TrajLift: Lifting Memory Recall Trajectories with Latency-Modulated Heat Diffusion

Jiawen Kang **Jinchao Li** **Mingyu Cui** **Junan Li** **Xixin Wu** **Helen Meng**
The Chinese University of Hong Kong, Hong Kong SAR, China
jwkang@se.cuhk.edu.hk

Abstract

Verbal memory retrieval carries rich diagnostic information for Alzheimer's disease through the order, hesitation, and categorical organization of recalled words, yet computational modeling of this retrieval process remains largely unexplored. A key challenge is that timing and hierarchical structure are diagnostically entangled: the diagnostic meaning of retrieval timing depends on its hierarchical context. We propose TrajLift, a spectral framework that models verbal recall as timed trajectories on a hierarchical graph. Our central mechanism is latency-modulated heat diffusion on the semantic hierarchy, a unified operator that lifts each recalled word into a representation of both multi-scale hierarchical structure and continuous retrieval timing. Formal analysis shows that the resulting representation is first-order separable into structural and temporal components with spectral selectivity across hierarchical levels. On synthetic benchmarks requiring structure-time interaction, TrajLift stays near-perfect on deep hierarchies where all baselines fail. On real-world clinical data, we show that TrajLift outperforms diverse baselines by jointly modeling timing and semantic structure, with gains that grow as retrieval timing becomes more informative. Code is available at `https://github.com/kjw11/TrajLift`.

1 Introduction

Alzheimer's disease (AD) is a prevalent neurodegenerative disorder in which memory impairment is identified as one of the most reliable early markers [1]. Verbal memory tests, which assess memory encoding and retrieval of newly learned word lists, have therefore been widely used as a standard AD screening protocol [2, 3]. Clinical and cognitive research has established that the retrieval process carries rich diagnostic information: the order, hesitation, and categorical organization of recalled words reflect how a subject searches through semantic memory [4, 5], and patients with AD exhibit distinctively altered patterns in this process [1, 3]. However, computational modeling of this retrieval process remains largely unexplored. This work aims to model the retrieval dynamics as a structured representation for automatic cognitive screening.

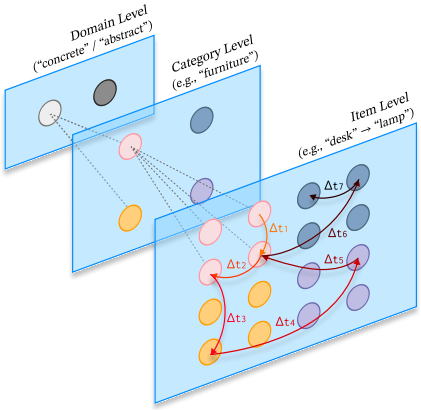


Figure 1: Verbal memory retrieval depicted as a timed trajectory on a three-level semantic hierarchy. Arrows indicate recall order; Δt_i denotes the latency between consecutive items.

In a typical verbal memory test protocol, subjects are presented with a fixed set of words organized under semantic categories (e.g., "desk" and "lamp" under "furniture") and asked to recall as many as possible after a delay. A subject's recall then produces an ordered sequence of words, each separated by a measurable latency. We therefore formalize verbal recall as *a timed trajectory on a hierarchical graph*, where nodes are words, edges are weighted by se-

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Inductive biases of existing approaches for trajectory learning on graphs.

Paradigm	Representative	Static Graph	Hierarchy	Continuous Time
Time-Aware Sequence	Time-LSTM [8]	-	✗	✓
Dynamic Graph Learning	TGN [9], TGAT [10]	✗	✗	✓
Session/Trajectory GNN	SR-GNN [11], TrajGAT [12]	✓	✗	✗
Hierarchical Pooling	DiffPool [13], SAGPool [14]	✓	✓ (learned)	✗
Ours	**Proposed**	✓	✓ (explicit)	✓

mantic proximity in the hierarchy, and each step records a visited node together with the inter-item latency. Figure 1 illustrates this representation.

Learning representations from such trajectories requires capturing the interaction between hierarchical structure and the continuous timing of each transition. For instance, in semantic fluency tasks, pauses within a semantic cluster and pauses at switches between clusters reflect distinct retrieval processes [6], and AD affects them differently [7] — illustrating that the diagnostic interpretation of timing depends on hierarchical context. This setting falls between established modeling paradigms, and the closest existing approaches come from adjacent fields that were designed for different settings. For example, dynamic graph models assume an evolving topology (i.e., changing edges), whereas the semantic hierarchy here is static across all subjects; conversely, graph-based trajectory methods operate on static structures but treat timing as discrete or omit it entirely, whereas retrieval latency is inherently continuous. We provide a detailed comparison in Table 1.

In this work, we model verbal retrieval as a graph heat diffusion process on the semantic hierarchy, where each recalled word is treated as a localized signal on the graph that spreads to its structural neighborhood. Heat diffusion on hierarchically weighted graphs has the well-known property that the diffusion time (i.e., a continuous scalar) controls the structural resolution of the resulting representation. We exploit this to lift each node visit to progressively coarser hierarchical scales, capturing within-category and cross-domain structure. We further extend this framework to encode temporal dynamics by letting the observed inter-item latency modulate the diffusion time within a bounded range, so that latency alters the representation at each scale without changing which hierarchical level that scale resolves. Formal analysis shows that under this formulation, the structural and temporal signals are first-order separable, and that the temporal encoding is sensitive to the hierarchical level. We call the resulting framework TrajLift and validate it on both synthetic benchmarks that target specific inductive biases and real-world clinical data for cognitive screening. Our contributions are summarized as follows:

- We formalize verbal memory retrieval as a timed trajectory on a hierarchical graph and propose TrajLift, a spectral framework grounded in graph heat diffusion that jointly encodes hierarchical structure and continuous timing in a single operator. This formulation also applies to other domains where ordered events unfold on fixed hierarchies (e.g., user clickstreams on content taxonomies).
- We provide formal analysis showing that the temporal modulation decouples structural and temporal signals to first order, and the multi-scale temporal encoding is spectrally selective across hierarchical levels.
- We validate the framework on synthetic benchmarks that isolate specific inductive biases and on a real-world clinical corpus for cognitive screening, where TrajLift outperforms both conventional clinical measures and computational baselines when retrieval timing is informative.

2 Related Work

2.1 Automatic AD Detection via Verbal Memory

Verbal memory assessments, such as the California Verbal Learning Test (CVLT) [2] and Hong Kong List Learning Test (HKLLT) [15], evaluate cognitive decline by requiring subjects to recall word lists across multiple trials. Recent work has moved beyond total scores to process-level measures of recall [16], however, such measures, like text-based encoders, reduce a recall sequence to aggregate statistics or a flat string. They overlook the fundamental nature of verbal retrieval as the navigation of a structured cognitive map [5], in which the sequence of recalled items forms a specific path through semantic space. To the best of our knowledge, graph-theoretic modeling of this retrieval path remains underexplored.

2.2 Inductive Biases in Graph Trajectory Learning

Modeling retrieval paths requires aligning inductive biases with the cognitive process: a continuous signal evolving over a static hierarchy. Existing paradigms, however, often misalign with this specific setting. First, regarding *dynamics*, Continuous-Time Dynamic Graph (CTDG) models [17, 18], such as TGN [9] and TGAT [10], predominantly focus on *dynamics of graphs* (evolving topology). This edge-centric bias is suboptimal for our scenario, where the semantic substrate is invariant and the diagnostic signal lies solely in the trajectory of node activations. Second, regarding *trajectory and hierarchy*, while spatial models like TrajGAT [12] and TrajDiff [19] capture sequential patterns, and encoders like Graphormers [20, 21] or R-GCNs [22] handle global structure, they typically process transitions as discrete steps. Consequently, these architectures lack a mechanism to model *continuous* inter-item latencies in relation to the structure in which they occur. They therefore cannot capture how the meaning of a latency (e.g., hesitation) depends on its position in the semantic hierarchy.

2.3 Graph Spectral Theory

To address the limitations of discrete message passing, graph spectral theory serves as a fundamental tool for analyzing dynamics on static structures [23–25]. By decomposing graph signals into orthogonal frequency components, spectral methods enable the decoupling of local high-frequency fluctuations from global low-frequency structural trends. Building on this foundation, recent research has evolved from fixed polynomial approximations to *continuous* and *adaptive* spectral filtering [26]. Notably, continuous diffusion models, such as GRAND [27] and GDC [28], reformulate information propagation as a continuous diffusion process governed by heat or oscillator equations across the graph. Unlike discrete aggregation, the spectral domain offers a continuous basis for signal evolution, providing a natural way to couple the physical timing of memory retrieval with diffusion on the graph, which underpins our proposed framework.

3 Method

We formalize the problem of representation learning for timed trajectories on hierarchical graphs and describe TrajLift, a spectral framework that jointly encodes hierarchical structure and retrieval timing through graph heat diffusion.

3.1 Problem Setting

Let $G = (V, W)$ be a weighted undirected graph with node set V of size N and adjacency matrix $W \in \mathbb{R}^{N \times N}$. The graph remains fixed across all observations and encodes prior knowledge about relationships among nodes. We consider the case where V is organized into a hierarchical partition with M nested levels. The following distance measure captures this structure.

Definition 1 (Ultrametric Distance)**.** *Given a hierarchical partition of V with M levels, the ultrametric distance $d_{\text{ultra}}(i, j)$ between nodes $i, j \in V$ is the level of their lowest common ancestor.*

Intuitively, $d_{\text{ultra}} = 1$ indicates sibling nodes sharing the same parent, $d_{\text{ultra}} = 2$ indicates cousins whose parents share a grandparent, and so on. We use this distance to specify the edge weights of G: $W_{ij} = w\big(d_{\text{ultra}}(i, j)\big)$, where $w \colon \{1, \ldots, M\} \to \mathbb{R}^+$ is a monotonically decreasing function. This assigns stronger connections to hierarchically closer nodes. In verbal memory assessment, the hierarchy is specified by the test protocol—words are grouped into categories (e.g., "desk" and "lamp" under "furniture") which belong to broader domains—and G remains the same for all subjects.

Definition 2 (Timed Trajectory)**.** *A timed trajectory on G is a sequence of timestamped node visits $\sigma = \big((v_1, t_1), (v_2, t_2), \ldots, (v_K, t_K)\big)$, where $v_k \in V$ is the visited node, $t_k \in \mathbb{R}^+$ is the timestamp, and $t_1 < t_2 < \cdots < t_K$. The **latency** between consecutive visits is $\delta_k = t_{k+1} - t_k$ for $k = 1, \ldots, K-1$. The trajectory length K may vary across observations and is itself part of the observed behavior.*

Goal. Given a trajectory σ on a hierarchical graph G, we aim to learn a representation $\mathbf{z} = f(\sigma, G) \in \mathbb{R}^d$ that captures both structural patterns and temporal dynamics for downstream classification. In verbal memory assessment, each trajectory records a subject's recall sequence together with inter-item latencies, and the classification target is cognitive status.

3.2 Multi-Scale Structural Representation

We now describe how to represent each node visit in a way that reflects its position within the hierarchy. A visit to node v_k is represented as the indicator vector $\mathbf{x}_k = \mathbf{e}_{v_k} \in \mathbb{R}^N$, which identifies the visited node but carries no information about its hierarchical neighborhood. To access the hierarchical structure encoded in G, below we analyze the graph through its Laplacian.

Let D be the degree matrix with $D_{ii} = \sum_j W_{ij}$. The normalized Laplacian is $L = I - D^{-1/2} W D^{-1/2}$. Since L is symmetric positive semi-definite, it admits an eigendecomposition $L = U \Lambda U^\top$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ with $0 = \lambda_1 \leq \cdots \leq \lambda_N$, and the columns of U are orthonormal eigenvectors. These eigenvectors form a spectral basis analogous to Fourier modes: those associated with small eigenvalues vary smoothly across the graph, while those with large eigenvalues oscillate rapidly. Because the edge weights of G decrease with ultrametric distance (Definition 1), the eigenvalues of L concentrate into bands corresponding to different levels of the hierarchy: low-frequency eigenvectors are smooth over fine-grained clusters and capture coarse partitions, while high-frequency eigenvectors retain local within-cluster variation.

Spectral lifting. To obtain a representation of $\mathbf{x}_k$ that incorporates hierarchical context, we diffuse it across G via the graph *heat kernel* $H_\tau = e^{-\tau L}$. The heat kernel attenuates each spectral component by $e^{-\tau \lambda_i}$, acting as a low-pass filter whose cutoff is controlled by a continuous scalar parameter, the *diffusion time* τ. This continuous parameterization is essential: unlike discrete-step operators such as polynomial graph filters, it allows the resolution of the hierarchy to be controlled by a single scalar. The *lifted representation* of a node visit is:

$$\mathbf{z}_k(\tau) \;=\; H_\tau \, \mathbf{x}_k \;=\; e^{-\tau L} \, \mathbf{x}_k. \tag{1}$$

Combined with the banded eigenvalue structure described above, this filtering gives rise to a precise relationship between diffusion time and hierarchical resolution. This property is well established for graphs with hierarchically structured edge weights; we state it for our construction:

Proposition 1 (Hierarchical resolution via heat diffusion)**.** *When the edge-weight ratios between adjacent hierarchy levels are sufficiently separated, the Laplacian spectrum decomposes into bands associated with the hierarchy levels. Since $e^{-\tau \lambda_i}$ decays faster for larger eigenvalues, increasing τ selectively suppresses higher-frequency bands, continuously interpolating between fine-grained and coarse representations.*

The proposition is stated for a balanced two-level hierarchy. For deeper hierarchies, the same argument applies recursively to each pair of adjacent levels. A formal statement and proof are provided in Appendix A.

Given Proposition 1, evaluating the lifting operator at different diffusion times accesses different hierarchical levels. We can therefore obtain representations at multiple resolutions by choosing several values of τ. In our setting, the hierarchy has three levels (items, categories, domains), so we instantiate three scales: $\mathbf{z}_k^{(s)} = H_{\tau_s} \mathbf{x}_k$ for $s \in \{\text{micro}, \text{meso}, \text{macro}\}$, aligned with the three levels respectively. Micro (small τ) preserves within-category structure, meso (medium τ) captures cross-category patterns within a domain, and macro (large τ) retains only cross-domain structure. The diffusion times are initialized from the spectral structure of L so that each scale's filter response aligns with a distinct eigenvalue band (Appendix D).

At this point, each recall event is represented with hierarchical context at three resolutions. However, two visits to the same node receive identical representations regardless of timing—a one-second latency and a ten-second latency after recalling "desk" produce the same lifted signal. The next section addresses this limitation.

3.3 Temporal Encoding

The lifted representations from §3.2 sample the continuous diffusion at three fixed diffusion times, each resolving a different hierarchical level. Because the ultrametric weighting produces well-separated eigenvalue bands (§3.2), the hierarchical resolution at each scale is robust to modulation of the diffusion time: as long as the modulated τ does not reach the range of an adjacent scale, shifting τ around a base value alters the within-band attenuation profile but does not change which hierarchical level is resolved. The log-compressed latency modulation bounds τ to at most three times its base value for 90% of observed latencies, which keeps the three scales in non-overlapping

ranges (Appendix D.1.3). This robustness creates a channel for temporal encoding: we can let the observed latency δ_k modulate the diffusion time at each scale, encoding temporal information without corrupting the structural representation.

To characterize the effect of this modulation, we expand $\mathbf{z}_k(\tau + \Delta\tau)$ to first order. Using $\frac{\partial}{\partial\tau} H_\tau = -LH_\tau$, we obtain

$$\mathbf{z}_k(\tau + \Delta\tau) \;=\; \underbrace{H_\tau \mathbf{x}_k}_{\text{zero-order}} \;-\; \Delta\tau \cdot \underbrace{L\, H_\tau \mathbf{x}_k}_{\text{first-order}} \;+\; O(\Delta\tau^2). \tag{2}$$

The zero-order term is the structural lift $\mathbf{z}_k(\tau)$, invariant to the modulation; the first-order term is proportional to $\Delta\tau$ and mediated by the graph Laplacian L, confirming that latency enters through a structure-aware correction while the structural base is preserved. We formalize this decomposition for reference in the spectral analysis that follows.

Lemma 1 (First-order residual separation)**.** *By Equation* (2)*, the modulated lift decomposes into a zero-order structural term invariant to latency and a first-order latency-induced perturbation mediated by the Laplacian L. The residual does not make time independent of structure; it isolates how latency perturbs the structural lift at a given scale.*

We define the *modulated diffusion time* at scale s for position k as

$$\tilde{\tau}_k^{(s)} \;=\; \tau_s^{\text{base}} \cdot \big(1 + \gamma_s \cdot \varphi(\delta_k)\big), \tag{3}$$

where $\tau_s^{\text{base}} = \text{softplus}(\beta_s)$ is the learnable base diffusion time at scale s, $\gamma_s \geq 0$ controls the modulation strength, and $\varphi(\delta) = \log(1 + \delta)$ compresses the dynamic range of observed latencies (discussed in Appendix D). Applying the heat kernel at the base and modulated diffusion times yields the *baseline lift* $\mathbf{z}_{\text{base},k}^{(s)} = H_{\tau_s^{\text{base}}} \mathbf{x}_k$ and the *modulated lift* $\mathbf{z}_{\text{mod},k}^{(s)} = H_{\tilde{\tau}_k^{(s)}} \mathbf{x}_k$. Their difference defines the *diffusion residual*:

$$\Delta\mathbf{z}_k^{(s)} \;=\; \mathbf{z}_{\text{mod},k}^{(s)} - \mathbf{z}_{\text{base},k}^{(s)}. \tag{4}$$

In practice we compute this difference exactly; Equation (2) characterizes its leading-order behavior, providing interpretability rather than serving as a computational approximation.

In the eigenbasis of L (§3.2), the first-order residual applies a *spectral response* to each frequency component of the input signal.

Proposition 2 (Band-pass temporal selectivity)**.** *Under the first-order approximation (Lemma 1), the temporal residual at scale s applies the spectral response*

$$-\Delta\tau_k^{(s)} \cdot r_s(\lambda), \qquad r_s(\lambda) \;=\; \lambda\, e^{-\tau_s^{\text{base}}\lambda}, \tag{5}$$

where $\Delta\tau_k^{(s)}$ controls the magnitude of the latency effect and r_s determines which spectral frequencies are most sensitive. The response r_s is band-pass-shaped: it vanishes at $\lambda = 0$ and as $\lambda \to \infty$, and reaches its maximum at $\lambda^\star = 1/\tau_s^{\text{base}}$.

To verify the peak, differentiate r_s with respect to the Laplacian eigenvalue: $\frac{\partial}{\partial\lambda}\big(\lambda\, e^{-\tau_s\lambda}\big) = e^{-\tau_s\lambda}(1 - \tau_s\lambda)$, which vanishes at $\lambda^\star = 1/\tau_s^{\text{base}}$. Since different scales have different τ_s^{base}, their spectral responses peak at different eigenvalues: the micro-scale residual is most sensitive to high-frequency within-category modes, while the macro-scale residual responds primarily to low-frequency cross-domain modes. This means that temporal patterns at different hierarchical levels are captured by different residual channels, enabling the model to distinguish temporal signals that arise at different structural resolutions. For example, within-category hesitation is captured by a different channel than cross-category navigation delays. When G exhibits separated eigenvalue bands, choosing diffusion scales whose peaks fall into different bands further limits cross-scale overlap; Appendix D provides a sufficient condition.

At each trajectory position, we concatenate the modulated lift and the diffusion residual at each scale with layer normalization: $\mathbf{z}_k^{(s)} = \big[\text{norm}(\mathbf{z}_{\text{mod},k}^{(s)}) \,\|\, \text{norm}(\Delta\mathbf{z}_k^{(s)})\big] \in \mathbb{R}^{2N}$. Applying this at all three scales yields $Z^{(\text{micro})}, Z^{(\text{meso})}, Z^{(\text{macro})} \in \mathbb{R}^{K \times 2N}$.

3.4 Trajectory Readout

The multi-scale representations are fused via a gated identity-residual architecture that progressively incorporates coarser context:

$$\begin{aligned} \mathbf{F}_0 &= Z^{(\text{micro})}, \\ \mathbf{F}_1 &= \mathbf{F}_0 + \sigma(\mathbf{F}_0 W_{\text{meso}}) \odot Z^{(\text{meso})}, \\ \mathbf{F} &= \mathbf{F}_1 + \sigma(\mathbf{F}_1 W_{\text{macro}}) \odot Z^{(\text{macro})}, \end{aligned} \tag{6}$$

where $W_{\text{meso}}, W_{\text{macro}} \in \mathbb{R}^{2N \times 2N}$ are learnable projections and $\sigma(\cdot)$ denotes the sigmoid function. The micro-scale serves as the identity pathway, preserving item-level dynamics; meso- and macro-scale information is incrementally gated in, mirroring the hierarchy where local patterns provide the foundation and coarser context serves as refinement.

The fused representation is projected to the target dimension $\mathbf{E} = \mathbf{F}W_{\text{proj}} + \mathbf{b}_{\text{proj}} \in \mathbb{R}^{K \times D}$. To obtain a trajectory-level representation, we aggregate the positional embeddings using a two-layer LSTM, which is well suited to the short, variable-length trajectories in our setting. The resulting representation $\mathbf{h} \in \mathbb{R}^D$ is passed to a linear classifier $\hat{\mathbf{y}} = \text{softmax}(W_c\mathbf{h} + \mathbf{b}_c)$, and the model is trained end-to-end by minimizing cross-entropy loss.

4 Data Description

4.1 Synthetic Data

To isolate and validate the necessity of joint structure-time modeling, we design a synthetic benchmark where neither modality alone provides a discriminative signal. We term this the *Structure-Time XOR (ST-XOR)* benchmark.

Table 2: Structural specifications of the hierarchy variants in the ST-XOR benchmark.

Variant	Levels	Structure	Cluster Configuration
ST-XOR	2	$16 \to 64$	16 clusters
ST-XOR-w	2	$32 \to 256$	32 clusters (**w**ider)
ST-XOR-wh	3	$4 \to 32 \to 256$	4 super- / 32 sub- (**w**ider + **h**igher)

Graph Construction. We construct hierarchies following the ultrametric substrate with three variants of increasing complexity, as detailed in Table 2.

Trajectory Generation. Each trajectory is generated via a sequential sampling process of length 12 from a uniformly sampled leaf node. Each step transitions to an intra-cluster node (same parent) or an inter-cluster node (different parent) with equal probability. Self-loops are permitted to decouple trajectory length from transition count.

Class Design. The two classes differ only in timing-structure correlation at a designated hierarchical level: (a) *Coherent*: fast intra-cluster, slow inter-cluster (μ_{fast}=0.5s, μ_{slow}=3.0s); (b) *Chaotic*: slow intra-cluster, fast inter-cluster. For ST-XOR and ST-XOR-w, cluster membership is defined at the 16- and 32-node levels, respectively. For ST-XOR-wh, it is defined at the 4-node super-cluster level, requiring models to perceive coarser structural distinctions—a more challenging setting.

XOR Trap. This design creates an XOR trap: both classes share identical marginal distributions over edge types (50% each) and identical mean latencies ($\approx$1.75s), rendering structure-only and time-only models ineffective. Only models capturing the structure-time interaction can distinguish the classes.

Dataset Statistics. Each variant comprises 2,000 balanced trajectories (1,000 per class), split into training (70%), validation (15%), and test (15%).

4.2 Real-world Data

We evaluate on a clinical speech corpus collected for cognitive screening among older adults. Each session includes the Hong Kong List Learning Test (HKLLT), a standardized verbal memory protocol requiring recall of a 16-word list organized under a three-level semantic hierarchy (see Appendix C.1).

Task Selection. HKLLT comprises multiple trials; we focus on Trial 4 (10-minute delayed recall) and Trial 5 (30-minute delayed recall), both clinically sensitive to early-stage AD. We treat these as separate benchmarks, denoted HKLLT-4 and HKLLT-5.

Preprocessing. Recordings were manually transcribed with word-level timestamps. We filtered task-irrelevant content (e.g., fillers and off-topic utterances) and retained only valid recalled items, yielding clean timed trajectories.

Participants. Inclusion criteria required participants to be aged 60+ with adequate hearing and vision. Each participant was labeled as normal cognition (NC), mild neurocognitive disorder (NCD), or major NCD by clinical experts. We merged mild and major NCD into a single NCD class. The corpus contains 402 participants. We exclude 15 participants with annotation issues in Trial 4 or Trial 5, so that the two benchmarks share the same 387 participants (NC = 233, NCD = 154).

5 Experiments

5.1 Experimental Settings

Evaluation Protocol. On the synthetic benchmarks, we report accuracy over five random seeds. On HKLLT, we report accuracy, macro-F1, and sensitivity (recall on the NCD class) over 15 stratified 80/20 train/test splits shared across all models. Scores are reported as mean and standard deviation. Comparisons with baselines use paired differences over the same splits, with Nadeau–Bengio corrected 95% confidence intervals [29]. For each type of baseline, we compare against its strongest model.

Hyperparameter Selection. In preliminary experiments, we randomly hold out 1/5 of the training set as a validation set to determine hidden dimensions for all tested models. For the proposed method, the diffusion times τ are initialized from the graph spectrum and the modulation coefficients γ from training-set latency statistics (Appendix D). No additional hyperparameter tuning is performed.

Trajectory Learning Baselines. To evaluate the representational capacity of our framework, we compare it against four categories of baselines: (a) time-informed sequential models: LSTM and Transformer with Time2Vec [30]; (b) dynamic graph models: TGN and TGAT; (c) time-informed graph models: TrajGAT and Graphormer with Time2Vec; (d) time-informed graph + sequential models: GAT [31] or R-GCN [22] combined with LSTM and Time2Vec. Category (d) serves as a controlled baseline to isolate the contribution of spectral lifting and temporal modulation. Baselines augmented with Time2Vec are denoted with a '$+$' superscript.

For hierarchy-aware comparisons, the static-graph baselines in categories (c) and (d) are provided with the same fixed semantic hierarchy used by TrajLift, whereas the dynamic graph baselines in category (b) are kept in their native temporal-edge formulation, as their architectures model evolving event edges rather than a fixed graph (Appendix D). As no prior work directly addresses this task, these baselines are selected for methodological coverage rather than recency.

NCD Detection Baselines. To compare learned representations against conventional approaches, we consider two paradigms: (a) We employ pretrained language models (Chinese BERT and RoBERTa) to encode recalled word sequences as flat text, followed by a lightweight classifier. These baselines entirely ignore retrieval dynamics. (b) Following clinical practice, we train logistic regression on handcrafted measures: recall count alone, or a low-level feature set of recall count, sequence length, category transitions, semantic clustering, mean inter-item latency, age, and years of education. The low-level feature set includes demographic inputs that TrajLift does not receive. Feature definitions are given in Appendix C, and a variant with all pairwise interactions in Appendix D.2.3.

5.2 Results

Synthetic Experiments. Table 3 reports classification accuracy on the ST-XOR benchmark. We highlight four findings: (a) Structure-agnostic models fail uniformly: LSTM$^+$ and Transformer$^+$ reach at most 68% on ST-XOR and stay near chance on the wider variants (49–53%). Without access to the graph structure, these models cannot distinguish intra-cluster from inter-cluster transitions, rendering the timing signal uninterpretable. (b) Models without explicit aggregation fail to scale: TGN and Graphormer$^+$ attain moderate accuracy on ST-XOR (73–76%), likely by implicitly memorizing cluster memberships for individual nodes. However, this strategy breaks down as the node count

Table 3: Classification accuracy (%) on the ST-XOR synthetic benchmark, mean ± SD over five random seeds. Accuracies above 90% are underlined.

Benchmark	*Time-informed Sequential Models*		*Dynamic Graph Models*		*Time-informed Graph Models*		*Time-informed Graph +Sequential Models*		**TrajLift (Ours)**
	LSTM$^+$	Transformer$^+$	TGN	TGAT	TrajGAT$^+$	Graphormer$^+$	GAT$^+$ w/ LSTM	R-GCN$^+$ w/ LSTM	
ST-XOR	$67.78_{\pm 0.43}$	$54.22_{\pm 4.13}$	$75.56_{\pm 5.35}$	$\underline{99.78}_{\pm 0.16}$	$\underline{95.89}_{\pm 1.03}$	$73.56_{\pm 2.11}$	$\underline{98.78}_{\pm 0.42}$	$\underline{99.44}_{\pm 0.31}$	$\underline{98.00}_{\pm 0.98}$
ST-XOR-w	$52.56_{\pm 0.96}$	$50.11_{\pm 2.74}$	$51.67_{\pm 2.13}$	$\underline{91.56}_{\pm 1.10}$	$82.78_{\pm 1.34}$	$55.00_{\pm 3.34}$	$\underline{99.33}_{\pm 0.51}$	$\underline{98.78}_{\pm 0.96}$	$\underline{99.22}_{\pm 0.85}$
ST-XOR-wh	$49.78_{\pm 0.31}$	$49.44_{\pm 2.38}$	$49.33_{\pm 2.99}$	$47.44_{\pm 2.08}$	$49.56_{\pm 0.68}$	$50.67_{\pm 1.36}$	$48.89_{\pm 1.57}$	$49.56_{\pm 0.68}$	$\underline{98.67}_{\pm 0.57}$

Table 4: NC-vs-NCD classification on HKLLT delayed recall (%), mean ± SD over 15 splits. Paired comparisons are shown in Figure 2. Sens. = sensitivity. Best result per column in bold.

	HKLLT-4			HKLLT-5		
Model	Acc	F1	Sens.	Acc	F1	Sens.
NCD Detection Baselines						
Recall count (LR)	$74.96_{\pm 4.59}$	$74.02_{\pm 4.36}$	$\mathbf{73.55}_{\pm 11.71}$	$75.90_{\pm 4.39}$	$75.19_{\pm 4.53}$	$74.84_{\pm 7.34}$
Low-level features (LR, 7-D)	$74.79_{\pm 4.07}$	$73.99_{\pm 4.28}$	$73.12_{\pm 8.93}$	$75.90_{\pm 3.97}$	$75.07_{\pm 4.00}$	$73.33_{\pm 7.26}$
BERT	$72.05_{\pm 5.10}$	$70.00_{\pm 5.20}$	$58.06_{\pm 6.90}$	$74.19_{\pm 3.94}$	$72.33_{\pm 4.32}$	$61.29_{\pm 7.42}$
RoBERTa	$75.56_{\pm 4.56}$	$73.51_{\pm 5.17}$	$60.97_{\pm 9.35}$	$73.76_{\pm 3.77}$	$71.74_{\pm 4.30}$	$60.11_{\pm 9.03}$
Trajectory Learning Baselines						
LSTM$^+$	$73.85_{\pm 4.46}$	$71.54_{\pm 5.27}$	$58.92_{\pm 11.53}$	$73.46_{\pm 4.58}$	$71.75_{\pm 4.95}$	$62.37_{\pm 8.81}$
Transformer$^+$	$75.00_{\pm 3.36}$	$72.45_{\pm 4.26}$	$57.10_{\pm 8.68}$	$73.50_{\pm 5.12}$	$71.32_{\pm 5.74}$	$58.71_{\pm 10.05}$
TGN	$69.02_{\pm 4.34}$	$64.41_{\pm 5.51}$	$42.80_{\pm 10.38}$	$70.47_{\pm 4.76}$	$66.58_{\pm 6.40}$	$47.96_{\pm 12.94}$
TGAT	$67.53_{\pm 5.81}$	$63.04_{\pm 6.13}$	$47.20_{\pm 7.80}$	$68.95_{\pm 4.75}$	$63.99_{\pm 5.38}$	$47.25_{\pm 8.80}$
TrajGAT$^+$	$74.62_{\pm 3.01}$	$72.01_{\pm 3.64}$	$56.13_{\pm 7.35}$	$75.13_{\pm 5.24}$	$73.63_{\pm 5.48}$	$64.95_{\pm 7.89}$
Graphormer$^+$	$74.57_{\pm 3.94}$	$72.43_{\pm 4.57}$	$59.78_{\pm 9.59}$	$75.56_{\pm 3.42}$	$73.79_{\pm 3.94}$	$63.66_{\pm 9.28}$
GAT$^+$ w/ LSTM	$74.49_{\pm 4.54}$	$72.04_{\pm 5.17}$	$57.53_{\pm 9.22}$	$73.33_{\pm 4.15}$	$71.04_{\pm 5.13}$	$58.60_{\pm 11.22}$
R-GCN$^+$ w/ LSTM	$75.77_{\pm 3.76}$	$73.67_{\pm 4.30}$	$60.54_{\pm 7.95}$	$72.99_{\pm 5.14}$	$71.09_{\pm 5.72}$	$60.32_{\pm 9.10}$
TrajLift (Ours)	$\mathbf{77.52}_{\pm 5.63}$	$\mathbf{76.22}_{\pm 5.97}$	$69.35_{\pm 10.42}$	$\mathbf{80.38}_{\pm 4.63}$	$\mathbf{79.67}_{\pm 4.66}$	$\mathbf{78.92}_{\pm 8.63}$

increases: both degrade to chance level on ST-XOR-w and ST-XOR-wh (49–55%). (c) Local aggregation fails on deeper hierarchies: TGAT, TrajGAT$^+$, and the hybrid baselines (GAT$^+$/R-GCN$^+$ with LSTM) achieve strong performance on two-level hierarchies (>82% on ST-XOR and ST-XOR-w), as neighborhood aggregation enables perception of immediate cluster structure.

However, all collapse to chance level on ST-XOR-wh (47–50%), where discrimination requires perceiving super-cluster relationships that lie beyond their local receptive fields. (d) TrajLift maintains consistent performance: Our method achieves 98–99% accuracy across all variants, including ST-XOR-wh, where all baselines fail. We attribute it to the fact that spectral lifting provides explicit access to multiple hierarchical scales through diffusion time, enabling robust structure-time reasoning on both shallow and deep hierarchies.

Real-world Detection. We first consider the results on HKLLT-5. On this trial, TrajLift achieves the best accuracy, F1, and sensitivity (Table 4), and Figure 2 shows that its paired accuracy gain is significant over three of the four compared baselines, holding on 13–15 of the 15 splits. In accuracy, TrajLift outperforms both handcrafted baselines by 4.5 percentage points, whereas adding mean latency, semantic clustering, category transitions, and demographic features to recall count brings no improvement. Such statistics aggregate over the whole trajectory, while TrajLift encodes each pause in the hierarchical context in which it occurs. On HKLLT-4, the paired accuracy differences

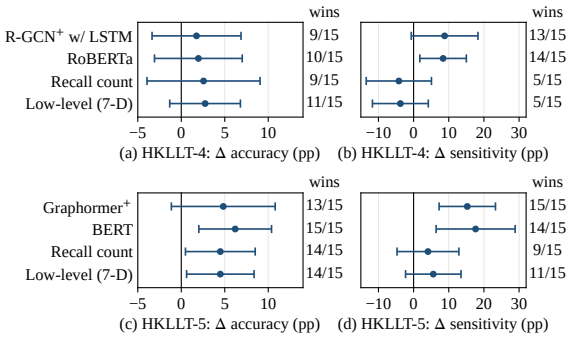


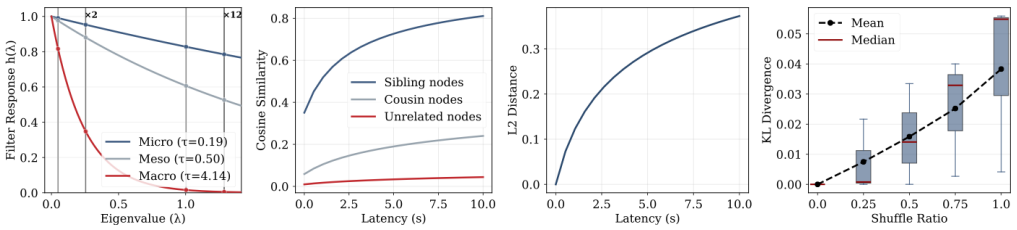
Figure 2: Paired differences between TrajLift and selected strong baselines, with 95% confidence intervals. Wins: splits on which TrajLift is better.

Table 5: Test-time latency perturbation (accuracy, %), mean ± SD over an independent set of 15 splits. Parentheses: change from the unperturbed input.

Input latencies	HKLLT-4	HKLLT-5
Original	$77.01_{\pm 5.68}$	$80.85_{\pm 4.73}$
$\delta_k \to 0$	$73.08_{\pm 4.05}$ (-3.93)	$74.02_{\pm 2.35}$ (-6.84)
$\delta_k \to \bar{\delta}$	$69.83_{\pm 6.94}$ (-7.18)	$63.59_{\pm 5.03}$ (-17.26)

are not significant, despite TrajLift still achieving the highest mean accuracy and F1 and being better on 9–11 of the 15 splits. Next, we examine why the two trials differ.

Differences Between Trials. To understand why TrajLift behaves differently on the two trials, which share the word list, participants, and hierarchy, we compare how strongly each signal separates the NC and NCD groups (Cohen's $|d|$, computed from the statistics in Appendix C). Recall count separates the groups similarly in both trials (1.44 vs. 1.50) and semantic clustering moderately more on HKLLT-5 (0.74 vs. 0.93), whereas the separation by mean inter-item latency roughly doubles (0.32 vs. 0.64). If TrajLift's advantage on HKLLT-5 stems from this timing signal, removing timing information at test time should hurt accuracy more on HKLLT-5 than on HKLLT-4. Table 5 shows that setting all latencies to zero or to their mean degrades accuracy on both trials, but the drop on HKLLT-5 is about twice as large (6.8 and 17.3 vs. 3.9 and 7.2 percentage points), mirroring the difference in latency effect sizes. These results suggest that TrajLift's advantage scales with the timing signal available in the data: it matches recall count alone when timing carries little additional information and exceeds it when timing becomes more informative.



(a) Spectral filter response (b) Structure-aware similarity (c) Time modulation effect (d) Time order sensitivity

Figure 3: (a) Heat kernel filter response at three scales. (b) Diffused representations preserve the ultrametric distance in cosine similarity. (c) Temporal modulation effect: embedding distance increases with latency. (d) Time order sensitivity: prediction divergence grows with shuffle ratio. All panels use a TrajLift model trained on one HKLLT-4 split.

5.3 Analysis

We now examine what TrajLift learns with a model trained on HKLLT-4: whether its representations reflect the semantic hierarchy, and how they respond to retrieval timing.

Does TrajLift capture hierarchical structure? Figure 3 visualizes the learned spectral filters and their effect on node representations. Panel (a) shows the heat kernel response $h(\lambda) = e^{-\tau\lambda}$ at three scales. The learned diffusion times exhibit clear separation: micro (τ=0.19) preserves high-frequency modes for intra-category distinctions, while macro (τ=4.14) retains only low frequencies capturing cross-domain structure. The learned filters thus span the hierarchy, from item-level detail at the micro scale to the domain split at the macro scale. Panel (b) shows cosine similarity between a node and others at different hierarchical distances. Same-category nodes maintain high similarity, while unrelated nodes remain distant, demonstrating that the learned representations faithfully encode the ultrametric substrate structure.

Is TrajLift sensitive to timing? Figure 3 examines the model's sensitivity to temporal information. Panel (c) shows the L2 distance between modulated and unmodulated embeddings as latency varies. The monotonic increase confirms that temporal modulation effectively encodes continuous-valued latencies into the embedding space. Panel (d) measures prediction sensitivity to temporal order via a shuffle test. We progressively shuffle latency sequences and compute KL divergence from original predictions. The increasing divergence with shuffle ratio demonstrates that the model relies on temporal ordering, not just marginal latency statistics.

6 Conclusion

We introduced TrajLift, a spectral framework that models verbal recall as timed trajectories on a semantic hierarchy. TrajLift lifts each recalled word onto the hierarchy by heat diffusion and lets the latency after each word modulate the diffusion time, so that structure and timing are encoded by a single operator. Our formal analysis shows that the resulting representation is first-order separable into structural and temporal components, with spectral selectivity across hierarchical levels. On synthetic benchmarks that require structure-time interaction, TrajLift remains near-perfect on deep hierarchies where all baselines fail. On real-world clinical data, it outperforms both conventional clinical measures and computational baselines, and its advantage is clearest when retrieval timing is more informative. More broadly, we believe TrajLift offers a perspective in which clinical knowledge of cognitive impairment could guide representation learning as an inductive bias, as a step toward more interpretable cognitive screening.

Limitations

The spectral analysis in this work assumes a balanced hierarchical partition, which holds for the HKLLT test protocol used in our experiments. Applications to domains where the semantic hierarchy is unbalanced—for instance, taxonomies with varying branching factors—would require extending the theoretical analysis, although empirically balance appears sufficient rather than necessary (Appendix B). On the empirical side, each ST-XOR variant places the structure-time interaction at a single hierarchical level, so the benchmark does not test whether combining multiple scales is necessary. Clinical datasets for AD screening are inherently small, limiting the statistical power and generalizability of the real-world experiments, and our NCD labels are syndromic rather than specific to AD. Besides, the data used in this study were collected from a local clinical population, and demographic or cultural differences across populations may introduce distributional shifts that we have not evaluated. Validating TrajLift on larger, multi-site cohorts and on other domains involving timed trajectories on hierarchical structures (e.g., navigation in knowledge graphs, structured browsing behavior) is a natural direction for future work.

Broader Impacts

This work is at the research stage and has not been deployed in any clinical or real-world setting. This work develops computational tools for early screening of cognitive impairment from verbal memory tests, which are non-invasive, low-cost, and widely administered. If validated at scale, such tools could support early detection in resource-limited settings where access to neuroimaging or biomarker testing is scarce. However, the model is intended as a screening aid, not a substitute for clinical diagnosis: false positives may cause unnecessary anxiety, while false negatives may delay intervention. As noted above, the model has been evaluated on a single local cohort, and deployment across different demographic, cultural, or linguistic populations requires careful re-validation to avoid systematic misdiagnosis. More broadly, cognitive assessment data is sensitive medical information; safeguards against unauthorized use are essential before any clinical deployment.

Ethical Statement

The clinical data used in this study were collected under a protocol approved by the relevant institutional ethics committee. All participants provided informed consent prior to participation. All data were de-identified before analysis, and no personally identifiable information is included in this paper or its supplementary materials.

Acknowledgments and Disclosure of Funding

We thank Dr. Jingyan Zhou for insightful discussions that shaped the framing and positioning of this work. This work is supported by the HKSARG Research Grants Council's Theme-based Research Grant Scheme (Project No. T45-407/19N), the CUHK Stanley Ho Big Data Decision Research Centre, and the Centre for Perceptual and Interactive Intelligence (CPII) Ltd, a CUHK-led InnoCentre under the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government.

References

[1] Alex Martin, Pim Brouwers, Christiane Cox, and Paul Fedio. On the nature of the verbal memory deficit in alzheimer's disease. *Brain and Language*, 25(2):323–341, 1985.

[2] Dean C Delis, Joel H Kramer, Edith Kaplan, and Beth A Ober. *California Verbal Learning Test–Second Edition (CVLT-II)*. The Psychological Corporation, San Antonio, TX, 2000.

[3] Alma Au, Agnes S Chan, and Helen Chiu. Verbal learning in alzheimer's dementia. *Journal of the International Neuropsychological Society*, 9(3):363–375, 2003.

[4] Allan M Collins and M Ross Quillian. Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, 8(2):240–247, 1969.

[5] Timothy EJ Behrens, Timothy H Muller, James CR Whittington, Shirley Mark, Alon B Baram, Kimberly L Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018.

[6] Paul J Gruenewald and Gregory R Lockhead. The free recall of category examples. *Journal of Experimental Psychology: Human Learning and Memory*, 6(3):225, 1980.

[7] Johannes Tröger, Nicklas Linz, Alexandra König, Philippe Robert, Jan Alexandersson, Jessica Peter, and Jutta Kray. Exploitation vs. exploration—computational temporal and semantic analysis explains semantic verbal fluency impairment in alzheimer's disease. *Neuropsychologia*, 131:53–61, 2019.

[8] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next: modeling user behaviors by time-lstm. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 3602–3608. AAAI Press, 2017. ISBN 9780999241103.

[9] 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.

[10] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

[11] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based Recommendation with Graph Neural Networks. In Pascal Van Hentenryck and Zhi-Hua Zhou, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, volume 33 of *AAAI '19*, pages 346–353, July 2019. doi: 10.1609/aaai.v33i01.3301346. URL `https://aaai.org/ojs/index.php/AAAI/article/view/3804`.

[12] Di Yao, Haonan Hu, Lun Du, Gao Cong, Shi Han, and Jingping Bi. Trajgat: A graph-based long-term dependency modeling approach for trajectory similarity computation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2275–2285, 2022.

[13] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4805–4815, Red Hook, NY, USA, 2018. Curran Associates Inc.

[14] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *Proceedings of the 36th International Conference on Machine Learning*, 09–15 Jun 2019.

[15] A Chan and I Kwok. Hong Kong List Learning Test (HKLLT): Manual and preliminary norms. *Department of Psychology, The Chinese University of Hong Kong, Hong Kong*, 1998.

[16] Marina Gasparini, Michele Scandola, Stefania Amato, Emanuela Salati, Elena Facci, Valeria Gobbetto, Giuseppe Bruno, Nicola Vanacore, Giuseppe Gambina, and Valentina Moro. Normative data beyond the total scores: a process score analysis of the rey's 15 word test in healthy aging and alzheimer's disease. *Neurological Sciences*, 45(6):2605–2613, 2024.

[17] Ryien Hosseini, Filippo Simini, Venkatram Vishwanath, and Henry Hoffmann. A deep probabilistic framework for continuous time dynamic graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17249–17257, 2025.

[18] Flavio Corradini, Flavio Gerosa, Marco Gori, Carlo Lucheroni, Marco Piangerelli, and Martina Zannotti. A systematic literature review of spatio-temporal graph neural network models for time series forecasting and classification. *Neural Networks*, page 108269, 2025.

[19] Xiao Zhang, Xingyu Zhao, Hong Xia, Yuan Cao, Guiyuan Jiang, Junyu Dong, and Yanwei Yu. Trajdiff: Diffusion bridge network with semantic alignment for trajectory similarity computation. *arXiv preprint arXiv:2506.15898*, 2025.

[20] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

[21] Ahsan Shehzad, Feng Xia, Shagufta Abid, Ciyuan Peng, Shuo Yu, Dongyu Zhang, and Karin Verspoor. Graph transformers: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

[22] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

[23] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2): 298–305, 1973.

[24] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[25] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

[26] Ling Ding, Chao Li, Di Jin, and Shifei Ding. Survey of spectral clustering based on graph theory. *Pattern Recognition*, 151:110366, 2024.

[27] Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International conference on machine learning*, pages 1407–1418. PMLR, 2021.

[28] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. *Advances in neural information processing systems*, 32, 2019.

[29] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003.

[30] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[32] Heiko Braak and Eva Braak. Neuropathological stageing of alzheimer-related changes. *Acta neuropathologica*, 82(4):239–259, 1991.

A Proof of Proposition 1

We prove Proposition 1 for a balanced two-level hierarchy and then describe how the result extends recursively to deeper balanced hierarchies.

A.1 Two-Level Case

Setup. Consider $N = Cn$ nodes partitioned into C clusters of size n. We set $W_{ii} = 0$, so weights are assigned only to distinct node pairs: all pairs within the same cluster receive weight w_1, and all pairs in different clusters receive weight w_2, with $\rho = w_1/w_2 > 1$. By the balanced block structure, the graph is regular with degree

$$d \;=\; (n-1)w_1 + n(C-1)w_2,$$

and the normalized Laplacian simplifies to $L = I - W/d$.

Eigenvalue structure. The block symmetry of W implies that its eigenspaces decompose into three invariant subspaces.

Trivial mode (multiplicity 1). The constant vector gives $\lambda_0 = 0$.

Coarse modes (multiplicity $C-1$). These eigenvectors are constant within each cluster and orthogonal to the constant vector. For such a vector $\mathbf{u}$, if node i lies in cluster a with cluster-level value c_a and $\sum_b c_b = 0$, then

$$(W\mathbf{u})_i \;=\; (n-1)w_1\, c_a + nw_2 \sum_{b\neq a} c_b \;=\; \big((n-1)w_1 - nw_2\big)\, c_a.$$

The corresponding Laplacian eigenvalue is

$$\lambda_{\text{coarse}} \;=\; 1 - \frac{(n-1)w_1 - nw_2}{d} \;=\; \frac{nCw_2}{d}.$$

Fine modes (multiplicity $C(n-1)$). These eigenvectors sum to zero within each cluster. For such a vector, inter-cluster contributions vanish (since each cluster's entries sum to zero), and within-cluster contributions satisfy $(W\mathbf{u})_i = -w_1 u_i$. The corresponding Laplacian eigenvalue is

$$\lambda_{\text{fine}} \;=\; 1 + \frac{w_1}{d} \;=\; \frac{nw_1 + n(C-1)w_2}{d}.$$

Spectral gap. Writing the eigenvalues in terms of $\rho = w_1/w_2$:

$$\lambda_{\text{coarse}} \;=\; \frac{nC}{(n-1)\rho + n(C-1)}, \qquad \lambda_{\text{fine}} \;=\; \frac{n(\rho + C - 1)}{(n-1)\rho + n(C-1)}.$$

The gap between the two non-trivial eigenspaces is

$$\gamma \;=\; \lambda_{\text{fine}} - \lambda_{\text{coarse}} \;=\; \frac{n(\rho-1)}{(n-1)\rho + n(C-1)}.$$

When $\rho \gg C$, the coarse eigenvalue is $O(C/\rho)$, while the fine eigenvalue and the gap are both $O(1)$. This confirms that for sufficiently large weight ratios, the Laplacian spectrum separates into two distinct bands corresponding to coarse (cluster-level) and fine (within-cluster) structure.

Heat kernel behavior. The heat kernel $H_\tau = e^{-\tau L}$ attenuates each eigenmode by $e^{-\tau\lambda_i}$. When $\rho \gg C$, there exists a non-empty intermediate range $1 \ll \tau \ll \rho/C$, and the two non-trivial bands respond differently:

- Coarse modes: attenuated by $e^{-\tau\lambda_{\text{coarse}}}$. Since $\lambda_{\text{coarse}} = O(C/\rho)$, this factor remains close to 1 as long as $\tau \ll \rho/C$.
- Fine modes: attenuated by $e^{-\tau\lambda_{\text{fine}}}$. Since $\lambda_{\text{fine}} = O(1)$, this factor becomes small when $\tau \gg 1$.

This yields three regimes of the diffusion time:

1. $\tau \ll 1$: both bands are preserved, and the lifted representation retains full within-cluster detail.
2. $1 \ll \tau \ll \rho/C$: fine modes are suppressed while coarse modes persist, yielding a cluster-level representation.
3. $\tau \gg \rho/C$: all non-trivial modes are suppressed, and the representation collapses to the uniform graph mean (which is the unique stationary component on a regular graph).

Therefore, increasing τ continuously interpolates from item-level to cluster-level representations, with the transition governed by the spectral gap γ. □

A.2 Multi-Level Extension

For a balanced nested hierarchy in which all groups at the same level have equal size and edge weights depend only on the ultrametric distance (Definition 1), the partition at each level is equitable. The subspaces of vectors that are constant on blocks at one level and sum to zero within their parent block are invariant under W and hence under L. The exact eigenvalues in the multi-level case depend on the branching factors and all weights across levels. Nevertheless, applying the two-level argument to each adjacent refinement shows that, when adjacent-level ratios $\rho_\ell = w(\ell)/w(\ell+1)$ are sufficiently separated, these invariant subspaces form spectral bands associated with successive hierarchy refinements, together with the trivial zero eigenvalue. Heat diffusion then suppresses these bands from fine to coarse as τ increases.

In our experimental setting, the HKLLT hierarchy is balanced: 2 domains $\times$ 2 categories per domain $\times$ 4 items per category, with weight ratios $w(1)/w(2) = 10$ and $w(2)/w(3) = 10$. These ratios are chosen to encourage separated spectral bands, which we target with the micro, meso, and macro diffusion scales defined in §3.2.

B Robustness on Synthetic Benchmarks

We examine three aspects of the ST-XOR-wh results: hierarchy balance, timestamp noise, and the choice of diffusion scales.

B.1 Unbalanced Hierarchies

Proposition 1 and Appendix A assume a balanced partition. We relax this assumption on ST-XOR-wh by drawing cluster sizes unevenly at a fixed total number of nodes. As Table 6 shows, TrajLift's accuracy remains around 99% as imbalance grows. Balance thus appears sufficient rather than necessary.

Table 6: Unbalanced hierarchies on ST-XOR-wh. CV: coefficient of variation of cluster sizes.

CV of cluster size	0	0.232	0.524	0.970
TrajLift accuracy (%)	98.67±0.57	98.67±1.18	98.87±0.61	99.00±0.41

B.2 Corrupted Timestamps

We perturb latencies multiplicatively at test time, $\delta \to \delta\varepsilon$ with $\log \varepsilon \sim \mathcal{N}(0, \sigma^2)$. To express noise strength on an interpretable scale, we report the discriminability of the fast and slow latency distributions that define the two classes, $d'(\sigma) = 1.80/\sqrt{0.184^2 + \sigma^2}$, where 1.80 is their separation and 0.184 their pooled standard deviation on the log scale, together with their overlap $2\Phi(-d'/2)$. As Table 7 shows, TrajLift retains 95% accuracy when the two distributions overlap by about a quarter, and approaches chance only when $d' < 1$, where the distributions themselves are barely separable.

B.3 Choice of Diffusion Scales

ST-XOR-wh has three hierarchy levels, matching the three diffusion scales of TrajLift. To test whether this match drives the result, we train TrajLift with each of the seven non-empty subsets of

Table 7: Test-time timestamp perturbation on ST-XOR-wh.

σ	0	0.25	0.50	0.75	1.00	1.50	2.00
d'	9.77	5.79	3.37	2.33	1.77	1.19	0.90
Overlap (%)	0	0.4	9.2	24.4	37.7	55.2	65.4
TrajLift accuracy (%)	98.67	99.13	97.87	95.00	79.40	60.80	53.53

{micro, meso, macro}. As Table 8 shows, every subset containing the macro scale reaches about 99%, regardless of the number of scales, whereas subsets without it fail or degrade. Performance thus depends on whether a scale resolves the level that carries the signal, not on the number of scales. With {micro} alone, the heat kernel is close to the identity ($\tau \approx 0.12$), and accuracy falls to the level of the structure-agnostic LSTM$^+$ (Table 3).

Table 8: TrajLift with subsets of diffusion scales on ST-XOR-wh (accuracy, %), mean $\pm$ SD over five random seeds.

Scales	Number	Accuracy
{macro}	1	99.67±0.75
{meso, macro}	2	99.47±0.30
{micro, macro}	2	99.27±0.60
{micro, meso, macro}	3	98.67±0.57
{meso}	1	94.47±3.74
{micro, meso}	2	53.20±2.33
{micro}	1	52.73±3.93

C Data Details

C.1 Hong Kong List Learning Test (HKLLT)

The Hong Kong List Learning Test (HKLLT) [15] is a standardized neuropsychological assessment designed for Chinese-speaking populations to evaluate verbal learning and episodic memory. Episodic memory impairment is among the earliest cognitive deficits observed in Alzheimer's disease, reflecting neuropathological changes in the hippocampus and entorhinal cortex that occur in the initial stages of the disease [32]. As the disease progresses, the frontal, temporal, and parietal association cortices become increasingly involved, leading to broader deficits in lexical-semantic functions [3]. The HKLLT is specifically designed to probe these capacities through the recall of a 16-word list organized under a three-level semantic hierarchy, as detailed in Table 9.

Table 9: HKLLT word list and semantic hierarchy. The 16 target words are organized into four categories under two domains.

Domain	**Category**	**Items**
Abstract	Relatives	Grandmother, Aunt, Niece, Cousin
	Countries	India, Switzerland, Laos, Chile
Concrete	Vegetables	Eggplant, Cucumber, Mustard Greens, Onion
	Furniture	Desk, Mirror, Lamp, Wardrobe

The full protocol comprises five trials: Trial 1 presents the 16 words orally in randomized order for immediate free recall; Trials 2–3 repeat the same list to measure learning rate; Trial 4 assesses delayed recall after a 10-minute non-verbal interval; Trial 5 assesses delayed recall after 30 minutes; a final recognition phase tests discrimination between targets and distractors.

In this work, we focus exclusively on Trial 4 and Trial 5. Clinical research has established that delayed recall is particularly effective for early-stage AD detection, as the rate of forgetting over these intervals is the most sensitive discriminator between patients with mild cognitive impairment and cognitively normal elderly [3].

C.2 Participant Demographics

Table 10 summarizes participant characteristics. The NC and NCD groups differ in age and education: the NCD group is older and has fewer years of education (Welch's t-test, NC vs. NCD: $t = -4.84$ for age and $t = 3.67$ for education; both $p < 0.001$), consistent with known epidemiological patterns. TrajLift receives no demographic input, and the low-level feature baseline (§5.1) includes both variables.

Table 10: Demographic characteristics of participants. Continuous variables are reported as mean $\pm$ SD; gender as Male/Female counts.

	NC	NCD
Number	233	154
Age (years)	70.8 ± 6.5	74.5 ± 7.8
Education (years)	8.8 ± 4.1	7.2 ± 4.4
Gender (M/F)	98/135	74/80

C.3 Human Subjects and Data Collection Protocol

The clinical cohort was collected under a protocol approved by the institutional research ethics committee. Data were collected by trained personnel following the standard HKLLT administration procedure. All participants provided informed consent and were compensated for their participation. The data contain protected health information and are not publicly released.

C.4 Measure Validation

To validate that our dataset exhibits expected patterns, we examine three commonly used measures via independent-samples t-tests (Table 11):

- **Semantic Clustering:** number of consecutive same-category recalls, reflecting sensitivity to hierarchical organization of semantic memory.
- **Recall Count:** total correctly recalled words (max 16), indexing overall retention capacity.
- **Recency Effect:** proportion of last-four-list items among recalls, measuring serial position effects.

Table 11: Measure validation via group comparison (t-test). Values are mean $\pm$ SD.

Measure	**Trial**	**NC**	**NCD**	t	p
Semantic Clustering	T4	3.33 ± 2.99	1.38 ± 2.23	6.91	$< .001$
	T5	2.99 ± 2.61	1.02 ± 1.47	8.48	$< .001$
Recall Count	T4	6.82 ± 2.50	3.21 ± 2.50	13.87	$< .001$
	T5	6.36 ± 2.62	2.69 ± 2.25	14.18	$< .001$
Recency Effect	T4	0.12 ± 0.12	0.12 ± 0.20	-0.02	.984
	T5	0.12 ± 0.11	0.13 ± 0.22	-0.65	.517

Both semantic clustering and recall count show significant group differences ($p < 0.001$), consistent with clinical findings that patients with AD recall fewer words and exhibit reduced categorical organization. Recency effects show no significant difference ($p > 0.5$), as expected in delayed recall, where the recency advantage dissipates.

C.5 Trajectory Statistics

Table 12 reports trajectory-level statistics by trial and group, computed per participant and averaged within each group. Specifically, sequence length counts all recalled items, including repetitions; mean inter-item latency averages the latencies δ_k within a trajectory; category transitions counts consecutive recalls from different categories, complementing semantic clustering (Appendix C.4); category coverage counts the distinct categories recalled; word coverage counts the distinct words recalled.

Table 12: Trajectory statistics by trial and group. Values are mean ± SD.

Statistic	Trial	NC	NCD
Sequence Length	T4	9.06 ± 4.60	4.52 ± 4.24
	T5	7.76 ± 3.86	3.42 ± 3.06
Mean Inter-Item Latency (s)	T4	2.70 ± 1.38	3.27 ± 2.14
	T5	2.39 ± 1.25	3.49 ± 2.08
Word Coverage	T4	6.82 ± 2.50	3.21 ± 2.50
	T5	6.36 ± 2.62	2.69 ± 2.25
Category Coverage	T4	3.57 ± 0.65	2.21 ± 1.40
	T5	3.40 ± 0.83	1.94 ± 1.38
Category Transitions	T4	4.73 ± 2.52	2.34 ± 2.37
	T5	3.79 ± 1.99	1.65 ± 1.84

D Implementation Details

D.1 TrajLift Configurations

D.1.1 Diffusion Time Initialization

The base diffusion time at each scale is initialized from the largest eigenvalue of L:

$$\tau_s^{\text{base}} = \frac{-\ln(\alpha_s)}{\lambda_{\max}}, \tag{7}$$

where $\alpha_s \in (0, 1)$ is a target retention rate controlling how much of the highest-frequency component survives at scale s. We set $\alpha_{\text{micro}} = 0.85$, $\alpha_{\text{meso}} = 0.25$, and $\alpha_{\text{macro}} = 0.002$, corresponding to local, community-level, and global diffusion respectively. High retention (micro) preserves within-cluster fine structure, while low retention (macro) suppresses all but the coarsest modes.

D.1.2 Modulation Coefficient Initialization

The modulation coefficient γ_s in Equation (3) controls the range of the latency-induced perturbation $\Delta\tau_k^{(s)} = \tau_s^{\text{base}} \cdot \gamma_s \cdot \varphi(\delta_k)$. We initialize γ_s from training set latency statistics:

$$\gamma_s = \frac{r - 1}{\varphi(\delta_{90})}, \tag{8}$$

where $r = 3.0$ is a target variation ratio and δ_{90} is the 90th percentile of inter-item latencies in the training set. This ensures that for the large majority of observed latencies, the modulated diffusion time $\tilde{\tau}_k^{(s)}$ stays within a factor of r of the base time: $\tilde{\tau}_k^{(s)} \leq r \cdot \tau_s^{\text{base}}$ for $\delta_k \leq \delta_{90}$. We apply $\varphi(\delta) = \log(1 + \delta)$ with latencies clipped at 20 seconds to prevent extreme values from producing excessive modulation.

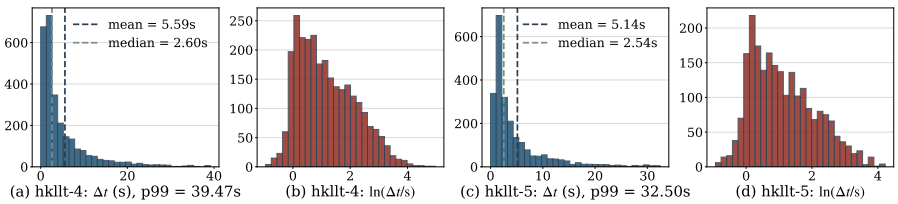


Figure 4: Distribution of inter-item latencies in the HKLLT dataset. (a,c) Raw latencies δ_k for Trials 4 and 5: the distributions are heavily right-skewed (mean $\gg$ median) with 99th percentiles exceeding 30s. (b,d) Log-transformed latencies $\ln(\delta_k)$: the distributions become approximately symmetric. Latencies exceeding 20s are clipped before transformation.

Choice of φ. Figure 4 shows the distribution of inter-item latencies in the HKLLT data. The raw latencies are heavily right-skewed (mean $\approx$ 5.6s versus median $\approx$ 2.6s for Trial 4, with the

99th percentile exceeding 39s), so a linear mapping would allow rare extreme pauses to dominate the modulation. After a log transform, the distribution becomes approximately symmetric, so φ concentrates representational sensitivity on the frequent short-to-moderate latency range while compressing the tail.

D.1.3 Spectral Gap Margin Verification

We verify the range of the latency-induced modulation claimed in §3.3.

From the eigenvalue analysis in Appendix A, the spectral gap between the fine and coarse eigenvalue bands of our HKLLT graph is

$$\gamma \;=\; \frac{n(\rho-1)}{(n-1)\rho + n(C-1)},$$

where n is the cluster size, C is the number of clusters, and ρ is the weight ratio at the relevant hierarchy level. For the category level ($n=4$, $C=4$, $\rho=10$), this gives $\gamma = \frac{4\times 9}{3\times 10+4\times 3} = \frac{36}{42} \approx 0.86$. The inverse spectral gap $1/\gamma \approx 1.17$ sets the scale at which modulation would begin to cross eigenvalue bands.

The maximum modulation at scale s is

$$\Delta\tau_{\max}^{(s)} \;=\; \tau_s^{\text{base}} \cdot \gamma_s \cdot \varphi(\delta_{90}).$$

By construction (Equation (8)), $\gamma_s \cdot \varphi(\delta_{90}) = r - 1 = 2.0$, so $\Delta\tau_{\max}^{(s)} = 2\,\tau_s^{\text{base}}$.

For the micro scale, $\tau_{\text{micro}}^{\text{base}} = -\ln(0.85)/\lambda_{\max}$. With $\lambda_{\max} \approx n/(n-1) = 4/3$ (from Appendix A), we get $\tau_{\text{micro}}^{\text{base}} \approx 0.12$, and therefore $\Delta\tau_{\max}^{(\text{micro})} \approx 0.24$. Since $0.24 \ll 1/\gamma \approx 1.17$, the micro-scale modulation stays well within the spectral gap margin. For the meso and macro scales, $\Delta\tau_{\max}$ exceeds this margin; since the modulated diffusion time is bounded by $3\,\tau_s^{\text{base}}$ and adjacent base times differ by more than a factor of three, modulation still does not move one scale into the range of another.

D.1.4 Band Separation of Multi-Scale Residuals

Proposition 2 establishes that the temporal residual at scale s has a band-pass-shaped spectral response peaking at $\lambda^\star = 1/\tau_s^{\text{base}}$. For the three scales to capture non-redundant temporal information, these peaks should fall in different eigenvalue bands.

From the initialization above, the peak frequencies are:

$$\frac{1}{\tau_{\text{micro}}^{\text{base}}} \;=\; \frac{\lambda_{\max}}{-\ln(0.85)} \;\approx\; 8.2, \qquad \frac{1}{\tau_{\text{meso}}^{\text{base}}} \;\approx\; 0.96, \qquad \frac{1}{\tau_{\text{macro}}^{\text{base}}} \;\approx\; 0.21.$$

In our HKLLT graph, the fine eigenvalues are near $\lambda_{\text{fine}} \approx n/(n-1) \approx 1.33$ and the coarse eigenvalues are near $\lambda_{\text{coarse}} \approx C/\rho \approx 0.4$. The micro peak ($\approx 8.2$) lies above the fine band, the meso peak (≈ 0.96) lies near the fine band, and the macro peak (≈ 0.21) lies below the coarse band. These peaks therefore target different spectral regions, confirming that the three temporal residual channels capture temporal information at distinct hierarchical resolutions.

D.1.5 Architecture

The sequence encoder is a 2-layer LSTM with a hidden dimension of 16 and a dropout rate of 0.1. While other sequential architectures, such as Transformers, are compatible with our framework, we adopt LSTM due to its robustness under limited clinical data. After LSTM, we use a linear classifier for classification.

D.2 Baseline Configurations

D.2.1 Shared Settings

All baselines embed each node into 8 dimensions via one-hot encoding followed by linear projection. The hidden dimension is set to 16 across all models. For time-augmented variants (denoted with '+'), we use a Time2Vec dimension of 4. All other training configurations (learning rate, optimizer, etc.) are identical to the proposed method.

D.2.2 Trajectory Learning Baselines

Table 13 summarizes the configuration for each baseline model. All models use a dropout rate of 0.1.

Table 13: Baseline model configurations.

Model	Configuration
LSTM$^+$	2 layers
Transformer$^+$	2 layers, 2 attention heads, feed-forward dimension 32
TGN	memory dimension 16, message dimension 16
TGAT	2 attention layers, 2 attention heads
TrajGAT$^+$	2 GAT layers, 2 attention heads
Graphormer$^+$	2 layers, 2 attention heads, feed-forward dimension 32
GAT$^+$ w/ LSTM	2 GAT layers, 2 attention heads, 1-layer LSTM
R-GCN$^+$ w/ LSTM	2 GNN layers, 2 relation bases, 1-layer LSTM

D.2.3 NCD Detection Baselines

For PLM baselines, we use Chinese BERT[1] and Chinese RoBERTa[2] with frozen encoders. We train a classifier with a hidden dimension of 16 and a dropout rate of 0.1 on [CLS] token representations. For handcrafted baselines, we train logistic regression on the measures listed in §5.1 (defined in Appendix C). Adding all 21 pairwise interactions to the low-level feature set (28-D) yields 74.53 and 75.56 accuracy on HKLLT-4 and HKLLT-5, below the 7-D model.

D.2.4 Graph Inputs

This subsection clarifies the graph inputs used by different trajectory-learning baselines. The hierarchy-edge construction applies only to the static-graph baselines in categories (c) and (d) of §5.1, i.e., TrajGAT+, Graphormer+, GAT+ w/ LSTM, and R-GCN+ w/ LSTM. It does not apply to the dynamic graph baselines TGN and TGAT, which are evaluated in their native continuous-time dynamic-graph setting using temporal item-to-item events.

For each recall trajectory used by the static-graph baselines, we construct a trajectory-specific graph. Item nodes correspond to the unique recalled items in the trajectory. Temporal transition edges connect consecutive recalled items in recall order and carry the inter-item latency encoded by Time2Vec. To expose the same semantic hierarchy used by TrajLift, we additionally include category and domain nodes and add hierarchy edges from each item node to its parent category node and parent domain node. Thus, these baseline graphs contain two edge types: temporal transition edges and semantic hierarchy edges.

The representation of edge types follows each baseline's native interface. For R-GCN+ w/ LSTM, temporal transition edges and hierarchy edges are encoded as separate relation types. For GAT+ w/ LSTM and TrajGAT+, both edge types are included in the same message-passing graph; edge attributes distinguish temporal transition edges from hierarchy edges, with Time2Vec latency features used for transition edges and a zero-latency/type indicator used for hierarchy edges. For Graphormer+, the combined graph is used to compute structural encodings, and edge-type information is included through edge embeddings distinguishing transition and hierarchy edges. No hierarchy edges are added to TGN or TGAT because these models represent the input as a stream of timestamped interaction events rather than as message passing over a fixed semantic hierarchy.

D.3 Training Protocol

All models are trained using the Adam optimizer with a learning rate of 0.001 and weight decay of 0.0001. The batch size is set to 16. For synthetic data, we apply early stopping with a patience of 20 epochs based on validation accuracy, up to a maximum of 100 epochs. For the HKLLT data, we set the training epochs to 50. To address class imbalance, class weights are computed using inverse frequency. All experiments are implemented in PyTorch 2.1.1 and PyTorch Geometric 2.6.1.

[1] `https://huggingface.co/google-bert/bert-base-chinese`
[2] `https://huggingface.co/hfl/chinese-roberta-wwm-ext`

D.4 Existing Assets and Licenses

We use the following existing assets. For pretrained language-model baselines, we use bert-base-chinese from Google and chinese-roberta-wwm-ext from HFL via HuggingFace. These models are used only as frozen encoders for baseline comparisons. We cite the corresponding model sources and respect the licenses and terms of use distributed with the HuggingFace repositories (Appendix D.2.3). For implementation, we use PyTorch 2.1.1+cu118 and PyTorch Geometric 2.6.1, which are released under BSD-style licenses. For graph and sequence baselines, we implement the compared architectures following their original papers and cite the corresponding publications; no pretrained checkpoints from these baselines are used. The HKLLT clinical dataset is proprietary clinical data used under an IRB-approved protocol and confidentiality agreements; it is not redistributed or repackaged. The synthetic ST-XOR benchmarks are generated by the authors and are released with the code under the license specified in the repository.

D.5 Compute Resources

All experiments were conducted on RTX8000 with 48 GB GPU memory using PyTorch 2.1.1+cu118 and PyTorch Geometric 2.6.1. A single training run of TrajLift takes approximately 2-8 minutes on each synthetic benchmark and 2-6 minutes on each HKLLT trial.

E Sample Visualization

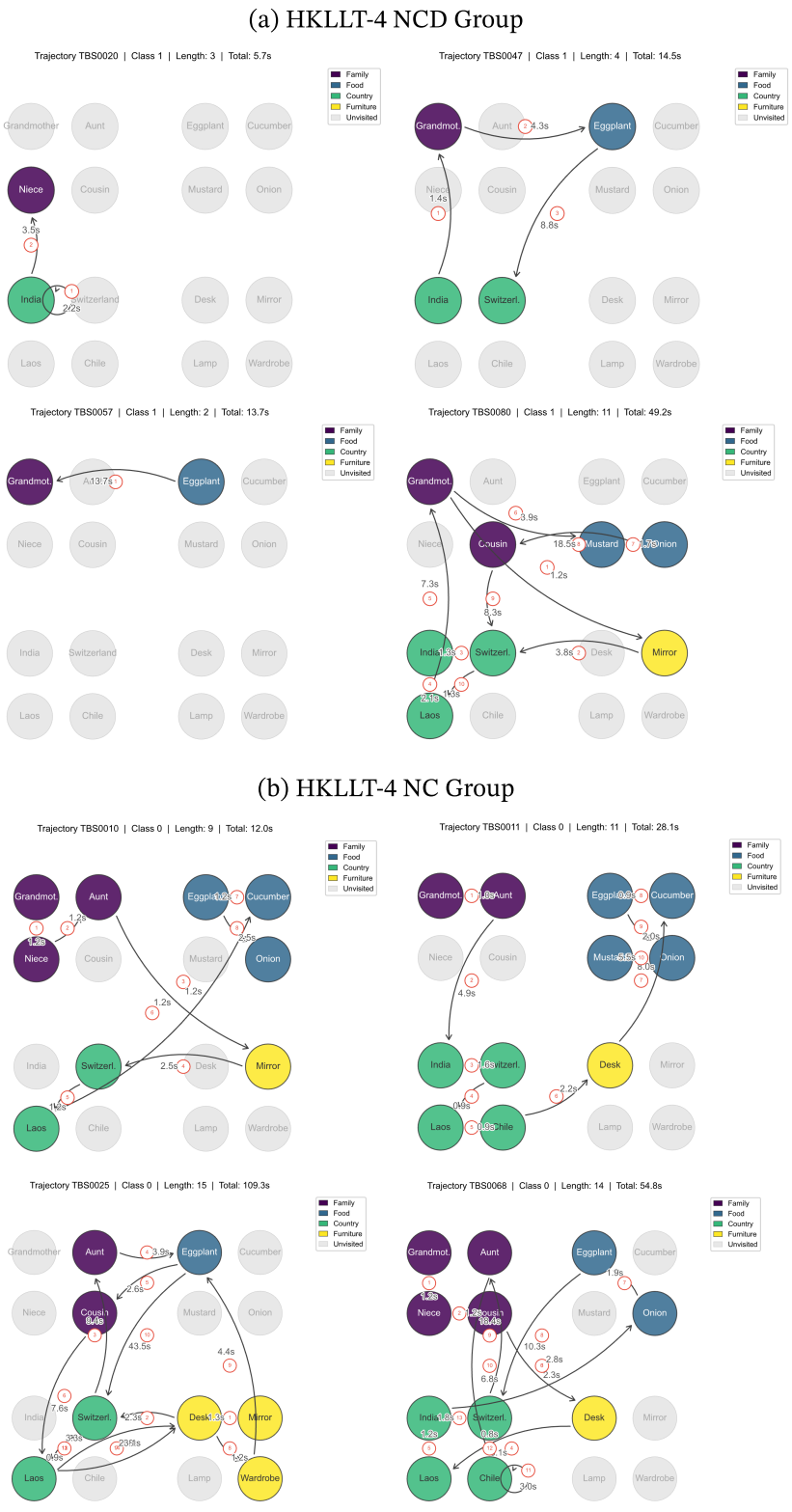


Figure 5: Example recall trajectories from HKLLT Trial 4 (Class 1 = NCD, Class 0 = NC). Words are laid out by category. Arrows follow recall order and are labeled with the inter-item latency in seconds.

(a) HKLLT-5 NCD Group



(b) HKLLT-5 NC Group



Figure 6: Example recall trajectories from HKLLT Trial 5 (Class 1 = NCD, Class 0 = NC). Words are laid out by category. Arrows follow recall order and are labeled with the inter-item latency in seconds.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are aligned with the proposed TrajLift framework, the theoretical analysis, and the empirical evaluations on synthetic benchmarks and a clinical HKLLT dataset. We also added the theoretical limitation statement in the Limitation section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A Limitation section was included to discuss the limitations of this work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper states theoretical results as propositions/lemmas and provides formal analysis in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide model architecture, data construction, preprocessing, train/validation/test splits, hyperparameter selection, baseline configurations, and evaluation protocols. The clinical data contain protected health information and cannot be released, but we provide detailed statistics and release the synthetic benchmark and code at https://github.com/kjw11/TrajLift.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The clinical data contain protected health information and cannot be released. The synthetic benchmark, data-generation code, model implementation, and experimental configurations are released at https://github.com/kjw11/TrajLift.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies data splits, metrics, validation protocol, hyperparameter selection, model configurations, optimizer, learning rate, batch size, number of epochs, etc., in the experimental section and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Synthetic experiments are repeated over five random seeds and HKLLT experiments over 15 stratified splits; tables report mean and standard deviation. Paired comparisons on HKLLT report Nadeau–Bengio corrected 95% confidence intervals.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the compute environment, including GPU/CPU type, memory, and approximate runtime per experiment for the reported experiments. The total compute used is hard to estimate due to server load fluctuation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The study follows the NeurIPS Code of Ethics. The clinical data are de-identified and handled under privacy and confidentiality requirements.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss potential positive impacts for scalable auxiliary cognitive screening as well as risks, including incorrect predictions, and privacy concerns in the Broader Impacts section.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We do not release protected clinical data. We state that TrajLift is intended only as an auxiliary screening research tool and should not be used for standalone clinical diagnosis without clinician oversight and external validation.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite and document the existing assets used in the paper, including pretrained BERT/RoBERTa models, software libraries, and the HKLLT protocol, and provide license or terms-of-use information where applicable.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The synthetic benchmark, generation scripts, and model code are released at https://github.com/kjw11/TrajLift with documentation in the repository README.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: The study does not involve crowdsourcing. For the clinical data, we describe the HKLLT protocol, inclusion criteria, participant demographics, and data collection procedure.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The clinical data collection and use of the HKLLT dataset were reviewed and approved by the institutional research ethics committee. The paper describes privacy protection, protected health information constraints, and clinical-use risks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as an important, original, or non-standard component of the core method. They were used only for language polishing, and no proprietary or sensitive clinical data were input into LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.