EMPOWERING MEMORY ASSISTANCE: AN EPISODIC MEMORY-BASED FRAMEWORK FOR PERSONALIZED RECOMMENDATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Artificial agents operating in dynamic environments require the ability to recall and contextualize past experiences to inform future behavior. Drawing inspiration from human episodic memory, we propose a cognitively grounded recommendation framework that models time-evolving personal experiences using a dynamic, multimodal memory architecture. Our system encodes temporally structured actions, places, and interactions into a hierarchical temporal graph network (TGN), enabling agents to disambiguate overlapping behavior patterns and anticipate future actions based on long-term experience. Unlike traditional recommendation or forecasting models that rely on static, task-specific patterns, our approach supports continual memory updates without retraining, and generalizes across varied activity sequences. Evaluated on a structured dataset derived from three years of egocentric recordings, our model significantly outperforms state-of-the-art baselines (e.g., AntGPT, DyRep, Palm) on next-activity prediction and sequence alignment metrics. This work introduces a scalable, cognitively inspired memory architecture with broad applications in lifelong learning, assistive robotics, and human-AI collaboration.

1 INTRODUCTION

Episodic memory underpins human cognition, enabling recall of personal experiences contextualized by time and place Tulving (2002). This capacity not only supports remembering the past but also underlies planning and anticipation of future actions—a capability artificial agents must emulate to operate effectively in dynamic environments. Applications such as personalized assistants, autonomous systems, and lifelong learning agents increasingly demand models that capture long-term behavior, adapt to evolving contexts, and make time-sensitive predictions. However, current large-scale models such as Palm (Chowdhery, Narang, and Devlin, 2022) and AntGPT (Zhao, Wang, Zhang, Fu, Do, Agarwal, Lee, and Sun, 2024) remain limited: they require extensive retraining for new contexts, lack continuity of experience, and struggle with temporal-semantic dependencies across multiple timescales.

We propose a cognitively inspired, memory-augmented framework that leverages Temporal Graph Networks (TGNs) (Rossi, Chamberlain, Frasca, Eynard, Monti, and Bronstein, 2020) to encode episodic experiences over extended horizons. Our architecture integrates multimodal observations (audio, visual, and linguistic cues) into a dynamic memory graph that evolves over time. Time-aware embeddings and day-wise memory organization enable recognition of recurring patterns, disambiguation of overlapping behaviors, and long-horizon prediction without retraining.

Key Challenges Addressed. Our framework addresses limitations in prior systems by: (i) supporting dynamic, lifelong adaptation to behavioral context shifts; (ii) integrating multimodal signals for richer, grounded representations; (iii) enabling temporal disambiguation through structured memory recall; (iv) generalizing across both short- and long-horizon activity prediction tasks.

Contributions. We introduce an episodic memory-based framework for temporal reasoning and adaptive behavior modeling that: (i) encodes and organizes multimodal memory using TGNs with time-aware abstraction; (ii) achieves state-of-the-art performance on structured activity datasets (e.g., Ego4D (Grauman, Westbury, Byrne, and Chavis, 2022; Zhou, Cao, Zheng, Zheng, and Liu,

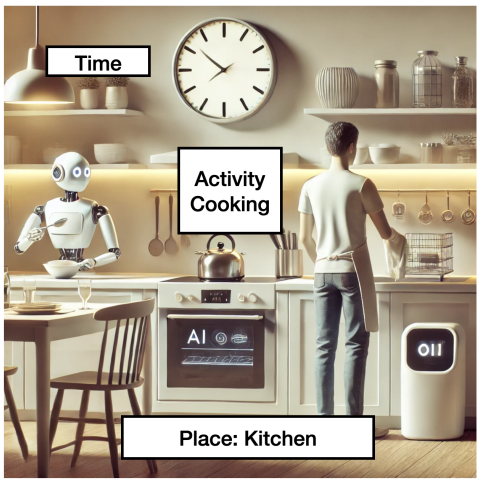

Figure 1: The agent observes the user performing an activity and records action, time, and place.

2025)); (iii) provides novel evaluations of temporal encoding and sampling strategies; (iv) supports practical applications in assistive memory, planning, and cognitive agent design.

Note. Throughout the paper, the term *user* refers to the human whose behaviors and contexts the agent learns and supports over time.

2 Related Work

Graphs for Episodic Memory Recommendation. Static graphs fail to capture temporal dynamics, evolving relationships, and incremental node additions, limiting their utility for episodic memory recommendation. Graph Neural Networks (GNNs) (Jin, Song, and Shi, 2019; Wang, Jiang, Syed, Conway, Juneja, Subramanian, and Chawla, 2020; Song, Li, Chang, Xie, Hao, and Qin, 2024) perform well in node classification but struggle in sequential, evolving contexts such as life-logging. Dynamic graph learning addresses this by modeling temporal interactions, with applications in social networks, transportation, and biology (Barros, Mendonça, Vieira, and Ziviani, 2021; Skarding, Gabrys, and Musial, 2021; Xue, Zhong, Li, Chen, Zhai, and Kong, 2022; Kazemi, Goel, Jain, Kobyzev, Sethi, Forsyth, and Poupart, 2020; Kumar, Zhang, and Leskovec, 2019). Benchmarks such as DGB, TGB, and TransactionTempGraph (Poursafaei and Huang, 2022; Huang and Poursafaei, 2024; Zhang, Luo, Lu, and He, 2024) and toolkits like DyGLib (Trivedi, Farajtabar, Biswal, and Zha, 2019; Yu, Sun, Du, and Lv, 2023b) provide temporal data resources, but often rely on limited features (e.g., bag-of-words, word2vec (Katz, 1985; Mikolov, Chen, Corrado, and Dean, 2013)) and lack support for ordered sequences or contextual reasoning. We extend temporal graph learning by incorporating spatial, temporal, and sequential cues for personalized recommendations, and by leveraging Text-Attributed Graphs (TAGs) (Sen, Namata, Bilgic, and Getoor, 2008; Wang and Shen, 2020; Yang, Liu, Xiao, Li, Lian, and Agrawal, 2021; Yan, Li, Long, Yan, and Zhao, 2023) enriched with large language models (Yu, Ren, Gong, Tan, Li, and Zhang, 2023a; He, Bresson, Laurent, Perold, LeCun, and Hooi, 2023; Pan, Zhang, Zhang, Hu, and Zhao, 2024; Tang, Yang, Wei, Shi, and Su, 2024; Ye, Zhang, Wang, Xu, and Zhang, 2024; Zhao, Zhuo, Shen, Qu, and Liu, 2023), bridging temporal, contextual, and visual modalities.

Action Anticipation. Classical models (Kingma, Salimans, Jozefowicz, Chen, Sutskever, and Welling, 2016; Kingma and Dhariwal, 2018; Rezende and Mohamed, 2015) predict actions from sequence patterns but lack long-term memory and personalization. Marked Temporal Point Processes (MTPPs) (Hawkes, 1971; Du, Dai, Trivedi, Upadhyay, Gomez-Rodriguez, and Song, 2016; Mei and Eisner, 2017; Zhang, Lipani, Kirnap, and Yilmaz, 2020; Zuo, Jiang, Li, Zhao, and Zha, 2020) model continuous-time events, while recent flow-based methods (Shchur, Biloš, and Günnemann, 2020; Mehrasa, Deng, Ahmed, and Chang, 2019) improve sampling efficiency but remain limited in multi-context generalization. Normalizing flows (Rezende and Mohamed, 2015; Mehrasa, Deng, Ahmed, and Chang, 2019) provide tractable sampling yet fail to adapt to evolving, personalized behaviors. Recent state-of-the-art approaches—Palm, AntGPT (Zhao, Wang, Zhang, Fu, Do, Agarwal, Lee, and Sun, 2024), iCVAE (Mascaro, Ahn, and Lee, 2024), ObjectPrompt (Zhang, Fu, Wang,

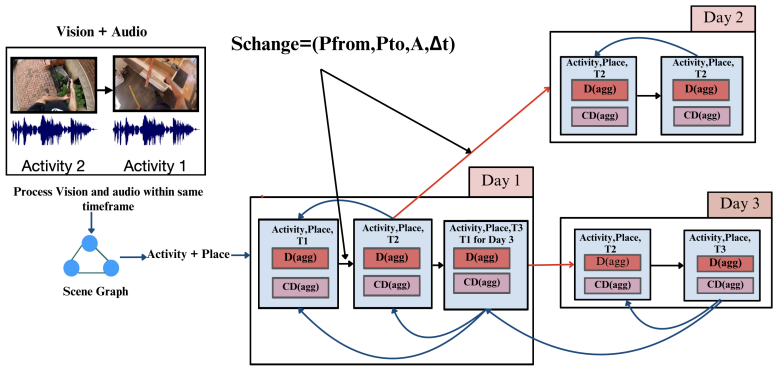

Figure 2: Episodic Memory Recommender: Visual representation of the Episodic Memory Recommender system. Nodes in a dynamic temporal graph store memories of actions, times, and places. The system updates node embeddings through a sequence of time-stamped events, utilizing aggregation functions to process daily experiences. Memory is continuously adapted using learnable functions to incorporate new interactions, facilitating context-aware action recommendations based on both historical and real-time data. D(agg) represents daily life activity aggregation and CD(agg) represents cross-activity aggregation. Curved arrows show that next activity information is saved in previous activity memory. Red arrows denote the same activity occurring at different times on other days.

Agarwal, Lee, Choi, and Sun, 2023), and Replai Mittal et al. (2022b)—leverage large-scale datasets such as Ego4D (Grauman, Westbury, Byrne, and Chavis, 2022) and EPIC-Kitchens Damen et al. (2018) but require retraining for new tasks and cannot model episodic memory or personalized anticipation. Our approach addresses these gaps by unifying temporal graphs, dynamic memory representations, and log-normal flows to enable multi-activity, long-term anticipation and personalized recommendations in evolving environments.

3 Approach

Our episodic memory-based recommendation system encodes experiences as combinations of character, action, time, and place, with a primary focus on action recommendation. The master is treated as the central character, with time represented in a detailed format (hours, days, months, and years). The agent operates in a dynamic environment, gathering audio and visual data, associating actions with specific locations and timestamps. These experiences are stored in episodic memory, enabling pattern analysis and the organization of data into structured memory modules. We define actions as fine-grained, low-level operations (e.g., *cutting*, *stirring*), while activities refer to higher-level combinations of actions (e.g., *cooking* composed of {*cut, stir, boil*}). In addition to visual cues, actions are extracted from dialogue using speech transcripts via Whisper (Radford, Kim, Xu, Brockman, McLeavey, and Sutskever, 2022). Spoken commands (e.g., "*stir the soup*") are parsed and concatenated with visual activity embeddings, improving recall for unobserved or occluded tasks.

3.1 Episodic Memory Recommender

We propose an episodic memory recommender leveraging temporal graph networks constructed using an encoder-decoder architecture. The encoder processes a continuous-time dynamic graph, represented as a sequence of time-stamped events, to generate node embeddings:

$$\mathbf{Z}(t) = \left(\mathbf{z}_1(t), \ldots, \mathbf{z}_{n(t)}(t)\right), \tag{1}$$

where $\mathbf{z}_i(t)$ denotes the embedding of node i at time t.

Node Embeddings. The embedding of each node is defined as:

$$\mathbf{z}_i(t) = f(\mathbf{e}_{\text{time}}, \mathbf{e}_{\text{activity}}, \mathbf{e}_{\text{place}}), \tag{2}$$

where $\mathbf{e}_{\text{time}}$, $\mathbf{e}_{\text{activity}}$, and $\mathbf{e}_{\text{place}}$ are feature embeddings for time, activity, and place, respectively. These embeddings incorporate semantic context extracted from transcripts (via Whisper)(Radford, Kim, Xu, Brockman, McLeavey, and Sutskever, 2022) and video segmentation, combining image and text features as in *VideoRecap* (Islam, Ho, Yang, Nagarajan, Torresani, and Bertasius, 2024).

In cases where only a single modality is used, such as textual or visual features, the model can adapt by using pre-trained models like BERT (Devlin, Chang, Lee, and Toutanova, 2019) for text-based inputs or other vision-based models for image/video-based inputs. The architecture remains flexible, allowing the concatenation of embeddings from these individual modalities to form the node representation.

Daywise Memory Representation. The episodic memory at time t is denoted as $\mathbf{z}_i(t)$, encapsulating historical experiences. When a new activity occurs at time T_j, its timestamp determines whether it aligns with existing nodes for the same day:

$$\mathcal{N}_{\text{day}} = \{i \mid T_i.\text{date} = T_j.\text{date} \wedge T_i.\text{time} \leq T_j.\text{time}\}, \tag{3}$$

where T_i.date represents the calendar date (day, month, year) of event i, and T_i.time is its occurrence time within that date.

If $\mathcal{N}_{\text{day}} \neq \emptyset$, the memory of each node $i \in \mathcal{N}_{\text{day}}$ is updated:

$$\mathbf{z}_i(t+1) = f_{\text{update}}(\mathbf{z}_i(t), \mathbf{e}_{\text{activity}_j}, \mathbf{e}_{\text{time}_j}, \mathbf{e}_{\text{place}_j}), \tag{4}$$

where f_{update} is a learnable function. If $\mathcal{N}_{\text{day}} = \emptyset$, a new node is initialized:

$$\mathbf{z}_j(t+1) = f(\mathbf{e}_{\text{time}_j}, \mathbf{e}_{\text{activity}_j}, \mathbf{e}_{\text{place}_j}). \tag{5}$$

Final Memory Representation. Each day's memory aggregates activities, times, and places using a function f_{agg}, which concatenates features, sequences actions, or computes statistical summaries:

$$\mathbf{z}_i(t) = f_{\text{agg}}(\{\mathbf{e}_{\text{time}_k}, \mathbf{e}_{\text{activity}_k}, \mathbf{e}_{\text{place}_k} \mid k \in \mathcal{N}_{\text{day}}\}). \tag{6}$$

Activity-Based Message Function. For each activity the agent/master performs on the next day involving a node i, a message is computed to update the memory of i. For an interaction activity $\mathbf{e}_{ij}(t)$ between nodes i and j at time t, the system computes:

$$\mathbf{m}_i(t) = \text{msg}_{\text{s}}\left(\mathbf{s}_i(t^-), \mathbf{s}_j(t^-), \Delta t, \mathbf{e}_{\text{action}_i}\right), \tag{7}$$

$$\mathbf{m}_j(t) = \text{msg}_{\text{d}}\left(\mathbf{s}_j(t^-), \mathbf{s}_i(t^-), \Delta t, \mathbf{e}_{\text{action}_j}\right). \tag{8}$$

For node-wise events involving only a single node i:

$$\mathbf{m}_i(t) = \text{msg}_{\text{n}}\left(\mathbf{s}_i(t^-), t, \mathbf{e}_{\text{action}_i}\right). \tag{9}$$

Here, $\mathbf{s}_i(t^-)$ is the memory of node i just before time t. The functions $\text{msg}_{\text{s}}, \text{msg}_{\text{d}}$, and msg_{n} are learnable message functions capturing temporal dynamics.

Message Aggregator. When multiple activities or place involve the same node i in a batch, an aggregation function combines the messages:

$$\bar{\mathbf{m}}_i(t) = \text{agg}\left(\mathbf{m}_i(t_1), \ldots, \mathbf{m}_i(t_b)\right), \tag{10}$$

where agg incorporates frequency-aware weighting and temporal decay. It ensures that frequent and recent patterns are prioritized, while obsolete ones are removed.

Memory Updater. The memory of a node is updated upon each event:

$$\mathbf{s}_i(t) = \text{mem}\left(\bar{\mathbf{m}}_i(t), \mathbf{s}_i(t^-)\right), \tag{11}$$

where mem is a learnable memory update function (e.g., LSTM (Staudemeyer and Morris, 2019) or GRU (Chung, Gulcehre, Cho, and Bengio, 2014)). Since our dataset captures activities on a daily basis over a span of three years, we employ LSTMs (Staudemeyer and Morris, 2019) to maintain day-wise temporal dependencies. This choice enables the model to effectively capture long-term patterns and trends across daily sequences while retaining critical contextual information from historical activity data. The dynamic evolution of node memories ensures that both recent and long-term interactions are reflected, supporting robust action anticipation and recommendation.

Embedding: In the episodic memory framework, each node $\mathbf{z}_i(t)$ is a concatenation of place, action, and time embeddings $\mathbf{e}_{\text{place}}, \mathbf{e}_{\text{activity}}, \mathbf{e}_{\text{time}}$, capturing the context of each event. The aggregation process uses frequency-aware weighting to prioritize frequent patterns and temporal decay to reduce the relevance of older patterns. The relevance score $\phi_i(t)$ is updated based on elapsed time, and nodes with a score below the threshold ϕ_{min} are deleted. The final episodic memory $\mathbf{Z}(t)$ consists of all valid (non-deleted) nodes:

$$\mathbf{Z}(t) = \{\mathbf{z}_i(t) \mid \phi_i(t) \geq \phi_{\text{min}},\ i \in \mathcal{N}(t)\}. \tag{12}$$

The time encoding mechanism utilizes the Wavelet time Encoding (WT) (Sasal, Chakraborty, and Hadid, 2022) to map timestamps into a meaningful feature representation. Given a sequence of timestamps $t_1, t_2, \ldots, t_n$, the wavelet coefficients for each timestamp are extracted via WT, yielding a fixed-length vector $\mathbf{z}_i(t)$:

$$\mathbf{z}_i(t) = \text{WT}(t_i, \text{wavelet}), \quad i = 1, 2, \ldots, n \tag{13}$$

Where: - $\mathbf{z}_i(t) \in \mathbf{R}^{\text{time\_dim}}$ represents the wavelet-transformed feature vector for timestamp t_i. - WT denotes the Wavelet time Encoding applied to t_i, producing wavelet coefficients.

To handle calendar encoding, additional periodic encodings, such as day (δ_i), month (μ_i), year (ν_i), and hour (τ_i), are incorporated. The final time encoding for each node is the concatenation of these features:

$$\mathbf{z}_i(t) = [\mathbf{t}_i \parallel \delta_i \parallel \mu_i \parallel \nu_i \parallel \tau_i] \tag{14}$$

Where δ_i is the day encoding for timestamp t_i, μ_i is the month encoding for timestamp t_i, ν_i is the year encoding for timestamp t_i, and τ_i is the hour encoding for timestamp t_i.

The edge features include the event place of both nodes connected by an interaction, as well as the time difference between them. Additionally, a temporal parameter $\delta_{ij}(t)$ is defined to capture this difference:

$$\delta_{ij}(t) = \boldsymbol{\phi}(t - t_j) \| \mathbf{e}_{\text{place}}(i, j), \tag{15}$$

where $\boldsymbol{\phi}(t - t_j)$ represents the time difference between nodes i and j at time t, and $\mathbf{e}_{\text{place}}(i, j)$ encodes the spatial relationship between the connected nodes.

The input to the l-th layer of the episodic memory recommender consists of node i's multimodal representation $\mathbf{h}_i^{(l-1)}(t)$ (a concatenation of its visual embedding $\mathbf{v}_i^{(l-1)}(t)$ and textual embedding $\mathbf{u}_i^{(l-1)}(t)$), the current timestamp t, and the multimodal representations of i's temporal neighborhood $\{\mathbf{h}_1^{(l-1)}(t), \ldots, \mathbf{h}_N^{(l-1)}(t)\}$. Each neighbor's representation is paired with temporal offsets $\{t - t_1, \ldots, t - t_N\}$ and interaction-specific multimodal features $\{\mathbf{e}_{i1}(t_1), \ldots, \mathbf{e}_{iN}(t_N)\}$.

The layer aggregates these inputs using multi-head attention:

$$\tilde{\mathbf{h}}_i^{(l)}(t) = \text{MultiHeadAttention}^{(l)}(\mathbf{q}^{(l)}(t), \mathbf{K}^{(l)}(t), \mathbf{V}^{(l)}(t)), \tag{16}$$

where the query $\mathbf{q}^{(l)}(t)$ is i's current multimodal state, while keys $\mathbf{K}^{(l)}(t)$ and values $\mathbf{V}^{(l)}(t)$ are derived from the multimodal neighborhood context:

$$\begin{aligned}\mathbf{C}^{(l)}(t) = \Big[&\mathbf{h}_1^{(l-1)}(t) \parallel \mathbf{e}_{i1}(t_1) \parallel \boldsymbol{\phi}(t - t_1),\ \mathbf{h}_2^{(l-1)}(t) \parallel \mathbf{e}_{i2}(t_2) \parallel \boldsymbol{\phi}(t - t_2),\ \ldots, \\ &\mathbf{h}_N^{(l-1)}(t) \parallel \mathbf{e}_{iN}(t_N) \parallel \boldsymbol{\phi}(t - t_N)\Big].\end{aligned} \tag{17}$$

The final node representation is:

$$\mathbf{h}_i^{(l)}(t) = \text{MLP}^{(l)}(\mathbf{h}_i^{(l-1)}(t) \parallel \tilde{\mathbf{h}}_i^{(l)}(t)), \tag{18}$$

where $\boldsymbol{\phi}(\cdot)$ encodes temporal offsets. This framework integrates episodic memory by combining multimodal node features and temporal dynamics to enhance recommendation accuracy.

Sequence-Level Prediction with CTC Loss. To anticipate the future sequence of actions or activities, we employ Connectionist Temporal Classification (CTC) loss, which aligns predicted outputs with target sequences without requiring exact frame-level alignment:

$$\mathcal{L}_{\text{CTC}} = -\log \sum_{\pi \in \mathcal{B}^{-1}(y)} P(\pi \mid \mathbf{Z}), \tag{19}$$

where $\mathcal{B}^{-1}(y)$ denotes all valid alignments for a target label sequence y, and $\mathbf{Z}$ are model outputs over time. **Example:** If the ground truth activity is *cooking*, composed of action sequence {`cut, stir, boil`}, the model may observe events at irregular time intervals and predict {`cut, cut, stir, boil`}. CTC allows matching such predictions to the true activity label, accommodating temporal variation in action lengths and ordering.

3.1.1 ILLUSTRATIVE SCENARIO

We demonstrate our system through a simplified three-day example in which the agent observes and supports the user ("master") throughout daily activities:

Day 1: At 8:00 AM, the master enters the kitchen. The agent tracks fine-grained actions: opening the fridge, pouring milk, stirring cereal, drinking coffee. By 8:30 AM, the master leaves for the office. The agent passively notes key events like lunch at 1:00 PM and playing guitar in the evening.

Day 2: The routine repeats with slight variations (e.g., pouring milk at 8:01 AM or playing guitar later), helping the agent learn consistent patterns while accommodating natural variability.

Day 3: The morning is similar, but instead of leaving, the master begins remote work on a laptop. The agent observes new patterns: an evening walk at 5:00 PM and an earlier dinner.

The agent encodes these episodes as temporal graphs using multimodal inputs (audio, visual, speech), with fine-grained actions as nodes (e.g., `pour_milk@8:01AM`) and timestamped edges capturing sequential dependencies. These graphs evolve via message-passing to retain temporal and contextual structure.

Such representations enable:

Next-Action Anticipation: Predicting likely next steps (e.g., start work, take a break) based on past routines.

Memory Support: Offering gentle prompts when the user hesitates or seems confused (e.g., *"You usually stir the cereal next—would you like to do that now?"*).

Procedure Recovery: Answering queries like *"How do I make tea?"* by retrieving observed sequences (e.g., boil water $\rightarrow$ steep tea $\rightarrow$ pour $\rightarrow$ drink).

Context-Aware Suggestions: Identifying patterns to suggest timely actions (e.g., *"Time for your walk?"* or *"Want to play guitar now?"*).

4 EXPERIMENTAL SETUP

Dataset. We build a structured dataset from Ego4D (Grauman, Westbury, Byrne, and Chavis, 2022; Zhou, Cao, Zheng, Zheng, and Liu, 2025), which is originally unorganized with respect to activity timelines. We reorganize the videos chronologically to yield three years of daily multimodal records covering gardening, exercising, work, social interactions, shopping, and hobbies. The dataset is derived entirely from existing Ego4D annotations—scenario-level annotations are grouped as activities, and fine-grained annotations as actions—without introducing new labels. To construct the 3-year timeline, we repeat certain sequences while maintaining temporal consistency. Balanced activity representation mitigates bias and enables modeling of continuity, variation, and long-term behavioral evolution.Details of dataset construction are as in Appendix A

Implementation Details. The model applies a time-scaling factor (1×10^{-6}) to weight sampling by temporal intervals, where larger values prioritize recent events and 0.0 yields uniform sampling. The walk encoder employs 10 attention heads with 5-step random walks to capture temporal-structural dependencies, optimized for two NVIDIA RTX 6000 GPUs (24GB each). Temporal evolution is modeled with a fixed gap of 10 units, and embedding dimensions are set to 100 (time) and 172 (position). EdgeBank memory provides unbounded storage with optional time-window and repeat-threshold modes. Training uses Adam (lr = 0.0001), dropout = 0.1, and early stopping with patience = 5. Data is split 70%/15%/15% for train/validation/test. Multimodal features (text and image embeddings from *videorecap* (Islam, Ho, Yang, Nagarajan, Torresani, and Bertasius, 2024)) enrich node representations. We compare against DyRep with time-interval–aware neighbor sampling, and note that hyperparameters are tunable for other datasets.

5 EXPERIMENTS AND RESULTS

Comparison with long term anticipation models In this experiment, we perform a comparison with state-of-the-art long-term anticipation models, evaluated based on their ability to predict future activities. Table 1 presents a comparison of different methods based on the Action metric, where lower values indicate better performance. The models used for comparison include RUSTLM (Mittal, Morgado, Jain, and Gupta, 2022a), ICVAE, (Mascaro, Ahn, and Lee, 2024) CLIP+Transformer

(Radford, Kim, Hallacy, and Ramesh, 2021; Vaswani, Shazeer, Parmar, and Uszkoreit, 2023), Object Prompt, (Zhang, Fu, Wang, Agarwal, Lee, Choi, and Sun, 2023) Palm, and AntGPT (Zhao, Wang, Zhang, Fu, Do, Agarwal, Lee, and Sun, 2024). These models have achieved state-of-the-art results in action anticipation benchmarks, which is why we selected them for evaluating our model's performance.

Evaluation Matrix: Following the evaluation protocols in the Ego4D LTA benchmark (Ishibashi, Ono, Kugo, and Sato, 2023), we report the Edit Distance (ED) (Przybocki, Sanders, and Le, 2006) metric, calculated as the distance over predicted sequences of actions. This metric captures the similarity between predicted and ground truth action sequences, accounting for minor variations. Lower ED values signify higher alignment between predictions and actual sequences, emphasizing the model's ability to anticipate actions accurately over extended time horizons.

Results:

Table 1 proves that episodic memory recommender can recommend sequences of activities better than other baselines, showing it can perform long-term anticipation and record environmental state as it evolves in relation to its master. For applications like social companions for individuals with memory disorders, it's crucial that predictions account for the person's lifestyle and activity evolution over time. Our model excels by integrating daily and yearly patterns, ensuring context is maintained; unlike others, it considers temporal dependencies and personal routines, making it better suited for long-term anticipation tasks and real-life applications in social companionship for people with memory disorders.

Method	**Action** $\downarrow$
RUSTLM	0.9432
ICVAE	0.9304
CLIP+Transformer	0.9290
Object Prompt	0.9276
Palm	0.9120
AntGPT	0.8853
Ours	**0.7220**

Table 1: Comparison of methods based on the Action metric (lower is better).

5.1 Comparison with Other Graph Models

We evaluate state-of-the-art graph-based models—widely applied in temporal domains such as social networks, transportation, and biology—on lifelog data. The task involves predicting the next activity (*Test Score*) and integrating unseen activities into an episodic memory recommender (*New Node Score*), thereby testing models' ability to support long-term anticipation and personalized activity prediction.

Evaluation Metrics. Following prior work (Bradley, 1997), we adopt Average Precision (AP) and AUC-ROC as metrics for activity recommendation.

Graph Model	**Test ROC Accuracy**	**Test Precision**	**New Node ROC Accuracy**	**New Node Precision**
JODIE	0.7088	0.6938	0.6302	0.5936
DyRep	0.7792	0.7575	0.7297	0.7308
TGAT	0.6670	0.6436	0.5430	0.5660
TGN	0.6820	0.7398	0.6298	0.5840
CAWN	0.6776	0.7024	0.6008	0.5602
TCL	0.6260	0.6623	0.5505	0.5985
GraphMixer	0.6153	0.6550	0.5610	0.5471
DyGFormer	0.6791	0.6626	0.5860	0.6290
Ours	**0.8664**	**0.8185**	**0.7962**	**0.8020**

Table 2: Comparison of graph models on link prediction for lifelog data.

As shown in Table 2, our model outperforms all baselines across both test and new-node settings. This indicates that capturing only recent activity patterns is insufficient for lifelog recommendation. Instead, maintaining structured sequential memory at daily resolution is essential for personalized, context-aware predictions, going beyond local neighborhood or similarity-based reasoning.

5.2 Ablation Studies

In the ablation studies, we aim to validate our approach by changing hyperparameters and evaluating specific parameters of the model. To achieve this, we measure the Test ROC AUC score, which assesses how effectively the agent can recommend the next activity. Additionally, we analyze the New Node score to evaluate how accurately the model can incorporate new nodes into its memory. These metrics provide insights into the model's recommendation capability and its ability to adapt and expand its episodic memory effectively.

Time Encoding. Table 3 reports the performance of different temporal encoding methods, including Sinusoidal (Sun, Yuan, Xu, Mai, Siddharth, Chen, and Marina, 2024), Cosine (Vaswani, Shazeer, Parmar, and Uszkoreit, 2023), Wavelet (Sasal, Chakraborty, and Hadid, 2022), Fourier, Gaussian (Ren, Wang, Jia, Laili, and Zhang, 2023), and learnable matrices. Among these, Wavelet encoding achieves the best results across all metrics, highlighting its effectiveness in activity recommendation tasks.

Time Encoding	Test ROC Accuracy	Test Precision	New Node ROC Accuracy	New Node Precision
Sinusoidal	0.779	0.705	0.611	0.591
Cosine	0.758	0.716	0.593	0.550
Wavelet	**0.807**	**0.819**	**0.796**	**0.805**
Fourier	0.659	0.547	0.616	0.661
Gaussian	0.617	0.537	0.581	0.555
Learnable	0.736	0.733	0.485	0.506

Table 3: Comparison of time encoding methods for link prediction tasks, measured by Test ROC Accuracy, Test Precision, New Node ROC Accuracy, and New Node Precision.

Wavelets are particularly effective as they decompose signals into both time and frequency components, enabling detection of patterns across multiple scales. This is well suited for lifelog data, where activities often follow periodic rhythms such as daily routines and seasonal variations. Unlike sinusoidal or cosine encodings with fixed frequencies, wavelets provide adaptive representations that better align with natural temporal evolution. This adaptability enhances modeling of both local and global temporal dependencies, leading to consistent gains in ROC and precision metrics, and confirming the suitability of wavelet encoding for personalized activity prediction.

Backbone Architecture Comparison. To evaluate the effect of backbone choice on pattern-oriented link prediction, we substituted different architectures into our framework and measured performance across four metrics. Results (Table 4) show substantial variation: while models such as TGN and TGAT perform competitively, DyRep consistently achieves the best results across Test ROC Accuracy, Test Precision, New Node ROC Accuracy, and New Node Precision.

Backbone	Test ROC Accuracy	Test Precision	New Node ROC Accuracy	New Node Precision
JODIE	0.6118	0.6226	0.5586	0.5682
DyFormer	0.6010	0.6187	0.6545	0.6656
TGAT	0.6618	0.7023	0.6125	0.6414
TGN	0.7071	0.7205	0.6102	0.5779
CAWN	0.5688	0.5339	0.4899	0.4689
TCL	0.6780	0.6584	0.5921	0.5956
GraphMixer	0.6249	0.6386	0.5236	0.4892
DyRep	**0.8066**	**0.8185**	**0.7962**	**0.8050**

Table 4: Backbone comparison on link prediction. DyRep outperforms all alternatives across test and new-node settings.

DyRep's advantage stems from its ability to capture both global and local temporal dependencies. Its event-driven formulation models evolving relationships and irregular dynamics, enabling robust performance in life-log data with mixed periodic and non-periodic behaviors. This adaptability makes it particularly effective for long-horizon sequential prediction tasks.

Loss Function Evaluation. We examined the effect of different loss functions on link prediction, with results shown in Table 5. Performance is reported across Test ROC Accuracy, Test Precision, New Node ROC Accuracy, and New Node Precision. CTC Loss yields the best performance overall, highlighting its strength in sequence alignment for temporally ordered prediction tasks. Cross Entropy performs competitively, especially in Test and New Node Precision, while Binary Cross Entropy and L1 provide moderate results. Mean Squared Error underperforms, confirming its limitations for categorical sequential data. Log Likelihood surpasses BCE but remains weaker than CE and CTC.

Negative Sampling Strategies Comparison We evaluated the impact of different negative sampling strategies on model performance for link prediction tasks. Negative sampling is crucial for distinguishing true connections from false ones, improving the model's ability to recommend future activities and maintain sequential memory structures. Table 5.2 presents results for three strategies:

Loss Function	Test ROC Accuracy	Test Precision	New Node ROC Accuracy	New Node Precision
Binary Cross Entropy	0.6070	0.6947	0.6097	0.5722
Cross Entropy Loss	0.6163	0.7181	0.5808	0.6739
Mean Squared Error	0.7708	0.7575	0.4802	0.5282
L1 Loss	0.6173	0.6577	0.5660	0.6204
Log Likelihood	0.6561	0.6124	0.5909	0.5497
CTC Loss	**0.8066**	**0.8185**	**0.7962**	**0.8040**

Table 5: Comparison of loss functions on link prediction. CTC Loss achieves the strongest results across all metrics.

Negative Sampling Techniques	Test ROC Accuracy	Test Precision	New Node ROC Accuracy	New Node Precision
Random	0.7725	0.7576	0.7335	0.7295
Historical	0.7650	0.7788	0.6990	0.7042
Inductive	**0.8066**	**0.8185**	**0.7962**	**0.8040**

Table 6: Performance comparison of different negative sampling techniques for link prediction tasks. Metrics include Test ROC Accuracy, Test Precision, New Node ROC Accuracy, and New Node Precision.

Random, Historical, and Inductive sampling. Random Sampling selects negative samples uniformly, achieving moderate results (Test ROC: 0.7725) but lacks contextual awareness. Historical Sampling uses past data to improve precision (0.7788) but struggles with generalization. Inductive Sampling outperforms both, aligning samples with temporal context and achieving the best metrics (Test ROC: 0.80664), making it ideal for episodic memory and activity recommendations.

Sampling Strategies We analyzed the performance of different sampling techniques for link prediction tasks, including Uniform, Recent, and Time Interval Aware approaches (Table 5.2). Uniform sampling, which selects nodes uniformly, achieved moderate results (Test ROC: 0.7659) but failed to prioritize relevant temporal dynamics. Recent sampling, focusing on the latest interac-

Sampling Techniques	Test ROC Accuracy	Test Precision	New Node ROC Accuracy	New Node Precision
Uniform	0.7659	0.7583	0.7739	0.7181
Recent	0.6750	0.7181	0.5808	0.6739
Time Interval aware	**0.8066**	**0.8185**	**0.7962**	**0.8040**

Table 7: Comparison of sampling techniques for link prediction tasks, showing Test ROC Accuracy, Test Precision, New Node ROC Accuracy, and New Node Precision.

tions, struggled with generalization, particularly for unseen nodes (New Node ROC: 0.5808). The Time Interval Aware method outperformed others across all metrics (Test ROC: 0.80664, New Node ROC: 0.7962), as it effectively captured temporal patterns and contextual relevance, demonstrating its utility for sequential and time-sensitive tasks like activity anticipation and memory modeling.

6 Conclusion Limitations and Future Work

This paper presents a novel episodic memory-based framework that integrates temporal graph networks with multimodal data for long-term action anticipation and activity recommendations. By leveraging daily and yearly activity patterns, the model achieves state-of-the-art performance in predicting complex future action sequences. Key innovations include adaptive memory representations, advanced time encoding, and robust sampling strategies, enabling the system to dynamically utilize past experiences. These capabilities make the model applicable in real-world scenarios, such as assistive technologies, adaptive robotics, and personalized recommendations. The framework offers societal benefits by aiding individuals with memory impairments through reminders and navigation guidance, while improving human-robot collaboration and task efficiency. Limitations include dependency on recurring patterns and large datasets, which may hinder performance in sparse environments. Additionally, the computational complexity of multimodal data processing and dynamic memory updates poses challenges in resource-constrained settings. Future work will focus on optimizing for such scenarios, integrating reinforcement learning for context-aware decisions, and expanding the framework to incorporate additional modalities, such as physiological and environmental data.

ACKNOWLEDGMENTS

The authors used a large language model (ChatGPT) solely to polish grammar and improve the clarity of writing. All research ideas, experiments, analyses, and conclusions are entirely the work of the authors.

REFERENCES

Claudio D.T. Barros, Matheus R. F. Mendonça, Alex B. Vieira, and Artur Ziviani. A survey on embedding dynamic graphs. *ACM Computing Surveys*, 55(1):1–37, 2021.

Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognit.*, 30:1145–1159, 1997. URL https://api.semanticscholar.org/CorpusID:13806304.

Aakanksha Chowdhery, Sharan Narang, and Jacob Devlin. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. URL https://arxiv.org/abs/1412.3555.

Dima Damen, Hazel Doughty, and Giovanni Maria Farinella. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *KDD*, 2016.

Kristen Grauman, Andrew Westbury, Eugene Byrne, and Zachary Chavis. Ego4d: Around the world in 3,000 hours of egocentric video, 2022. URL https://arxiv.org/abs/2110.07058.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *International Conference on Learning Representations*, 2023.

Shenyang Huang and Poursafaei. Temporal graph benchmark for machine learning on temporal graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

Tatsuya Ishibashi, Kosuke Ono, Noriyuki Kugo, and Yuji Sato. Technical report for ego4d long term action anticipation challenge 2023, 2023. URL https://arxiv.org/abs/2307.01467.

Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video recap: Recursive captioning of hour-long videos, 2024. URL https://arxiv.org/abs/2402.13250.

Yilun Jin, Guojie Song, and Chuan Shi. Gralsp: Graph neural networks with local structural patterns, 2019. URL https://arxiv.org/abs/1911.07675.

Jerrold J. Katz. *The philosophy of linguistics*. Oxford University Press, 1985.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *NeurIPS*, 2016.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1269–1278, 2019.

Esteve Valls Mascaro, Hyemin Ahn, and Dongheui Lee. Intention-conditioned long-term human egocentric action forecasting, 2024. URL https://arxiv.org/abs/2207.12080.

Nazanin Mehrasa, Ruizhi Deng, Mohamed Osama Ahmed, and Bo Chang. Point process flows. *arXiv preprint arXiv:1910.08281*, 2019.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *NeurIPS*, 2017.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International conference on learning representations*, 2013.

Himangi Mittal, Pedro Morgado, Unnat Jain, and Abhinav Gupta. Learning state-aware visual representations from audible interactions, 2022a. URL https://arxiv.org/abs/2209.13583.

Himangi Mittal, Pedro Morgado, Unnat Jain, and Abhinav Gupta. Learning state-aware visual representations from audible interactions. *arXiv preprint arXiv:2209.13583*, 2022b.

Bo Pan, Zheng Zhang, Yifei Zhang, Yuntong Hu, and Liang Zhao. Distilling large language models for text-attributed graph learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 1836–1845, 2024.

Farimah Poursafaei and Huang. Towards better evaluation for dynamic link prediction. *Advances in Neural Information Processing Systems*, 35:32928–32941, 2022.

Mark Przybocki, Gregory Sanders, and Audrey Le. Edit distance: A metric for machine translation evaluation. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odijk, and Daniel Tapias (eds.), *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL https://aclanthology.org/L06-1088/.

Alec Radford, Jong Wook Kim, Chris Hallacy, and Aditya Ramesh. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL https://arxiv.org/abs/2212.04356.

Lei Ren, Haiteng Wang, Zidi Jia, Yuanjun Laili, and Lin Zhang. Time-varying gaussian encoder-based adaptive sensor-weighted method for turbofan engine remaining useful life prediction. *IEEE Transactions on Instrumentation and Measurement*, 72:1–11, 2023. doi: 10.1109/TIM.2023.3291733.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Lena Sasal, Tanujit Chakraborty, and Abdenour Hadid. W-transformers: A wavelet-based transformer framework for univariate time series forecasting. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 671–676. IEEE, December 2022. doi: 10.1109/icmla55696.2022.00111. URL http://dx.doi.org/10.1109/ICMLA55696.2022.00111.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, and Lise Getoor. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *ICLR*, 2020.

Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *iEEE Access*, 9:79143–79168, 2021.

Wenfeng Song, Shuai Li, Tao Chang, Ke Xie, Aimin Hao, and Hong Qin. Dynamic attention augmented graph network for video accident anticipation. *Pattern Recognition*, 147:110071, 2024. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.110071. URL https://www.sciencedirect.com/science/article/pii/S0031320323007689.

Ralf C. Staudemeyer and Eric Rothstein Morris. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 2019. URL https://arxiv.org/abs/1909.09586.

Chuanhao Sun, Zhihang Yuan, Kai Xu, Luo Mai, N. Siddharth, Shuo Chen, and Mahesh K. Marina. Learning high-frequency functions made easy with sinusoidal positional encoding, 2024. URL https://arxiv.org/abs/2407.09370.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, and Su. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1836–1845, 2024.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.

Endel Tulving. Episodic memory: From mind to brain. *Annual review of psychology*, 53:1–25, 02 2002. doi: 10.1146/annurev.psych.53.100901.135114.

Ashish Vaswani, Noam Shazeer, Niki Parmar, and Jakob Uszkoreit. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Daheng Wang, Meng Jiang, Munira Syed, Oliver Conway, Vishal Juneja, Sriram Subramanian, and Nitesh V. Chawla. Calendar graph neural networks for modeling time structures in spatiotemporal user behaviors, 2020. URL https://arxiv.org/abs/2006.06820.

Kuansan Wang and Shen. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, 2020.

Guotong Xue, Ming Zhong, Jianxin Li, Jia Chen, Chengshuai Zhai, and Ruochen Kong. Dynamic network embedding survey. *Neurocomputing*, 472:212–223, 2022.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, and Zhao. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264, 2023.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, and Agrawal. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34:28798–28810, 2021.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Natural language is all a graph needs. In *Findings of the Association for Computational Linguistics*, pp. 1955–1973, 2024.

Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang Zhang. Empower text-attributed graphs learning with large language models (llms). *arXiv preprint arXiv:2310.09872*, 2023a.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36:67686–67700, 2023b.

Ce Zhang, Changcheng Fu, Shijie Wang, Nakul Agarwal, Kwonjoon Lee, Chiho Choi, and Chen Sun. Object-centric video representation for long-term action anticipation, 2023. URL https://arxiv.org/abs/2311.00180.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes processes. In *ICML*, 2020.

Zhen Zhang, Bingqiao Luo, Shengliang Lu, and Bingsheng He. Live graph lab: Towards open, dynamic and real transaction graphs with nft. *Advances in Neural Information Processing Systems*, 36, 2024.

Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, and Liu. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*, 2023.

Qi Zhao, Shijie Wang, Ce Zhang, Changcheng Fu, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Antgpt: Can large language models help long-term action anticipation from videos?, 2024. URL https://arxiv.org/abs/2307.16368.

Wenqi Zhou, Kai Cao, Hao Zheng, Xinyi Zheng, and Miao Liu. X-lebench: A benchmark for extremely long egocentric video understanding, 2025. URL https://arxiv.org/abs/2501.06835.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *ICML*, 2020.

A Detailed Dataset Construction from Ego4D

Our structured timeline dataset is derived entirely from existing Ego4D (Grauman, Westbury, Byrne, and Chavis, 2022; Zhou, Cao, Zheng, Zheng, and Liu, 2025) annotations. To clarify the construction process for reviewers, we detail how metadata, scenarios, moments, and fine-grained actions are utilized:

1. Video selection. We use all Ego4D videos containing scenario-level and moment-level annotations, as specified in the official JSON metadata (e.g., `ego4d.json`). Each video entry includes a unique `video_uid`, duration, frame rate, resolution, and device information.

2. Scenario-level grouping. Each video contains one or more scenarios describing high-level activities (e.g., "jobs related to construction/renovation company: Director of work, tiler, plumber, electrician, handyman"). These scenarios are grouped as *activities* in our dataset.

3. Moment-level fine-grained actions. Ego4D provides moment-level annotations capturing detailed actions performed by people within a scenario (e.g., hammering, pouring, typing, lifting). These are mapped as *actions* under the corresponding activity, preserving temporal order within the video.

4. Chronological organization. Videos are reordered into a daily timeline to simulate continuous multi-year activity logs. For each day, we concatenate scenarios and associated actions, maintaining intra-video temporal consistency.

5. Timeline extension and balancing. To cover a 3-year period, sequences are repeated where necessary while preserving order. Activities are approximately balanced across categories such as work, social interactions, hobbies, and shopping to reduce bias and enable long-term modeling.

6. Data storage. Each day is stored as a folder containing the reordered videos, scenario-level activities, and fine-grained action annotations. This structure allows downstream models to access both high-level and detailed behavioral cues.

This approach ensures full reproducibility using only existing Ego4D metadata, while providing detailed temporal and behavioral context for long-term multimodal modeling.