al. 2024; Le et al. 2024; Rossetto et al. 2024; Gago et al. 2024; Vopálková et al. 2024). Competitions like the Lifelog Search Challenge (LSC) have developed methods for retrieving lifelog moments using semantic queries based on activity, location, or objects. These systems use concept-based tagging, image embeddings, and rule-based query expansion for episodic retrieval. However, they lack mechanisms for longitudinal reasoning and temporal transitions, as noted in (Tran et al. 2025) survey. Even with temporal metadata, most systems treat data statically and fail to model evolving behavior. Our system, Lens, addresses this gap by identifying trigger points (anchors) in the TKG and analyzing subgraph evolution across time snapshots. These evolving subgraphs indicate stable routines, emerging interests, or decaying habits, enabling richer lifelog intelligence for questions like “Have my social interactions decreased?” or “Am I spending more time in unfamiliar locations?”

Reasoning over Knowledge Graphs

Research (Chen et al. 2022; Han et al. 2020; Christopoulos, Baltasou, and Tsihlias 2023; Li et al. 2021) focuses on identifying influential nodes in graphs and knowledge graphs using techniques like PageRank, Personalized PageRank, SimRank, MetaPath2Vec(Dong, Chawla, and Swami 2017), and Node2Vec2Rank(Mandros et al. 2024). These methods find hubs, bottlenecks, or semantically rich nodes for analysis triggers. However, traditional methods struggle with temporal dynamics in lifelogging, where events like new purchases or unfamiliar locations signify changes rather than prominence.

Subgraph extraction is also an active area, with methods like attention-guided expansion (Han et al. 2020), graph communities detected using fitness score (Christopoulos,

Baltsou, and Tsihlias 2023) and RDF similarity scoring (Le, Abel, and Gouspillou 2024; Maillot and Bobed 2018; Mandros et al. 2024) are also explored but lack mechanisms to track evolution signatures like growth or decay. Our approach includes context-aware subgraph expansion to monitor their evolution across daily TKG snapshots, enabling a reasoning module across behavioral domains like travel and health.

Classes	
:Person -- PhoneUser -- GalleryPerson (FaceID) -- ContactPerson	:Media -- Image -- Video -- Document
:Location (lat, lon, State, Country)	:DeviceInfo
:Scene	:MessageEvent
:AppUsage	:Date
:CallLogEvent	:ActivityEvent
:NotificationEvent	:ReminderEvent
:BatteryLevel	:WifiStateEvent
:BluetoothStateEvent	:SearchQuery

Figure 2: Outlines the TKG ontology, defining entities (e.g., Person, Location) for unified data representation.

Temporal KG Construction

Fragmented and heterogeneous smartphone data is unified into a TKG, an RDF-based structure with precise timestamps. The TKG integrates diverse data sources (sensors, apps, logs) into a coherent framework, linking entities (e.g., locations, persons) via meaningful relations (e.g., "appearsIn"). A key challenge is addressing data sparsity where user preferences and device settings lead to inconsistent data availability. For instance, we conducted a survey in February 2025 for 1 million Android devices which reveal that 78% of Android users globally keep their location settings disabled, necessitating lifelog intelligence systems to perform robust temporal reasoning even with sparse and user specific data.

Formally, we define the TKG as a set of reified quadruples (Benelhaj-Sghaier, Gillet, and Leclercq 2024) (where reifications introduces a statement node linking four elements (s, p, o, τ) to itself):

$$\mathcal{G} = \{(s, p, o, \tau) | s, o \in \mathcal{V}, p \in \mathcal{R}, \tau \in \mathcal{T}\} \quad (1)$$

where, \mathcal{V} represents the set of entities (s is the subject, o is object), \mathcal{R} is the set of predicate (p), \mathcal{T} is the set of timestamps, with τ indicating the temporal context of the triple. \mathcal{V} is composed of persistent nodes \mathcal{V}_p (e.g., Person, Location, contact, etc.) and \mathcal{V}_f transient/fading nodes (appear for a single instant like Image, CallLog, App-usageEvent, etc.) such that $\mathcal{V} = \{\mathcal{V}_p \cup \mathcal{V}_f\}$.

Entity, Predicate construction from Smartphone Data:

Building a TKG from smartphone data necessitates transforming raw, unstructured, and device-specific sensor outputs into semantically meaningful, time-annotated entities.

Unlike traditional knowledge graphs built from structured text or web corpora, smartphone data is inherently low-level and multimodal, encompassing images, text, locations, and app interactions which requires entity and relation extraction with temporal precision.

To create a TKG, we design a mapping ontology (shown in Fig. 2). Gallery images are processed on-device using a CLIP model (Radford et al. 2021) to detect scenes, faces and objects. Detected faces are assigned unique URIs and matched across images for consistency. Text from OCR and app logs is normalized, while location data is geocoded and timestamped. These processed entities are integrated into the TKG as RDF triples, with explicit typing and URIs.

Temporal Point Process (TPP)

Following the approach in (Trivedi et al. 2017), we utilize Temporal Point Processes (TPPs) to simulate user behavioral data across diverse events and modalities. The dataset is designed to encapsulate varied user preferences and contextual scenarios. For instance, a 'Morning Run' is modeled as a high-intensity burst of events in a short time window, whereas a 'Full Day Travel' activity is characterized by low-intensity events spread across the entire day.

To model the generation of our temporal data, we treat the arrival of each fact (edge) as an event governed by a TPP (Cox and Lewis 1972). We employ two distinct processes to simulate a wide range of user behavior. First, a non-homogeneous Poisson process where intensity of event arrival increases linearly with the time since the last event (t_{last}). This intensity function follows a Rayleigh distribution, is effective for simulating events that naturally grow and fade.

$$\lambda(t|t_{last}) = \alpha(t - t_{last}) \quad (2)$$

Here, α is a scaling parameter that controls the event's overall rate. In our context, an event is the creation of a new triple (s, p, o) associated with a particular modality.

Second, to capture bursty, self-reinforcing user behavior, we utilize Hawkes process (Hawkes 1971) where each event temporarily increases the intensity or likelihood of future events (like a self-excitation). The intensity function of Hawkes process is defined as:

$$\lambda(t) = \mu + \sum_{t_i < t} \alpha e^{-\beta(t-t_i)} \quad (3)$$

where μ is a baseline rate, α is excitation strength of past event t_i and β controls the decay rate of influence over time. This formulation captures both growth (via self-reinforcement) and decay (via exponential fading), mimicking real-world event surges and drop-offs.

Methodology

Our objective is to extract and track semantically meaningful subgraphs that represent evolving aspects of an individual's behavior over time from a TKG. In this section, we detail the DCE and Lens architecture separately and show the training procedure for each individual component.

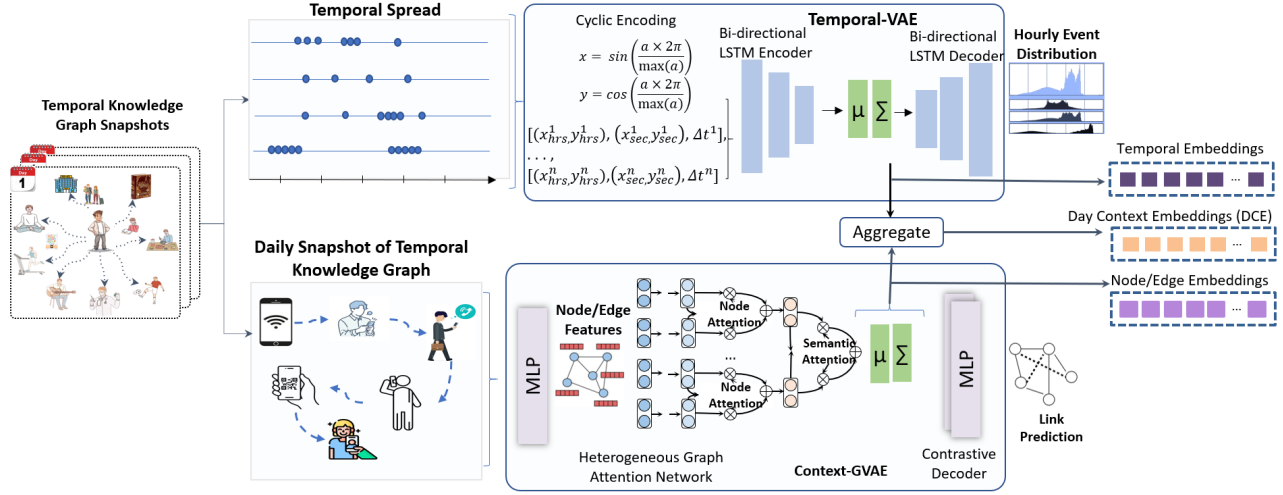


Figure 3: Shows the Day Context Embedding (DCE) module uses a dual VAE architecture, training two specialized models Temporal-VAE and Context-GVAE.

ContextGraph: Lifelog Intelligence Framework

We define the problem of lifelog intelligence as the task of identifying and tracking semantically meaningful, context specific subgraphs within the TKG that reflect key behavioral routines or transitions of the user. These subgraphs are rooted at anchor nodes, which act as focal points for understanding relevant behavior such as recurring app-usage, or changes in social interaction patterns.

To address this, we define a daily or fixed window snapshot $\mathcal{G}_d \subseteq \mathcal{G}$, representing the user’s semantic footprint on day t .

$$\mathcal{G}_d = \{(s, p, o) \mid (s, p, o, \tau) \in \mathcal{G} \wedge \text{day}(\tau) = d\} \quad (4)$$

Where τ is the timestamp of the event occurrence. We then compute the day dynamics representation using DCE and then extract a subgraph using Lens framework.

DCE: Day Context Embeddings To learn a comprehensive embedding for each daily snapshot, we employ a dual Variational Autoencoder (VAE) architecture. This approach, which we term the Day Context Embedding (DCE) module, is composed of two specialized models trained independently: a Temporal-VAE to encode the temporal rhythm of events, and a Context-GVAE to encode the relational graph structure and semantics (with whom, where, and what).

Existing graph embedding techniques like Node2Vec and DeepWalk are powerful tools for learning the structural topology of networks. However, when applied to user graphs, they only capture a structural snapshot of the graph while remaining entirely blind to the crucial temporal and contextual dimensions. The sequence, timing, and rhythm of events are fundamental aspects of human behavior that are lost in a purely structural representation. For example, a visit to a “restaurant” at 1 PM implies a different activity (lunch) than a visit at 8 PM (dinner).

DCE architecture is depicted in Fig. 3. It takes the day snapshot as input and return the day embeddings \mathbf{z}^d , node embeddings \mathbf{e}_v^d , and the edge/predicate embeddings \mathbf{e}_p^d as:

$$(\mathbf{z}^d, \{\mathbf{e}_v^d\}_{v \in V_p^d}, \{\mathbf{e}_p^d\}_{p \in E_d}) = f_{DCE}(\mathcal{G}_d, \theta) \quad (5)$$

where, $\mathbf{z}_d \in R^k$, $\mathbf{e}_v^d \in R^m$, $\mathbf{e}_p^d \in R^m$ (we use $k = 128$ and $m = 64$).

The Temporal-VAE is designed to compress a sequence of event timestamps into a latent vector that captures the day’s temporal signature. The input for a day d is a time-ordered sequence of events $T_d = (t_1, \dots, t_N)$. Each timestamp t_i is transformed into a 5-dimensional feature vector x_i incorporating cyclic encoding and the normalized time lag between events.

The core of the encoder is a bidirectional LSTM with a hidden dimension of 256, which processes the feature sequence to produce the parameters of a 64-dimensional latent distribution, μ_t and $\text{log}\sigma_t^2$. The model is trained using Adam optimizer to minimize the evidence lower bound:

$$\mathcal{L}_{temporal} = E_{q(z|x)}[\text{log}p(y|z)] - \beta \cdot D_{KL}(q(z|x) \parallel p(z)) \quad (6)$$

Here, the term $\text{log}(y|z)$ is the log-likelihood of the true hourly event histogram y given the latent sample z , measured via a KL divergence loss. The second term regularizes the latent space, with $\beta = 0.5$.

Concurrently, the Context-GVAE (Context Graph VAE) learns a latent representation of the day’s complex, heterogeneous graph structure. The model’s encoder is built upon the Heterogeneous Graph Attention Network (HAN (Wang et al. 2021)). Initially, a dictionary of linear layers projects the 32-dimensional input features of each node type into a common 64-dimensional space. Subsequently, two layers of HANConv, each with 4 attention heads, are applied to learn rich, context-aware node embeddings. These are aggregated via mean pooling to produce a single graph-level vector, which is then mapped to the 32-dimensional latent distribution parameters, μ_g and $\text{log}\sigma_g^2$. The model is trained using Adam optimizer on a link prediction task. The reconstruction loss is a max-margin objective that maximizes the score

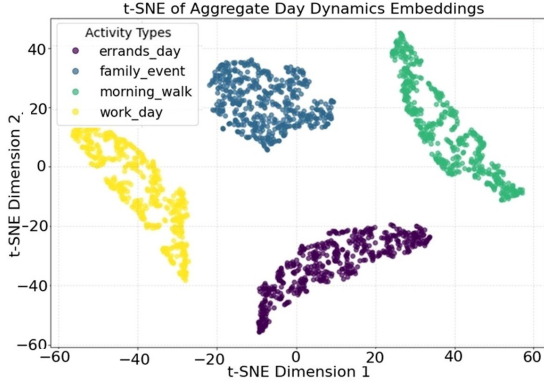


Figure 4: t-SNE embeddings of DCE (aggregate) on events generated using TPP.

of true edges (u,v) over negatively sampled ones (u,v') :

$$\mathcal{L}_{recon} = - \sum_{(u,v) \in E} \log(\sigma(\text{score}(u,v) - \text{score}(u,v'))) \quad (7)$$

where the score is the dot product of node embeddings. This is combined with standard KL divergence regularizer to form the total loss for the context-GVAE model, with $\beta = 4$.

$$\mathcal{L}_{context} = \mathcal{L}_{recon} + \beta \cdot D_{KL}(q(z|\mathcal{G}_d)||p(z)) \quad (8)$$

Fused embeddings from both VAEs form DCE (of 128 dimension) representing day dynamics in a latent vector.

Lens Module: Tracking Anchor Driven Subgraphs The Lens framework is depicted in Fig. 5. The lens framework trigger when an anchor node is identified and then a subgraph is expanded around the anchor.

Context-Aware Anchor Node Selection In the TKG snapshot, we define a context input, which is a curated list of domains (e.g., health, travel, work), is used as input to find important persistent nodes that align with the context domain. From these filtered nodes only, an anchor is identified. If context is empty, then all persistent nodes are candidate anchors.

Filtered nodes exhibiting abrupt change from previous TKG snapshot may depict an important or new event. For example, a sudden payment pattern may signify new travel/event bookings. The final anchor node is selected as:

$$a_v(v) = \{v | \frac{\|(e_v^d - e_v^{d-1})\|_2}{\|(e_v^{d-1})\|_2} > \tau_v \wedge v \in \mathcal{V}_p\} \quad (9)$$

where v is a persistent node and $\tau_v = 0.8$ is a threshold. Nodes with top-k scores are selected as anchors for subgraph construction.

Subgraph Expansion and Pruning From each anchor node a , we expand a subgraph around it by traversing its immediate neighborhood i.e., RDF edges (a,p,o) . For each triple, we compute an attention score:

$$\kappa(a,p,o) = \cos(e_a + e_p, e_o) \quad (10)$$

Here, e_i is the DCE node and edge embedding. This attention score depicts how contextually aligned a neighbor node

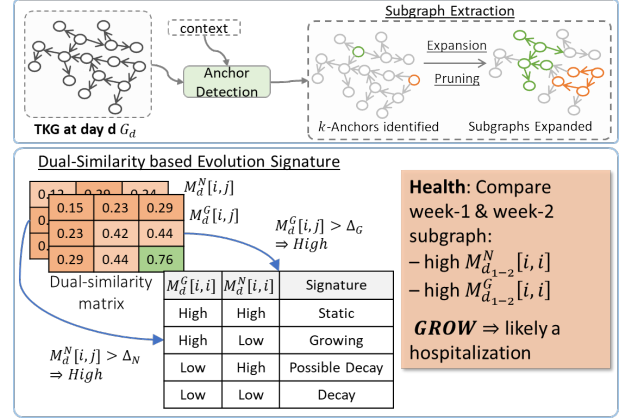


Figure 5: Lens Architecture: from a TKG snapshot, finds a volatile anchor and then expands a subgraph. A dual-similarity matrices finds evolution signature.

o is with the anchor a along relation p . We include o in the subgraph if $\kappa(a,p,o) \geq \tau_s$, where $\tau_s = 0.7$ is a threshold. The process continues up to k -hops or until no further nodes pass the threshold.

Temporal Evolution Signature

For i^{th} subgraph $S_i^{d_1}$ extracted at day d_1 , we track its evolution in future (say d_2) as snapshot $S_i^{d_2}$. We compute two similarity matrices:

1. Graph-level similarity matrices:

$$M_{d_1-2}^G[i,j] \forall i \in S_i^{d_1}, \forall j \in S_i^{d_2} \text{ using DCE embeddings } \mathbf{z}^{d_1}(S_i^{d_1}) \text{ (for } S_i^{d_1}):$$

$$M_{d_1-2}^G[i,j] = \cos(S_i^{d_1}, S_j^{d_2}) \quad (11)$$

The similarity matrices $M_{d_1-2}^G[i,j]$ is the cosine similarity between i^{th} subgraph $S_i^{d_1}$ at d_1 and j^{th} subgraph $S_i^{d_1}$ at d_2 . If it signifies mild change then Node-level similarity will provide more confidence to find whether the subgraph has decayed.

2. Node-level similarity score:

$$M_{d_1-2}^N[i,j] \forall i \in S_i^{d_1}, \forall j \in S_i^{d_2} \text{ using DCE node embeddings } (e_v^{d_1}[i] \forall v \in S_i^{d_1}):$$

$$M_{d_1-2}^N[i,j] = \frac{\sum_{v \in S_i^{d_1}} \cos(e_v^{d_1}[i], e_v^{d_2}[j])}{\#Nodes(S_i^{d_1})} \quad (12)$$

The evolution signature ϵ_{t_1-2} is identified by classifying each pair using node-level threshold $\theta_N (= 0.8)$ and subgraph-level threshold $\theta_G (= 0.7)$:

1. Static: $M_G^{d_1-2}[i,i] > \theta_G, M_N^{d_1-2}[i,i] > \theta_N$
2. Growing: $M_G^{d_1-2}[i,i] > \theta_G, M_N^{d_1-2}[i,i] \leq \theta_N$
3. Possible decay: $M_G^{d_1-2}[i,i] \leq \theta_G, M_N^{d_1-2}[i,i] > \theta_N$
4. Decay: $M_G^{d_1-2}[i,i] \leq \theta_G, M_N^{d_1-2}[i,i] \leq \theta_N$
5. Drift (i,j): $M_G^{d_1-2}[i,j] > \theta_G, M_N^{d_1-2}[i,j] > \theta_N$

ϵ_t captures whether the structure (Graph-level), values (Node-level), or both have changed, allowing us to reason about behavioral stability or change. We then aggregate these signatures to detect drift, lifestyle change, or routines.

Experiments and Results

To evaluate ContextGraph’s DCE and Lens modules, we conduct experiments assessing embedding quality and downstream task performance (e.g., classification, subgraph analysis) on public datasets. Baselines include established methods. We performed extensive ablation experiments to find best parameters and component architecture and only important experiments are included in the main paper.

Experimental Setup

Datasets: To comprehensively train and evaluate our framework, we first generate a large-scale TPP dataset that mirrors the complex, multi-modal nature of smartphone data as in (Trivedi et al. 2017). The dataset is designed to capture a diverse range of user activities, profiles, and contextual situations. The arrival of events within each activity is modeled using a Temporal Point Process (TPP). Additionally, we utilize real smartphone data from 25 users across 4 countries, encompassing over 1.36 million multimodal interactions per user, spanning at least one year. We use these datasets to validate the DCE combined with Lens framework to infer the changing life-aspects. We use DCE (pre-trained on TPP dataset) to generate embeddings for two distinct case studies on the real smartphone data. We then construct a TKG from these dataset which links each modality as RDF triples. For comparing embedding performance alone, we use two public datasets Enzyme dataset (Rossi and Ahmed 2015) containing 600 graphs for Graph classification [homogenous graphs and no time information] and DBLP dataset, a large heterogenous open-source bibliographic dataset (Yang and Leskovec 2012) having 317080 nodes for node classification (predicting research area of authors represented by nodes).

Evaluation metrics and performance comparison: We validate DCE on both TPP and real smartphone TKGs. For classification, we use precision and recall, and F1-score metrics. We compare performance against a set of well established methods like DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), Node2Vec (Grover and Leskovec 2016) (for homogenous graphs), HAN (Wang et al. 2021) for heterogenous Graph based node classification, PELT (Liu et al. 2025) for frequency based change point detection. For Node2vec and DeepWalk, which produce node-level embeddings, we derive a single graph-level embedding by taking the mean of all node embeddings within the graph. For PELT, we compute hour event frequency of each modality which serves as features for classification.

DCE Performance Evaluation

The Temporal-VAE and Context-GVAE are trained independently on the TPP dataset, empirically yielding more stable and interpretable latent spaces. Each VAE minimizes reconstruction error and KL divergence from a standard normal prior. We perform two experiment to validate the generated embeddings.

Embedding Space Analysis First, to qualitatively and quantitatively assess the learned representations, we analyze the structure of the embedding space using t-SNE plots. We generated several single day events using TPP and with perturbed dynamics. For instance, if an evening run’s temporal pattern transitions to resemble a home-based activity (e.g., steps taken at home). We visualize the embeddings using t-SNE to provide a qualitative understanding of the cluster separation. Fig. 6 shows the temporal, context, aggregate of both (DCE) embedding spaces, which clearly demonstrates the contribution of each component and the model’s handling of complex, multi-activity days. The evening run and family event which has same temporal dynamics are now separated in the aggregate DCE embeddings after context information is aggregated, similarly work-solo and work-day are also separated. Furthermore, A high-quality embedding space should exhibit low intra-class distance (samples of the same activity are close) and high inter-class distance (samples of different activities are far apart) which is clear from the plots that DCE effectively represents the day dynamics. We also compute the average inter-class and intra-class Euclidean distances averaged for the specific events generated in TPP-derived data as shown in Table 2.

Performance Comparison on Downstream Tasks To rigorously evaluate the effectiveness of DCE, we perform classification on three distinct test sets, first, Out-of-Distribution (OOD) TPP Data: A completely new, unseen TPP-simulated dataset to test generalization. Second, on public Graph Classification dataset (Rossi and Ahmed 2015) to isolate the contribution of the aggregate graph context (Context-GVAE). This dataset, originally designed for molecular graph classification, challenges the model to infer structural properties independent of temporal data. Third, we leverage a public heterogeneous Graph dataset, DBLP (Yang and Leskovec 2012). The task is to predict the research of any user in the graph.

We assess our proposed embeddings against state-of-the-art methods and a frequency-based baseline (PELT) across three datasets for classification tasks. We also include a frequency based method (PELT) to show that the efficacy of change-point detection method on the classification task. For classification, we train a Random Forest classifier on the embeddings (features for PELT) from each method and report standard classification metrics. Table 1 shows the performance comparison and highlights the efficacy of DCE embeddings across both the TPP-derived and public datasets, demonstrating their effectiveness in capturing complex temporal patterns. Node2Vec and DeepWalk works for homogeneous graphs only, whereas DCE works for both. HAN is a supervised training method which requires retraining on each graph. We also compare the runtime efficiency of DCE with Node2Vec and DeepWalk on a public dataset. On average for a single graph, Node2Vec/DeepWalk took 100 seconds (1 min 40 sec) whereas DCE (aggregate) took only 8 seconds. This demonstrates DCE’s significant speed advantage, reducing embedding computation time by 87.5% compared to Node2Vec methods.

The results strongly indicate the superiority of the pro-

Embedding Type	ODD TPP-derived			Enzyme Dataset			DBLP Dataset		
	Precision	Recall	F1	Precision	Recall	F1	Macro-P.	Macro-R.	Macro-F1
DeepWalk	0.538	0.533	0.534	0.220	0.217	0.213	-	-	-
Node2Vec	0.588	0.578	0.562	0.186	0.183	0.182	-	-	-
PELT	0.384	0.578	0.461	-	-	-	0.28	0.34	0.30
HAN	0.286	0.391	0.341	0.381	0.292	0.242	0.533	0.512	0.452
Temporal-VAE	0.733	0.733	0.733	-	-	-	0.89	0.91	0.90
Context-GVAE	0.447	0.444	0.446	0.443	0.450	0.444	0.70	0.68	0.69
Aggregate (DCE)	0.713	0.711	0.708	-	-	-	0.88	0.90	0.89

Table 1: Performance comparison of our proposed DCE with state-of-the-art methods on TPP-derived and 2 public datasets.

Embedding Type	Intra Class	Inter Class
Temporal-VAE (Avg)	0.031	1.130
Context-VGAE (Avg)	0.001	0.110
DCE-Agg (Avg)	0.056	1.144

Table 2: Average inter and intra class distance between 6 simulated activity classes across 2 years TPP data.

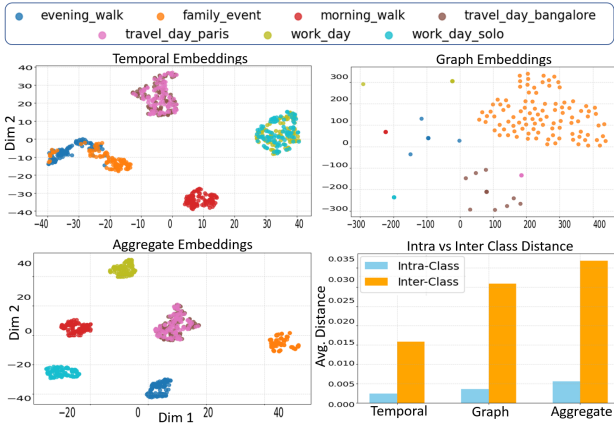


Figure 6: t-SNE plots of the generated embeddings on TPP data events with perturbed dynamics.

posed DCE framework. The distance metrics and t-SNE plots reveal that our aggregate embeddings create well-defined, separable clusters for distinct activities. Most importantly, the downstream classification task demonstrates that DCE (Aggregate) consistently outperforms all baseline methods, on both simulated and public datasets. The training was performed once on the very large TPP-derived data which also highlights the inductive nature of DCE embeddings. The purely temporal and frequency based method (PELT) fail to capture the critical contextual aspect. Whereas, node2vec and other graph embedding based methods also fail to capture the temporal dynamics of the data. By effectively encoding both the temporal (when) and context (whom, where and what) aspect of the data, DCE provides a more holistic and powerful representation leading to significant improved performance on smartphone data.

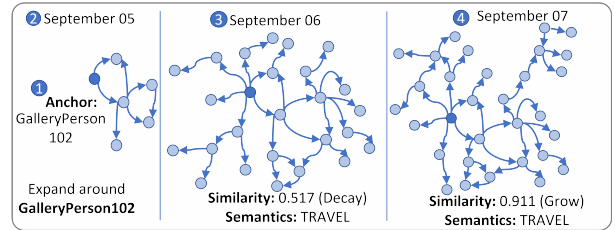


Figure 7: Shows Anchor guided subgraph extraction. Anchor node GalleryPerson102 (highlighted) triggers subgraph extraction on September 5. Subgraphs around same anchor are detected on September 6 and 7, demonstrating temporal inconsistency on 6th signalling change in routine.

Lens Module Performance Evaluation

We employ the Lens module to identify life-aspects (e.g., daily routines, social interactions) as dynamically evolving subgraphs derived from daily TKG snapshots. For a given input context, an anchor node is selected to extract a relevant subgraph, whose evolution is analyzed by identifying its temporal signature over time. Figure 7 illustrates an extracted subgraph using a GalleryPerson as the anchor node. The subgraph evolves based on user activities; for instance, it decayed on September 06 and subsequently grew. We analyze the evolution signatures (as growing or decaying) of these subgraphs as the subgraph is evolving, capturing patterns in daily life dynamics.

Conclusion

We introduced ContextGraph, a framework that models smartphone multimodal lifelog data as Temporal Knowledge Graphs (TKGs). The framework produces Day Context Embeddings (DCE), latent vectors that encode daily temporal dynamics via a Temporal-VAE and social contexts (e.g., with whom, where and what) through a Context-GraphVAE. Our experiments showcase DCE’s versatility, outperforming baselines in graph/node classification and reasoning on the Enzyme and DBLP datasets. Additionally, the Lens module leverages DCE to identify semantically meaningful subgraphs anchored to specific nodes, analyzing their evolution (growth, decay, or stability) through temporal signatures. Our results validate DCE’s accuracy in day pattern encoding and Lens’s capability to derive insights from real-world smartphone data.

References

- Ali, S.; Khusro, S.; Khan, A.; and Khan, H. 2022. *Smartphone-Based Lifelogging: Toward Realization of Personal Big Data*, 249–309. Cham: Springer International Publishing. ISBN 978-3-030-75123-4.
- Balog, K.; and Kenter, T. 2019. Personal knowledge graphs: A research agenda. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, 217–220.
- Benelhaj-Sghaier, S.; Gillet, A.; and Leclercq, É. 2024. Knowledge Graph Multilevel Abstraction: A Property Graph Reification Based Approach. In *International Conference on Research Challenges in Information Science*, 12–19. Springer.
- Chakraborty, P.; and Sanyal, D. K. 2023a. A comprehensive survey of personal knowledge graphs. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(6): e1513.
- Chakraborty, P.; and Sanyal, D. K. 2023b. A Personal Knowledge Graph for Researchers. In *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*, 129–132.
- Chen, Z.; Zhao, X.; Liao, J.; Li, X.; and Kanoulas, E. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowledge-based systems*, 251: 109134.
- Christopoulos, K.; Baltsoy, G.; and Tsihlias, K. 2023. Local community detection in graph streams with anchors. *Information*, 14(6): 332.
- Cox, D. R.; and Lewis, P. A. W. 1972. Multivariate point processes. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, volume 3, 401–448. University of California Press Berkeley.
- Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 135–144.
- Gago, A.; Kaluza, B.; Bartolomeu, E.; Pandeirada, J.; Ribeiro, R.; and Neves, A. J. 2024. MEMORIA: A Memory Enhancement and MOment Retrieval Application at the LSC2024. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge*, 99–104.
- Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.
- Gurrin, C.; Zhou, L.; Healy, G.; Werner, B.; Dang-Nguyen, D.-T.; Hodges, S.; Ór Jónsson, B.; Lokoč, J.; Rossetto, L.; Tran, M.-T.; and Schöffmann, K. 2024. Introduction to the Seventh Annual Lifelog Search Challenge, LSC’24. ACM.
- Gyrard, A.; Gaur, M.; Shekarpour, S.; Thirunarayan, K.; and Sheth, A. 2018. Personalized health knowledge graph. In *CEUR workshop proceedings*, volume 2317, 5.
- Han, Z.; Chen, P.; Ma, Y.; and Tresp, V. 2020. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International conference on learning representations*.
- Hawkes, A. G. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1): 83–90.
- Ho-Le, M.-Q.; Do-Huu, H.-H.; Ho, D.-K.; Le-Hinh, N.-T.; Vo-Hoang, H.-V.; Ninh, V.-T.; and Tran, M.-T. 2024. SnapSeek: An Interactive Lifelog Acquisition System for LSC’24. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge*, 24–29.
- Le, H.-B.; Nguyen, T.-N.; Le, T.-K.; Tran, M.-T.; Nguyen, T.-B.; Ninh, V.-T.; Zhou, L.; and Gurrin, C. 2024. LifeSeeker 6.0: Leveraging the linguistic aspect of the lifelog system in LSC’24. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge*, 53–57.
- Le, N. L.; Abel, M.-H.; and Gouspillou, P. 2024. Exploring Weighted Property Approaches for RDF Graph Similarity Measure. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 3104–3109. IEEE.
- Li, Z.; Jin, X.; Li, W.; Guan, S.; Guo, J.; Shen, H.; Wang, Y.; and Cheng, X. 2021. Temporal knowledge graph reasoning based on evolutionary representation learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 408–417.
- Liu, S.; Jin, J.; Liu, S.; Li, T.; Liu, C.; Zhao, Y.; Li, Y.; and Song, N. 2025. Change Point Detection For FOGs Scale Factor Test Data Using Pruned Exact Linear Time (PELT) Algorithm. *IEEE Transactions on Instrumentation and Measurement*.
- Maillot, P.; and Bobed, C. 2018. Measuring structural similarity between RDF graphs. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC ’18, 1960–1967*. New York, NY, USA: Association for Computing Machinery. ISBN 9781450351911.
- Mandros, P.; Gallagher, I.; Fanfani, V.; Chen, C.; Fischer, J.; Ismail, A.; Hsu, L.; Saha, E.; DeConti, D. K.; and Quackenbush, J. 2024. node2vec2rank: Large Scale and Stable Graph Differential Analysis via Multi-Layer Node Embeddings and Ranking. *bioRxiv*.
- Montoya, D.; Tanon, T. P.; Abiteboul, S.; Senellart, P.; and Suchanek, F. M. 2018. A knowledge base for personal information management. In *LDOW2018-Linked Open Data Workshop at the World Wide Web Conference*.
- Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’14*, 701–710. New York, NY, USA: Association for Computing Machinery. ISBN 9781450329569.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*.
- Rastogi, N.; and Zaki, M. J. 2020. Personal health knowledge graphs for patients. *arXiv preprint arXiv:2004.00071*.
- Rossetto, L.; Kyriakou, A.; Lange, S.; Ruosch, F.; and Wang, R. 2024. LifeGraph 4-lifelog retrieval using multimodal

knowledge graphs and vision-language models. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge*, 88–92.

Rossi, R. A.; and Ahmed, N. K. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*.

Skjæveland, M. G.; Balog, K.; Bernard, N.; Łajewska, W.; and Linjordet, T. 2024. An ecosystem for personal knowledge graphs: A survey and research roadmap. *AI Open*, 5: 55–69.

Spiess, F.; Schuldt, H.; and Rossetto, L. 2024. Spatiotemporal Lifelog Analytics in Virtual Reality with vitrivr-VR. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge*, 7–11.

Tran, A.; Bailer, W.; Dang-Nguyen, D.-T.; Healy, G.; Hodges, S.; Jónsson, B. .; Rossetto, L.; Schoeffmann, K.; Tran, M.-T.; Vadicamo, L.; et al. 2025. The State-of-the-Art in Lifelog Retrieval: A Review of Progress at the ACM Lifelog Search Challenge Workshop 2022-24. *arXiv preprint arXiv:2506.06743*.

Trivedi, R.; Dai, H.; Wang, Y.; and Song, L. 2017. Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, 3462–3471. JMLR.org.

Tysse Hordvik, M.; Østby, J. T.; Dang Nguyen, D. T.; Kesavulu, M.; Nguyen, T.-N.; Le, T.-K.; Gurrin, C.; and Tran, M.-T. 2024. LifeLens 2.0: Improving Efficiency and Usability in Lifelog Retrieval Systems through UX/UI Design. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge*, 41–46.

Vopálková, Z.; Yaghob, J.; Stroh, M.; Schlegel, U.; and Lokoc, J. 2024. Searching temporally distant activities in lifelog data with prak tool V2. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge*, 111–116.

Vuong, G. H.; Ho, V.-S.; Nguyen Dang, T. T.; Thai, X.-D.; Nguyen-Ho, T.-L.; Pham, M.-K.; Le, T.-K.; Ninh, V.-T.; Healy, G.; Gurrin, C.; et al. 2024. LifeInsight2. 0: An Enhanced Approach for Automated Lifelog Retrieval in LSC'24. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge*, 1–6.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Cui, P.; Yu, P.; and Ye, Y. 2021. Heterogeneous Graph Attention Network. *arXiv:1903.07293*.

Yang, J.; and Leskovec, J. 2012. Defining and Evaluating Network Communities based on Ground-truth. *arXiv:1205.6233*.