TraXion: Rethinking Pre-training Frameworks for Mobility and Beyond

Shang-Ling Hsu
University of Southern California
Los Angeles, California, USA
hsushang@usc.edu

Mark Tenzer
Novateur Research Solutions
Ashburn, Virginia, USA
mtenzer@novateur.ai

Cyrus Shahabi
University of Southern California
Los Angeles, California, USA
shahabi@usc.edu

Khurram Shafique
Novateur Research Solutions
Ashburn, Virginia, USA
kshafique@novateur.ai

Abstract

Human mobility differs from text and from generic time series in three structural ways: visits are tuple-valued events whose meaning depends on the joint distribution over location, time, and activity; users carry persistent signatures across trajectories; and visits are not independent across users, since co-location at shared places is a primary signal. Existing pre-training recipes for mobility import objectives from language modeling, treating trajectories as sentences and visits as tokens, an analogy that fails against each of the three properties above. These properties define a broader class, multi-entity spatiotemporal event streams (MESES), spanning enterprise authentication logs, electronic health records, and other event-stream domains where entities share infrastructure, schedules, or contexts. We make the properties precise as three axioms that any pre-training framework for MESES should satisfy, and introduce TraXion, whose objectives and architecture are jointly designed to meet them. A single TraXion checkpoint per dataset leads task-specific baselines on every location-ranking and detection metric across six public mobility datasets covering anomaly detection, next-POI recommendation, next-visit prediction, and social-link prediction. The same recipe, applied unchanged to enterprise authentication logs and ICU mortality prediction, matches or exceeds prior work on both, showing that event streams from domains as different as mobility, security, and healthcare can be modeled under a single framework.[1]

1 Introduction

Across mobility, security, healthcare, and many other domains, a common kind of data appears: multiple entities generate sequences of discrete events, where each event is a tuple over space, time, and activity, and where the entities are coupled through the contexts in which their events occur. Human mobility is the canonical example. Users visit points of interest, dwell for a time, and meet at the same places as family, coworkers, and friends. The same structure appears in many event-stream domains where multiple entities share infrastructure, schedules, or contexts. For example, in enterprise authentication, users access shared hosts under role-conditioned schedules; in electronic health records, ICU stays draw from a shared catalog of clinical actions (medications, lab tests, treatments). We call this class of data *multi-entity spatiotemporal event streams* (**MESES**).

[1] Code: https://github.com/ktxlh/TraXion

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

Self-supervised pre-training is the natural way to build general-purpose representations from MESES data, but the dominant pre-training frameworks were designed for language and fail on MESES in specific, characterizable ways. Autoregressive and masked token prediction, the recipes that drive most modern sequence models for MESES, were developed under a linguistic analogy: events are like words, and entity-trajectories are like sentences [52, 79, 19, 32, 20, 22, 49]. Three properties of MESES data, however, distinguish events from language tokens at a structural level. First, a token's identity is fully captured by a vocabulary index, but a MESES event is a tuple over location, time, duration, and activity, and its meaning depends on the joint distribution over these axes: the same activity at different times, or at the same context by different entities, is not the same event. Second, a sentence is drawn from a shared linguistic process, but each entity in MESES carries a persistent signature that a token-level model has no direct mechanism to represent. Third, sentences are largely independent across speakers, but MESES events are explicitly not independent across entities, because co-occurrence at shared contexts is a primary signal rather than noise. A pre-training framework that ignores these three properties cannot model MESES data faithfully.

This paper makes three contributions. First, we define MESES as a class of data unified by tuple-valued events, persistent entity signatures, and shared-context structure, and formalize these as three axioms that any pre-training framework for MESES should satisfy (§3.2). Second, we introduce **TraXion**, a pretraining framework that couples a noise-detection objective and an entity-prototype contrastive objective on a factorized Transformer backbone with sequence, feature, and co-occurrence attention axes, satisfying the three axioms (§4). Third, we evaluate TraXion on three MESES instances spanning mobility, enterprise security, and clinical event data (§5). On six public mobility datasets covering anomaly detection, next-POI recommendation, next-visit prediction, and social-link prediction, a single TraXion checkpoint per dataset leads specialized task-specific baselines on every location-ranking and detection metric, with visit-level anomaly AP improving by 12 points over the strongest baseline. On the two non-mobility MESES instances, TraXion transfers without architectural change: scored without fine-tuning on the LANL enterprise authentication log, it attains the highest mean on four of six metrics and is statistically indistinguishable from five dedicated log-anomaly detectors, and on eICU-CRD and MIMIC-IV it leads seven of eight mortality-prediction columns. Together, these contributions identify a pre-training recipe shaped by what MESES data demands rather than what language modeling assumes, transferring across mobility, security, and healthcare.

2 Related work

Three nearby data classes might appear to subsume MESES. It is not a generic multivariate time series: events are sparse and discrete in time rather than sampled at regular intervals. It is not a generic event stream: events are coupled across entities through shared contexts. And it is not a sequence-of-tokens domain: events are tuple-valued rather than atomic. We position TraXion against each in turn.

Marked temporal point processes and irregular multivariate time series. Standard MTPP families assume neither a shared metric context set nor population-scale entity coupling. Multi-dimensional Hawkes processes [76, 37] and Dirichlet-Hawkes mixtures [14] couple streams pairwise or through clustering priors and scale super-linearly in the number of entities. Neural-MTPP variants [15, 42, 80, 66, 50] and irregular-MTS models (GRU-D [5], mTAN [51], TFT [35]) target intensity estimation or forecasting under proper likelihoods. TraXion pre-trains representations for downstream classification and ranking and is not a substitute on likelihood tasks (next-event time, hazard, return-time).

Self-supervised pre-training for sequences. Our noise detection objective ports ELECTRA's replaced-token-detection [10] from atomic tokens to tuple-valued events. Masked-modeling alternatives reconstruct corrupted positions (BERT [11], MAE [24], TST [71]). TS2Vec [70], SimMTM [12], and PatchTST [44] extend the recipe for time series. Sequential recommenders (SASRec [27], BERT4Rec [54], S^3-Rec [77], CL4SRec [61]) share an item embedding table between input and supervision; TraXion specializes the construction by making the positive class an *entity* rather than an item, the structural commitment that

follows from persistent entity signatures. The closest prior art on this point is trajectory-user-linking methods [75, 6, 7, 72], which supervise on entity identity directly but does not retain a learned per-entity prototype that re-enters the input. Earlier trajectory-specific pre-training (CTLE [36], TALE [58], TrajCL [4]) is closest in input modality. ME-POIs [52] applies the same construction — a learned prototype that serves as the supervision target — on the context side, learning prototype representations of POIs from mobility traces.

Foundation models for mobility, urban, and event streams. Trajectory foundation models target input modalities other than POI-visit streams: UniTraj [79] pre-trains on billion-scale dense GPS with self-supervised trajectory masking, TrajFlow [32] flow-matches GPS coordinates, and MobilityGPT [23] autoregressively generates road-link sequences; we benchmark UniTraj as a representative cross-modality baseline. UniST [69] pre-trains across urban tasks on grid-structured spatio-temporal data. AgentMove [17], Mobility-LLM [19], and GPT4TS [78] prompt or freeze pre-trained text models rather than pre-training from event data, which decouples the representation from the shared-context structure characteristic of MESES. CoBAD [59] models collective behavior across entities, the closest prior art to TraXion's co-occurrence axis, but couples streams through a per-event likelihood at training time rather than a pre-training mechanism, so the cross-entity signal cannot be reused across downstream tasks. In adjacent domains, BEHRT [33], Med-BERT [48], CEHR-BERT [45], and Ethos [49] pre-train on EHR event sequences, and LogBERT [20], DeepLog [13], and Tuor-AAAI17 [56] on enterprise log streams; all process each entity's stream independently.

TraXion differs from prior work along two axes. Supervision decomposes into self-supervised pre-training and a supervised fine-tune, rather than a single end-to-end stage. Cross-entity coupling enters through learned attention and a shared prototype table, rather than a pre-committed graph, a hand-built likelihood, or single-stream processing. A per-baseline comparison appears in Appendix F.

3 Multi-entity spatiotemporal event streams

We formalize the data class introduced in §1 and state three axioms that any pre-training framework over it should satisfy.

3.1 The MESES schema

A MESES dataset is a finite set of entities $\mathcal{U}$ generating events. We index entity $u \in \mathcal{U}$'s events in chronological order, so the t-th event of u is a tuple $e_{u,t} = (u, x_{u,t}, \tau_{u,t}, a_{u,t}, \delta_{u,t})$ over a shared finite context set $\mathcal{X}$ equipped with an embedding $\phi : \mathcal{X} \to \mathbb{R}^{d_x}$, where $x_{u,t} \in \mathcal{X}$ is the event's context, $\tau_{u,t}$ is its timestamp, $a_{u,t}$ is an activity category, and $\delta_{u,t}$ is an optional duration ($\delta = \emptyset$ for instantaneous events). The embedding ϕ varies by MESES instance: latitude/longitude in mobility, a UMAP of the user–host access matrix in enterprise authentication (Appendix C.5), and a UMAP of the patient–context co-occurrence matrix in electronic health records (Appendix C.6). Two entities *co-occur* when their events fall at the same context $x \in \mathcal{X}$ within a small temporal window. We call $\mathcal{X}$ the *substrate* of the MESES instance, and the distribution over the substrate for each entity u is its *signature*.

3.2 Three axioms for MESES pre-training

We state three axioms that any pre-training framework for MESES should satisfy. Each axiom is motivated by a property of MESES introduced in §3.1, and none is enforced by autoregressive next-event prediction. We use "framework" to mean the joint specification of pre-training objectives and architectural mechanisms; neither alone is sufficient to satisfy the axioms.

A1 (joint-attribute density). *Let $p(e_{u,t} \mid \mathcal{H}_{u,t})$ denote the conditional distribution over the event tuple given an encoding $\mathcal{H}_{u,t}$ of the entity's history. A pre-training framework satisfies A1 if its training signal depends on the joint distribution over the event's attributes (x, τ, a, δ) rather than on marginal or factorized distributions over them.*

An event's meaning is determined by the joint configuration of its attributes: the same activity at different times, or at the same context by different entities, is not the same event. A signal consistent only with the marginals does not penalize configurations that are plausible along each axis individually but implausible in combination.

A2 (entity-coupled representation). *Let $r(e_{u,t})$ denote the per-event representation produced by the framework's encoder, and $g(u)$ a representation associated with entity u. A pre-training framework satisfies A2 if entity identity enters $r(e_{u,t})$ as an input feature, $g(u)$ is constrained by a supervisory signal, and the input-side and supervisory-side entity representations are coupled through a shared mechanism.*

Each entity carries a persistent signature in its distribution over the substrate, and downstream tasks that depend on entity identity (anomaly detection relative to an entity's normal behavior, social-link prediction over entity pairs) require the representation to retain entity-distinguishing information. A loss that depends only on local prediction does not penalize representations that lose entity-distinguishing information across events.

A3 (shared context). *Let $\mathcal{X}$ be the substrate of a MESES instance. A pre-training framework satisfies A3 if its representations exploit two forms of recurrence over $\mathcal{X}$: that contexts are shared across entities, so different entities visiting the same $x \in \mathcal{X}$ provide mutually informative signal; and that contexts recur within an entity's history, so the entity's distribution over $\mathcal{X}$ is itself informative.*

Co-occurrence at shared contexts carries population-level information invisible to per-entity sequence modeling, and within-entity recurrence makes learning the entity's signature feasible.

Why language-derived recipes do not enforce any of the three. Both autoregressive next-event prediction and masked-token reconstruction supervise at the token level. Each event's attributes are predicted from local context only, with no constraint on their joint structure (A1), no signal targeting the entity producing them (A2), and no mechanism for one entity to inform another (A3). A model trained either way may incidentally pick up substrate structure through shared parameters, but the objective neither requires nor measures such learning.

4 Method

We instantiate the framework from §3 as TraXion, a pre-training recipe that satisfies A1–A3 through two coupled objectives and a factorized backbone.

4.1 Overview

For an entity $u \in \mathcal{U}$ with event sequence $(e_{u,1}, \ldots, e_{u,T_u})$, TraXion produces (i) a sequence encoder f_θ that maps the sequence to a matrix of contextualized event embeddings $H_u = f_\theta(e_{u,1}, \ldots, e_{u,T_u}) \in \mathbb{R}^{T_u \times d}$, and (ii) a shared entity embedding table $P \in \mathbb{R}^{|\mathcal{U}| \times d_f}$ whose row p_u we call the *prototype* of entity u. We write $\mathcal{H}_{u,t}$ for the t-th row of H_u, the contextualized representation of event $e_{u,t}$. The prototype table P plays a dual role: its rows are input to f_θ as per-event entity tokens (§4.4) and serve as supervisory targets in a contrastive loss (§4.3).

The pre-training loss combines the two objectives:

$$\mathcal{L}(\theta, \psi) = \mathcal{L}_{\text{noise}}(\theta, \psi) + \gamma\, \mathcal{L}_{\text{prototype}}(\theta), \tag{1}$$

where ψ parametrizes the noise detection head and $\gamma > 0$ balances the two terms. The noise detection objective teaches the encoder what plausible event sequences look like for entities over the substrate; the prototype objective teaches it who the sequences belong to. Neither objective uses task-specific labels, so TraXion trains on event logs from any MESES instance.

4.2 Noise-detection objective

The noise detection objective trains the encoder to detect events whose attributes are corrupted relative to the entity u's normal pattern. A stochastic operator η perturbs the *context* and *time* of some of u's events; the encoder flags which events were perturbed.

Operator and loss. Let $E_u = (e_{u,1}, \ldots, e_{u,T_u})$ be a chronologically ordered window of events for entity u. Applied to E_u, the operator η produces a corrupted window $\tilde{E}_u = \eta(E_u)$ and a per-event binary target $y_u \in \{0,1\}^{T_u}$ with $y_{u,t} = 1$ on perturbed events. With probability p_{norm} the window is left unchanged ($\tilde{E}_u = E_u$, $y_u = \mathbf{0}$); otherwise a random fraction of events is flagged, and each one independently receives a context perturbation, a time perturbation, or both (Appendix B.1.5). A linear head g_ψ on the contextualized embedding $\mathcal{H}_{u,t} = f_\theta(\tilde{E}_u)_t$ predicts per-event logits trained with masked binary cross-entropy:

$$\mathcal{L}_{\text{noise}} = \frac{1}{|\mathcal{V}|} \sum_{(u,t)\in\mathcal{V}} \text{BCE}(g_\psi(\mathcal{H}_{u,t}),\, y_{u,t})\,, \tag{2}$$

over the non-padded events $\mathcal{V}$.

Context perturbation. An event's context $x_{u,t}$ is resampled by drawing a point uniformly from the bounding box of $\phi(\mathcal{X})$, snapping to the nearest context $x' \in \mathcal{X}$ (the nearest POI in mobility, the nearest host in enterprise authentication), and replacing the event's activity $a_{u,t}$ with the activity associated with x'. The perturbed event remains locally realistic in geometry but is disconnected from the entity's typical distribution over the substrate.

Time perturbation. When event B is flagged, we resample its timestamp uniformly within the window between its neighbors A and C, leaving B's context unchanged. The new τ'_B disrupts B's timing without changing the context where the event occurred.

How this addresses A1. Detecting whether an event's context or time has been perturbed requires modeling the joint distribution over the event's attributes: a perturbed context at a typical hour, or a perturbed time at the entity's typical context, is implausible only jointly. This choice makes the objective MESES-universal: context and time are the two axes every instance shares, so the same operator runs on mobility, authentication, and clinical events.

4.3 Entity-prototype contrastive objective

The noise-detection objective trains the encoder to model events; the prototype objective trains it to remember whose events they are. The supervisory signal is the entity identifier, which is present in every event of a MESES dataset and requires no additional labeling.

Operator and loss. For each minibatch B of entities, let $\mathcal{V}^+ \subseteq \mathcal{V}$ denote the unperturbed events ($y_{u,t} = 0$). For each such event we take the ℓ_2-normalized event representation $\bar{\mathcal{H}}_{u,t}$ and the ℓ_2-normalized prototype $\bar{p}_u$, and minimize the InfoNCE loss [57]

$$\mathcal{L}_{\text{prototype}} = \frac{1}{|\mathcal{V}^+|} \sum_{(u,t)\in\mathcal{V}^+} -\log \frac{\exp\big(\bar{\mathcal{H}}_{u,t}^\top \bar{p}_u/\beta\big)}{\sum_{u'\in B} \exp\big(\bar{\mathcal{H}}_{u,t}^\top \bar{p}_{u'}/\beta\big)}, \tag{3}$$

where β is a temperature. The numerator pulls each event representation toward its own entity's prototype; the denominator pushes it away from the prototypes of the others.

Anchors in Eq. 3 are restricted to unperturbed events ($\mathcal{V}^+$) so the prototype summarizes the entity's normal behavior rather than corrupted events (Appendix B.1.2).

How this addresses A2. The prototype table P meets A2's three requirements through a single mechanism: the same vector p_u enters each event representation as an input feature and serves as the supervisory target in Eq. 3. The contrastive loss therefore shapes p_u to carry exactly the entity-distinguishing information the downstream tasks of §3.2 require.

4.4 Backbone architecture

The backbone (Figure 1) produces the contextualized event representation $\mathcal{H}_{u,t}$ used by both pre-training objectives, and it implements the architectural mechanism through which representations exploit cross-entity context. The first role is shared with most sequence encoders. The second is what makes the backbone MESES-specific. TraXion adapts factorized attention over the sequence and feature axes from prior work on multivariate sequence modeling and extends it with a third axis for co-occurrence.

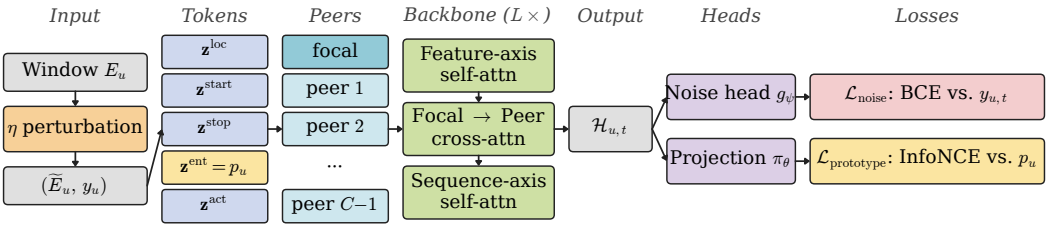

Figure 1: Pre-training architecture of TraXion. Yellow marks the entity prototype p_u, which is both an input token and the target of $\mathcal{L}_{\text{prototype}}$. Equations are in Appendix A.

Per-event feature tokens. Each event is represented as F tokens of dimension $d_f = d/F$: a multi-scale Space2Vec [40] encoding of $\phi(x_{u,t})$ over N_s scales in $[\lambda_{\min}, \lambda_{\max}]$, two Time2Vec [29] encodings of $\tau_{u,t}$ and $\tau_{u,t} + \delta_{u,t}$ (collapsed to one when $\delta = 0$), the entity prototype p_u looked up from P (factored as $\mathcal{U} \rightarrow h$-latent $\rightarrow d_f$ for large instances, $h \ll d_f$; Appendix B.2), and a linear projection of the activity $a_{u,t}$.

Factorized attention with a co-occurrence axis. Attention is applied independently along the sequence, feature, and co-occurrence axes of a rank-5 tensor (B, T, F, C, d_f), where C is the size of the co-occurrence window (the focal event plus up to $C-1$ peers); factorizing the first two axes is standard for multivariate sequences [38, 74] and reduces the $O((TFC)^2)$ cost of full attention to a sum of standard sub-layers. The third axis is what makes the backbone MESES-specific: for each focal event we retrieve up to $C-1$ temporally proximate events from other entities at the same context x (Appendix B.1.4), and the co-occurrence sub-layer attends from the focal event to its peers but not vice versa, since each peer is itself the focal slot in its own window (Appendix A.3).

How this addresses A3. A3 requires exploiting two forms of substrate recurrence: contexts shared across entities, and contexts recurring within an entity's history. The co-occurrence axis addresses the first by making peer events at shared contexts directly available to the focal event. The prototype table P, trained over the full corpus by Eq. 3, addresses the second by encoding each entity's distribution over substrate contexts in a fixed vector. The prototype captures the entity's typical pattern across $\mathcal{X}$, and the co-occurrence axis captures the population-level pattern at a specific x.

5 Experiments

Datasets. We evaluate on three MESES instances. For mobility, we use six public human-mobility datasets: NUMOSIM-LA [53], Urban Anomalies-Berlin/-Atlanta (UA) [1], Foursquare-Tokyo [67], Gowalla-Stockholm and Gowalla-Austin [8]. For enterprise authentication, we use a subsampled cohort of the LANL unified host-and-network authentication log [30] (190,944 events from 1,598 users; Appendix C.5). For electronic health records, we use PhysioNet eICU-CRD [47] (197,307 ICU stays from 138,176 patients) and MIMIC-IV [26] (94,407 stays from 65,360 patients), both with in-hospital mortality as the label. Per-dataset statistics are in Appendix C.1. Across these instances we cover four downstream tasks: visit/agent-level anomaly detection (§5.1, §5.4), next-POI and next-visit prediction (§5.2), social-link inference (§5.3), and ICU mortality prediction (§5.5).

Protocol. For every dataset we pre-train one shared backbone (§4.4) with the objectives of §4.2–4.3, then fine-tune once per downstream task with a per-task head described in the corresponding subsection. Whether to bypass the pre-trained co-occurrence sub-layer at fine-tune is decided per seed on the validation criterion: both routes are trained for every seed, and the selected one is reported. Rationale in Appendix G.

Metrics. For binary classification (anomaly detection, ICU mortality, social-link) we report AP (average precision) and Max F1, with AUROC where applicable. We additionally report Sens@Sp.9 (sensitivity at 90% specificity) for ICU mortality. For ranking (next-POI, next-visit, social-link) we report H@k (hit-rate at k) and MRR (mean reciprocal rank). For next-visit time prediction we report T±60: for models that emit a density, the predicted

probability mass within 60 min of the ground-truth check-in time; for point-prediction baselines, the fraction of predictions falling within that window. Both are conditioned on the true next location, following TrajGPT [25].

Implementation and reporting. Splits, cohorts, hyperparameters, and the iterative HP search applied identically to TraXion and retuned baselines are in Appendices B–D, B.3. Runs use one NVIDIA RTX 6000 Ada (48 GB); every end-to-end pre-train plus fine-tune completes within 5 days, with measured per-stage times in Appendix E.2. Baselines use released code except UniTraj† [79] (adapted; Appendix D.1), COAST [62] (re-implemented; Appendix D.3), and some LANL log baselines (re-implemented; Appendix D.5). Tables 1–5 report mean ± sample standard deviation; higher is better. TraXion is at 5 seeds in every cell, as is every baseline that leads a column; the remaining baselines are at 3–5 seeds, and EBM is deterministic by construction (single run). **Bold** marks the best mean in each column and underline the second best. $*$ marks $p \leq .05$ wins *or* losses for TraXion against the strongest per-column baseline (one-sided Wilcoxon rank-sum on the per-seed scores). Appendix E.1 reports percentile bootstrap intervals for the anomaly-detection results of Tables 1 and 4.

5.1 Human mobility anomaly detection

Anomaly detection on three simulated benchmarks (ground-truth per-visit/per-agent labels for inserted-visit attacks and behavioral shifts) directly tests the noise-detection objective (§4.2); UA-Berlin and UA-Atlanta (`centralized` mode) provide only agent-level labels. We fine-tune TraXion with dataset-specific self-supervised objectives matched to each benchmark's anomaly type (Appendix B.2).

Table 1: Human mobility anomaly detection results. UA := Urban Anomalies.

	NUMOSIM-LA (visit)		NUMOSIM-LA (agent)		UA-Berlin		UA-Atlanta	
Method	AP	Max F1	AP	Max F1	AP	Max F1	AP	Max F1
DeepBayesic [16]	$.0002_{\pm.000}$	$.0009_{\pm.001}$	$.0020_{\pm.000}$	$.0045_{\pm.001}$	$.2177_{\pm.036}$	$\underline{.3014}_{\pm.034}$	$.1548_{\pm.008}$	$.2396_{\pm.005}$
CoBAD [59]	$.0067_{\pm.002}$	$.0330_{\pm.010}$	$.0143_{\pm.004}$	$.0628_{\pm.010}$	$.1408_{\pm.002}$	$.2236_{\pm.008}$	$.1625_{\pm.003}$	$.2436_{\pm.008}$
UniTraj† [79]	$.0010_{\pm.000}$	$.0084_{\pm.005}$	$.0052_{\pm.001}$	$.0196_{\pm.001}$	$\underline{.2728}_{\pm.018}$	$.2805_{\pm.012}$	$\underline{.3237}_{\pm.019}$	$\underline{.3094}_{\pm.023}$
ICAD [3]	$\underline{.0775}_{\pm.057}$	$\underline{.1382}_{\pm.062}$	$\underline{.1455}_{\pm.085}$	$\underline{.2282}_{\pm.088}$	$.1353_{\pm.011}$	$.2361_{\pm.022}$	$.1644_{\pm.021}$	$.2448_{\pm.009}$
USTAD [60]	$.0079_{\pm.001}$	$.0428_{\pm.003}$	$.0208_{\pm.002}$	$.0769_{\pm.007}$	$.1531_{\pm.004}$	$.2364_{\pm.005}$	$.2328_{\pm.008}$	$.2988_{\pm.006}$
TraXion	$\mathbf{.1950}_{\pm.010}$*	$\mathbf{.2901}_{\pm.009}$*	$\mathbf{.5764}_{\pm.015}$*	$\mathbf{.6485}_{\pm.011}$*	$\mathbf{.4132}_{\pm.014}$*	$\mathbf{.4261}_{\pm.024}$*	$\mathbf{.4141}_{\pm.023}$*	$\mathbf{.4214}_{\pm.022}$*

Discussion. TraXion tops all 8 columns, every one of them significantly, with AP gains of +.117/+.431 over the strongest baseline (ICAD) at NUMOSIM-LA visit-/agent-level and +.140/+.090 over UniTraj† on UA-Berlin/-Atlanta. Which co-occurrence route validation selects tracks task structure: it keeps the pathway on NUMOSIM, where inserted-visit anomalies are focal-vs-crowd mismatches at the place, and bypasses it on UA-Atlanta (4 of 5 seeds), where behavioral shifts at preserved place and time leave co-occurrence uninformative; UA-Berlin splits 3 active to 2 bypass (Appendix G).

5.2 Next POI and next visit prediction

We predict the next visit on three LBSN city subsets, decomposed into location ranking (H@10/H@20) and time prediction (T±60); Table 2 stacks next-POI (location only) above next-visit baselines (joint location and time).

Discussion. On next-POI, TraXion tops H@10/H@20 on all three cities and every one of those six margins is significant, leading the strongest baseline by +.050/+.088/+.072 H@10 (MobTCast on Tokyo, LoTNext on Stockholm/Austin). On next-visit, TraXion tops H@10/H@20 on all three cities as well and every one of those six margins is significant, beating the strongest next-visit baseline TAPT by +.109/+.063/+.097 H@10. Mobility-LLM and A-NHP lead T±60 on all three cities, at 2.0–3.9× and 1.2–1.5× our value, respectively, but at 3.9–6.0× and 5.3–21× lower H@10, placing each on a distinct location–time Pareto point.

Table 2: Next-POI and next-visit prediction results.

Method	Foursquare-Tokyo			Gowalla-Stockholm			Gowalla-Austin		
	H@10	H@20	T±60	H@10	H@20	T±60	H@10	H@20	T±60
Next POI recommendation									
UniTraj† [79]	.0020$_{\pm.000}$	.0038$_{\pm.001}$		.0182$_{\pm.005}$	.0233$_{\pm.005}$		.0089$_{\pm.001}$	.0169$_{\pm.002}$	
GETNext [68]	.1607$_{\pm.010}$	.2118$_{\pm.013}$		.0584$_{\pm.002}$	.0780$_{\pm.003}$		.1463$_{\pm.012}$	.1873$_{\pm.013}$	
COAST [62]	.3478$_{\pm.001}$	.4109$_{\pm.002}$	N/A	.1388$_{\pm.003}$	.1627$_{\pm.003}$	N/A	.1296$_{\pm.001}$	.1609$_{\pm.000}$	N/A
MobTCast [65]	.3827$_{\pm.002}$	.4510$_{\pm.002}$		.1943$_{\pm.005}$	.2245$_{\pm.006}$		.1459$_{\pm.004}$	.1834$_{\pm.005}$	
LoTNext [63]	.3318$_{\pm.002}$	.3958$_{\pm.002}$		.2362$_{\pm.002}$	.2786$_{\pm.001}$		.2034$_{\pm.003}$	.2514$_{\pm.002}$	
TraXion	**.4330**$_{\pm.002}$*	**.5248**$_{\pm.002}$*		**.3237**$_{\pm.003}$*	**.3946**$_{\pm.003}$*		**.2755**$_{\pm.005}$*	**.3426**$_{\pm.006}$*	
Next visit prediction									
THP [80]	.0831$_{\pm.001}$	.1036$_{\pm.000}$	.0620$_{\pm.003}$	.0460$_{\pm.001}$	.0610$_{\pm.001}$	.0909$_{\pm.002}$	.0466$_{\pm.005}$	.0742$_{\pm.008}$	.1584$_{\pm.004}$
A-NHP [66]	.0182$_{\pm.009}$	.0219$_{\pm.011}$	.2708$_{\pm.000}$	.0486$_{\pm.003}$	.0641$_{\pm.004}$	.2028$_{\pm.052}$	.0528$_{\pm.008}$	.0752$_{\pm.008}$	.2496$_{\pm.002}$
TrajGPT [25]	.1106$_{\pm.057}$	.1420$_{\pm.070}$	.1566$_{\pm.031}$	.0475$_{\pm.000}$	.0636$_{\pm.002}$	.1221$_{\pm.029}$	.0562$_{\pm.032}$	.0793$_{\pm.041}$	.1523$_{\pm.014}$
TAPT [64]	.2745$_{\pm.006}$	.3690$_{\pm.008}$	.1111$_{\pm.013}$	.2575$_{\pm.004}$	.3303$_{\pm.003}$	.1044$_{\pm.007}$	.1818$_{\pm.006}$	.2474$_{\pm.006}$	.1099$_{\pm.004}$
Mobility-LLM [19]	.0840$_{\pm.001}$	.1063$_{\pm.001}$	**.4307**$_{\pm.027}$	.0536$_{\pm.001}$	.0692$_{\pm.001}$	**.6173**$_{\pm.074}$	.0717$_{\pm.006}$	.1049$_{\pm.004}$	**.5614**$_{\pm.073}$
TraXion	**.3838**$_{\pm.003}$*	**.4835**$_{\pm.003}$*	.2181$_{\pm.004}$*	**.3203**$_{\pm.007}$*	**.3916**$_{\pm.007}$*	.1601$_{\pm.006}$*	**.2787**$_{\pm.001}$*	**.3465**$_{\pm.003}$*	.1650$_{\pm.011}$*

5.3 Social link inference

Predicting platform-graph links from visit histories alone directly tests A3: co-located visits at overlapping times are the primary observable, and pre-training sees no graph supervision.

Table 3: Social link inference results.

Method	Foursquare-Tokyo			Gowalla-Stockholm			Gowalla-Austin		
	AP	H@10	MRR	AP	H@10	MRR	AP	H@10	MRR
UniTraj† [79]	.5126$_{\pm.008}$	.1570$_{\pm.007}$	.0736$_{\pm.002}$	.4337$_{\pm.011}$	.1273$_{\pm.010}$	.0637$_{\pm.003}$	.4965$_{\pm.032}$	.1549$_{\pm.008}$	.0766$_{\pm.004}$
EBM [46]	.7317	.3233	.1760	.7445	.4977	.2994	.7845	.3716	.1801
USRC [9]	.8688$_{\pm.001}$	.5347$_{\pm.003}$	.2686$_{\pm.002}$	.8256$_{\pm.008}$	.5282$_{\pm.005}$	.2943$_{\pm.005}$	.8983$_{\pm.009}$	.6094$_{\pm.007}$	.3352$_{\pm.008}$
H^3GNN [34]	.9137$_{\pm.002}$	.7080$_{\pm.010}$	.4270$_{\pm.011}$	.8625$_{\pm.003}$	.5968$_{\pm.008}$	.3658$_{\pm.011}$	.8863$_{\pm.003}$	.6215$_{\pm.017}$	.3377$_{\pm.019}$
LBSN2Vec [67]	.9042$_{\pm.001}$	.7127$_{\pm.002}$	.4839$_{\pm.002}$	.8208$_{\pm.001}$	.6628$_{\pm.002}$	.4874$_{\pm.002}$	.8632$_{\pm.002}$	.6220$_{\pm.003}$	.3970$_{\pm.002}$
TraXion	**.9344**$_{\pm.001}$*	**.7801**$_{\pm.002}$*	**.5347**$_{\pm.011}$*	**.9276**$_{\pm.001}$*	**.7426**$_{\pm.003}$*	**.5870**$_{\pm.005}$*	**.9373**$_{\pm.001}$*	**.7423**$_{\pm.007}$*	**.4997**$_{\pm.024}$*

Discussion. TraXion attains the highest mean in all 9 columns, and every one of them is significant. AP gains are +.021/+.065/+.039 across Tokyo/Stockholm/Austin, and on Gowalla-Austin the margins over the strongest baseline are +.120 H@10 and +.103 MRR. Clearing every graph-aware baseline (LBSN2Vec, H^3GNN, USRC) without any graph supervision at pre-training is direct evidence for A3: co-located visits suffice to recover platform-level social ties.

5.4 Enterprise authentication logs

We run the same pre-training code, with no architectural changes, on the LANL unified-host-and-network authentication log (749 red-team lateral-movement events as ground truth), a non-mobility MESES instance with no native geometry. We score TraXion's pre-trained backbone directly, with no fine-tuning on LANL (scoring protocol in Appendix D.6). We report event-level and user-level metrics (per-`src_user` max-pool, Tuor convention [56]). Appendix E.4 reports results on two perturbations of the cohort, a different benign-user sample and twice the per-user event cap.

Table 4: Anomaly detection on LANL enterprise authentication logs.

Method	Event-level			User-level		
	AP	AUROC	Max F1	AP	AUROC	Max F1
LogGPT [22]	.0055$_{\pm.001}$	.7881$_{\pm.008}$	.0217$_{\pm.006}$	.2172$_{\pm.036}$	.8358$_{\pm.011}$	.3362$_{\pm.064}$
UniTraj† [79]	.0075$_{\pm.001}$	.7998$_{\pm.024}$	.0437$_{\pm.001}$	.1961$_{\pm.009}$	.7936$_{\pm.010}$	.2648$_{\pm.014}$
NuLog [43]	.0059$_{\pm.000}$	.8558$_{\pm.003}$	.0237$_{\pm.002}$	.2337$_{\pm.053}$	.8359$_{\pm.010}$	.3585$_{\pm.059}$
LogBERT [20]	.0079$_{\pm.001}$	**.8877**$_{\pm.006}$	.0213$_{\pm.004}$	.2311$_{\pm.094}$	.8502$_{\pm.008}$	.3133$_{\pm.052}$
Tuor-AAAI17 [56]	.0112$_{\pm.003}$	.8680$_{\pm.008}$	.0604$_{\pm.019}$	.1717$_{\pm.033}$	.8875$_{\pm.003}$	.2875$_{\pm.023}$
DeepLog [13]	.0066$_{\pm.002}$	.8669$_{\pm.017}$	.0238$_{\pm.011}$	.2351$_{\pm.050}$	**.8940**$_{\pm.011}$	.3594$_{\pm.058}$
TraXion (pre-train only)	**.0173**$_{\pm.011}$	.8594$_{\pm.052}$	**.0683**$_{\pm.030}$	**.2392**$_{\pm.039}$	.8611$_{\pm.033}$	**.4151**$_{\pm.042}$

Discussion. Every baseline here is a log-anomaly detector trained on LANL with its own self-supervised objective; TraXion is pre-trained on the same data with its generic objective and scored without fine-tuning, and it lands in the same range as all of them across all six metrics — attaining the highest mean on four (event-level AP and Max F1, user-level AP and Max F1), with LogBERT leading event-level AUROC and DeepLog user-level AUROC.

None of the six differences is significant, so the fair reading is parity rather than a ranking. The MESES recipe therefore carries off mobility without task-specific training: no LANL fine-tuning and no architectural changes.

5.5 ICU mortality prediction

In-hospital mortality on two EHR corpora — eICU-CRD (197,307 ICU stays, 17,673 deaths, 9.0%) and MIMIC-IV (94,407 stays, 11,333 deaths, 12.0%) — is a second non-mobility MESES instance. Each stay maps into the (entity, context, time, category) schema of §4.2 via 507 (eICU) and 520 (MIMIC) clinical-action concepts, with the patient as the entity and the stay as the prediction unit; splits are patient-level so all stays of a patient land in one fold (Appendix C.3, fine-tune head in Appendix B.2). Every sequence baseline with a static-feature path receives the same side features as TraXion.

Table 5: ICU mortality prediction on eICU-CRD and MIMIC-IV.

	eICU-CRD				MIMIC-IV			
Method	AUROC	AP	Max F1	Sens@Sp.9	AUROC	AP	Max F1	Sens@Sp.9
UniTraj† [79]	.7574$_{\pm.013}$	.2993$_{\pm.024}$	.3500$_{\pm.016}$	.4197$_{\pm.023}$	.7681$_{\pm.015}$	.3753$_{\pm.029}$	.4037$_{\pm.018}$	.4271$_{\pm.026}$
Raindrop [73]	.8446$_{\pm.004}$	.4347$_{\pm.012}$	.4434$_{\pm.007}$	.5574$_{\pm.010}$	.8650$_{\pm.006}$	.5820$_{\pm.019}$	.5452$_{\pm.013}$	.6205$_{\pm.015}$
SimMTM [12]	.8790$_{\pm.002}$	.5565$_{\pm.004}$	.5162$_{\pm.002}$	.6406$_{\pm.005}$	.8748$_{\pm.002}$	.5725$_{\pm.003}$	.5362$_{\pm.004}$	.6219$_{\pm.006}$
WaveGNN [21]	.8086$_{\pm.002}$	.3905$_{\pm.004}$	.3985$_{\pm.003}$	.4815$_{\pm.005}$	.9032$_{\pm.002}$	.6477$_{\pm.003}$	.5875$_{\pm.005}$	.6886$_{\pm.013}$
STraTS [55]	.8938$_{\pm.001}$	**.5965**$_{\pm.003}$	.5449$_{\pm.004}$	.6779$_{\pm.007}$	.8866$_{\pm.001}$	.6052$_{\pm.005}$	.5546$_{\pm.004}$	.6480$_{\pm.004}$
TraXion	**.9006**$_{\pm.001}$*	.5705$_{\pm.004}$*	**.5551**$_{\pm.004}$*	**.7074**$_{\pm.005}$*	**.9149**$_{\pm.002}$*	**.6812**$_{\pm.007}$*	**.6221**$_{\pm.004}$*	**.7432**$_{\pm.006}$*

Discussion. TraXion leads 7 of the 8 columns, each significantly: all four MIMIC-IV metrics over the strongest EHR baseline there (WaveGNN, by +.012 AUROC, +.034 AP, +.035 Max F1, +.055 Sens@Sp.9) and eICU AUROC, Max F1 and Sens@Sp.9 over STraTS. The single loss is eICU AP, where STraTS's .5965 beats our .5705 significantly. That the prototype-conditioned denoising objective clears both corpora's strongest baseline on 7 of 8 metrics, on a substrate with no geometry and no mobility, is the transfer claim of §3.2 at scale.

5.6 Ablation study

We ablate seven design choices (C1–C7) spanning the pre-training objective, prototype routing, and backbone, with each variant pre-trained and fine-tuned on UA-Berlin (mobility AD), Gowalla-Austin (next-visit, social link), and LANL (auth-log AD). All rows of a column family use the same fixed co-occurrence route. The protocol differs from that of the per-task tables (Appendix E.3), so the Full TraXion row of Table 6 does not match those tables. Cells with ± report the mean and standard deviation over five seeds; cells without ± are single runs.

Table 6: Ablation results.

	Mobility AD		Next-Visit		Social Link		Auth-Log AD	
Variant	AP	MaxF1	H@10	T±60	AP	H@10	AP	MaxF1
Full TraXion	.4290$_{\pm.018}$	.4386$_{\pm.021}$	.2800$_{\pm.003}$	.1641$_{\pm.018}$	.9009$_{\pm.006}$	.6495$_{\pm.006}$	.2392$_{\pm.039}$	.4151$_{\pm.042}$
C1: w/o $\mathcal{L}_{\text{prototype}}$	.1249	.2162	.1471	.1057	.8893$_{\pm.010}$	.6489$_{\pm.016}$	.1440	.2812
C2: w/o $\mathcal{L}_{\text{noise}}$	.3855	.3729	.2588	.1510	.8844$_{\pm.011}$	.5895$_{\pm.052}$	.0550	.1165
C3: w/o prototype-as-input	.1671	.2235	.1787	.1190	.8826$_{\pm.012}$	.6368$_{\pm.020}$	.1621	.3158
C4: next-token pretrain	.3629	.4046	.2810$_{\pm.002}$	.1639$_{\pm.011}$	.9091$_{\pm.003}$	.6765$_{\pm.010}$	.1584	.2222
C5: masked-token pretrain	.4066$_{\pm.022}$	.4040$_{\pm.018}$	.2893$_{\pm.004}$	.1678$_{\pm.007}$	.9090$_{\pm.003}$	.6739$_{\pm.005}$	.1499	.2785
C6: no pre-training	.1320	.2161	.2587	.1522	.9170$_{\pm.003}$	.6907$_{\pm.009}$	.2778$_{\pm.097}$	.3823$_{\pm.081}$
C7: flat transformer encoder	.1241	.2192	.0410	.1318	.8965$_{\pm.006}$	.6425$_{\pm.016}$	.0186	.0479

Discussion. C1 (no $\mathcal{L}_{\text{prototype}}$) drops mobility-AD AP by 71% and next-visit H@10 by 47%, and C3 (prototype kept as a contrastive positive but removed from input tokens) is the weakest arm on social-link AP. C1 and C3 score below Full in every column. The arms that *substitute* one pre-training objective for another — C4 (autoregressive next-token) and C5 (masked-token, UniTraj-style [79]) — do not behave uniformly across tasks: Full leads C5 by +.035 Max F1 on mobility anomaly detection, whereas C5 leads Full by +.009 next-visit H@10, and C4 and C5 lead Full on social link by +.008 AP and by +.027 and +.024 H@10. Each of these differences is significant ($p \leq .03$, one-sided Wilcoxon rank-sum over five seeds). C6 (no pre-training) is the strongest social-link arm. On next-visit T±60,

C4 and C5 are within .004 of Full, and C1 and C3 are .058 and .045 below it. C7 (flat transformer) is the weakest variant on next-visit H@10 (.0410 against Full's .2800) and on both auth-log AD metrics (.0186/.0479 against .2392/.4151); on social link it trails Full by under .005 AP and .007 H@10, and neither difference is significant ($p \geq .15$). Across tasks, the entity prototype (C1, C3) and the factorized attention (C7) account for the largest drops on anomaly detection and next-visit prediction, the denoising objective (C2, C5) helps mainly on anomaly detection, and on social link the strongest arm is the one without pre-training (C6).

6 Conclusion

We introduced *multi-entity spatiotemporal event streams* (MESES), a class of data found in domains as varied as mobility, security, and clinical events. We formalized three axioms (A1–A3) that any pre-training framework for MESES should satisfy, and showed that the dominant next-token pre-training recipes — autoregressive and masked alike — enforce none of the three.

We present TraXion as a framework that satisfies all three. TraXion's noise-detection and entity-prototype objectives on a co-occurrence-augmented backbone are jointly designed to satisfy A1–A3, and a single TraXion checkpoint per dataset matches or beats specialized baselines across six mobility cohorts and four tasks. Ablations are consistent with the axiom-aligned mechanism where the axioms bite hardest: removing the prototype objective collapses anomaly detection (-71% AP) and next-visit ranking (-47% H@10), and removing its input-side role gives the largest social-link AP drop. The same backbone transfers without architectural change to two non-mobility instances: scored without fine-tuning on the LANL authentication log, it is statistically indistinguishable from five dedicated log-anomaly detectors, and on eICU-CRD and MIMIC-IV it leads seven of eight mortality-prediction columns. That is what makes TraXion a framework for MESES rather than a domain-specific tool.

The MESES framing extends in two ways. Many other event streams (e-commerce sessions, telecom usage, IoT readings, public-health surveillance) fit the same $(\text{entity}, \text{context}, \text{time}, \text{category})$ schema, and pre-training a unified TraXion backbone over a multi-instance MESES corpus is the natural next step. More broadly, TraXion illustrates a principle that should generalize beyond MESES: pre-training objectives are most effective when designed to match structural properties of the data, rather than imported from other domains. Identifying analogous axioms for other data classes (multi-modal sensor streams, knowledge graphs, scientific time series) is a natural extension of this principled methodology.

Limitations. TraXion is discriminative, not a TPP substitute (it trails on T±60); the co-occurrence sub-layer is dispatched per seed on a validation criterion rather than learned; and we pre-train one checkpoint per dataset rather than a unified MESES backbone (Appendix G).

Acknowledgments and Disclosure of Funding

This research has been funded in part by the NSF award DMS-2428039, and the Intelligence Advanced Research Projects Activity (IARPA) via the Department of Interior/Interior Business Center (DOI/IBC) contract number 140D0423C0033. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes, notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA or the U.S. Government. The authors gratefully acknowledge the compute and data resources provided by Google Cloud Platform through the Google Cloud Research Credits program, as well as the unrestricted cash award from Google Research.

References

[1] Hossein Amiri, Ruochen Kong, and Andreas Züfle. Urban Anomalies: A simulated human mobility dataset with injected anomalies. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection*, pages 1–11, 2024.

[2] Anthropic. Claude Code. `https://www.anthropic.com/claude-code`, 2025.

[3] Bita Azarijoo, Maria Despoina Siampou, John Krumm, and Cyrus Shahabi. ICAD: A self-supervised autoregressive approach for multi-context anomaly detection in human mobility data. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, pages 595–606, 2025.

[4] Yanchuan Chang, Jianzhong Qi, Yuxuan Liang, and Egemen Tanin. Contrastive trajectory similarity learning with dual-feature attention. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2933–2945. IEEE, 2023.

[5] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018.

[6] Wei Chen, Shuzhe Li, Chao Huang, Yanwei Yu, Yongguo Jiang, and Junyu Dong. Mutual distillation learning network for trajectory-user linking. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization., 2022.

[7] Wei Chen, Chao Huang, Yanwei Yu, Yongguo Jiang, and Junyu Dong. Trajectory-user linking via hierarchical spatio-temporal attention networks. *ACM Transactions on Knowledge Discovery from Data*, 18(4):1–22, 2024.

[8] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1082–1090, 2011.

[9] Chen Chu, Cyrus Shahabi, Emmanuel Tung, and Khurram Shafique. One model, many cities: A transferable social relationship inference framework for human mobility data. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, pages 66–76, 2025.

[10] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[12] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. SimMTM: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36:29996–30025, 2023.

[13] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1285–1298, 2017.

[14] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-Hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 219–228, 2015.

[15] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

[16] Minxuan Duan, Yinlong Qian, Lingyi Zhao, Zihao Zhou, Zeeshan Rasheed, Rose Yu, and Khurram Shafique. Back to Bayesics: Uncovering human mobility distributions and anomalies with an integrated statistical and neural framework. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection*, pages 56–67, 2024.

[17] Jie Feng, Yuwei Du, Jie Zhao, and Yong Li. AgentMove: A large language model based agentic framework for zero-shot next location prediction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1322–1338, 2025.

[18] Jun Gao, Wei Wang, Changlong Yu, Huan Zhao, Wilfred Ng, and Ruifeng Xu. Improving event representation via simultaneous weakly supervised contrastive learning and clustering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

[19] Letian Gong, Yan Lin, Xinyue Zhang, Yiwen Lu, Xuedi Han, Yichen Liu, Shengnan Guo, Youfang Lin, and Huaiyu Wan. Mobility-LLM: Learning visiting intentions and travel preference from human mobility data with large language models. *Advances in Neural Information Processing Systems*, 37:36185–36217, 2024.

[20] Haixuan Guo, Shuhan Yuan, and Xintao Wu. LogBERT: Log anomaly detection via BERT. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[21] Arash Hajisafi, Maria Despoina Siampou, Bita Azarijoo, Zhen Xiong, and Cyrus Shahabi. WaveGNN: Integrating graph neural networks and transformers for decay-aware classification of irregular clinical time-series. In *2025 IEEE International Conference on Big Data (BigData)*, pages 1934–1943, 2025. doi: 10.1109/BigData66926.2025.11401906.

[22] Xiao Han, Shuhan Yuan, and Mohamed Trabelsi. LogGPT: Log anomaly detection via GPT. In *2023 IEEE International Conference on Big Data (BigData)*, pages 1117–1122, 2023.

[23] Ammar Haydari, Dongjie Chen, Zhengfeng Lai, Michael Zhang, and Chen-Nee Chuah. MobilityGPT: Enhanced human mobility modeling with a GPT model. *IEEE Transactions on Intelligent Transportation Systems*, 27(1):1681–1694, 2025.

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[25] Shang-Ling Hsu, Emmanuel Tung, John Krumm, Cyrus Shahabi, and Khurram Shafique. TrajGPT: Controlled synthetic trajectory generation using a multitask transformer-based spatiotemporal model. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, pages 362–371, 2024.

[26] Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10(1), 2023. doi: 10.1038/s41597-022-01899-x.

[27] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206, 2018. doi: 10.1109/ICDM.2018.00035.

[28] Andrej Karpathy. autoresearch. `https://github.com/karpathy/autoresearch`, 2025.

[29] 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.

[30] Alexander D Kent. Comprehensive, multi-source cyber-security events data set. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2015.

[31] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021.

[32] Peiran Li, Jiawei Wang, Haoran Zhang, Xiaodan Shi, Noboru Koshizuka, Chihiro Shimizu, and Renhe Jiang. TrajFlow: Nation-wide pseudo GPS trajectory generation with flow matching models. In *The Fourteenth International Conference on Learning Representations*, 2026.

[33] Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. BEHRT: transformer for electronic health records. *Scientific Reports*, 10(1):7155, 2020.

[34] Yongkang Li, Zipei Fan, and Xuan Song. Heterogeneous hyperbolic hypergraph neural network for friend recommendation in location-based social networks. *ACM Transactions on Knowledge Discovery from Data*, 19(3):1–29, 2025.

[35] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764, 2021.

[36] Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4241–4248, 2021.

[37] Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *International Conference on Machine Learning*, pages 1413–1421. PMLR, 2014.

[38] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[40] Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. Multi-scale representation learning for spatial feature distributions using grid cells. In *International Conference on Learning Representations (ICLR)*, 2020.

[41] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29), 2018.

[42] Hongyuan Mei and Jason M Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. *Advances in Neural Information Processing Systems*, 30, 2017.

[43] Sasho Nedelkoski, Jasmin Bogatinovski, Alexander Acker, Jorge Cardoso, and Odej Kao. Self-supervised log parsing. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, 2020.

[44] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[45] Chao Pang, Xinzhuo Jiang, Krishna S Kalluri, Matthew Spotnitz, RuiJun Chen, Adler Perotte, and Karthik Natarajan. CEHR-BERT: Incorporating temporal information from structured ehr data to improve prediction tasks. In *Machine Learning for Health*, pages 239–260. PMLR, 2021.

[46] Huy Pham, Cyrus Shahabi, and Yan Liu. EBM: an entropy-based model to infer social strength from spatiotemporal data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 265–276, 2013.

[47] Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eICU collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5(180178), 2018. doi: 10.1038/sdata.2018.178.

[48] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ Digital Medicine*, 4(1):86, 2021.

[49] Pawel Renc, Yugang Jia, Anthony E. Samir, Jaroslaw Was, Quanzheng Li, David W. Bates, and Arkadiusz Sitek. Zero shot health trajectory prediction using transformer. *NPJ Digital Medicine*, 7(1):256, 2024.

[50] Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. Neural temporal point processes: A review. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 4585–4593. International Joint Conferences on Artificial Intelligence Organization, 2021.

[51] Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

[52] Maria Despoina Siampou, Shushman Choudhury, Shang-Ling Hsu, Neha Arora, and Cyrus Shahabi. Mobility-embedded POIs: Learning what a place is and how it is used from human movement. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, pages 113054–113067. PMLR, 2026.

[53] Chris Stanford, Suman Adari, Xishun Liao, Yueshuai He, Qinhua Jiang, Chenchen Kuai, Jiaqi Ma, Emmanuel Tung, Yinlong Qian, Lingyi Zhao, et al. NUMOSIM: A synthetic mobility dataset with anomaly detection benchmarks. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection*, pages 68–78, 2024.

[54] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450, 2019.

[55] Sindhu Tipirneni and Chandan K Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17, 2022.

[56] Aaron Tuor, Samuel Kaplan, Brian Hutchinson, Nicole Nichols, and Sean Robinson. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. In *AAAI Workshops*, pages 224–231, 2017.

[57] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[58] Huaiyu Wan, Yan Lin, Shengnan Guo, and Youfang Lin. Pre-training time-aware location embeddings from spatial-temporal trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5510–5523, 2021.

[59] Haomin Wen, Shurui Cao, and Leman Akoglu. CoBAD: Modeling collective behaviors for human mobility anomaly detection. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, pages 197–209, 2025.

[60] Haomin Wen, Shurui Cao, Zeeshan Rasheed, Khurram Hassan Shafique, and Leman Akoglu. Uncertainty-aware spatio-temporal human mobility modeling and anomaly detection. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, pages 328–331, 2025.

[61] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1259–1273. IEEE, 2022.

[62] Bada Xin, Xin Wan, Zhuojun Jiang, Faqiang Liu, Su Chen, Rong Yang, and Qingyun Liu. COAST: Contrastive learning with augmented spatio-temporal encoding for next POI recommendation. In *ICASSP 2025–2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[63] Xiaohang Xu, Renhe Jiang, Chuang Yang, Zipei Fan, and Kaoru Sezaki. Taming the long tail in human mobility prediction. *Advances in Neural Information Processing Systems*, 37: 54748–54771, 2024.

[64] Yuanbo Xu, Hongxu Shen, Yiheng Jiang, and En Wang. Where and when: predict next POI and its explicit timestamp in sequential recommendation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 3507–3515, 2025.

[65] Hao Xue, Flora Salim, Yongli Ren, and Nuria Oliver. MobTCast: Leveraging auxiliary trajectory forecasting for human mobility prediction. *Advances in Neural Information Processing Systems*, 34:30380–30391, 2021.

[66] Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *International Conference on Learning Representations*, 2022.

[67] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. Revisiting user mobility and social relationships in LBSNs: a hypergraph embedding approach. In *The World Wide Web Conference*, pages 2147–2157, 2019.

[68] Song Yang, Jiamou Liu, and Kaiqi Zhao. GETNext: trajectory flow map enhanced transformer for next POI recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1144–1153, 2022.

[69] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. UniST: A prompt-empowered universal model for urban spatio-temporal prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4095–4106, 2024.

[70] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. TS2Vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987, 2022.

[71] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124, 2021.

[72] Hao Zhang, Wei Chen, Xingyu Zhao, Jianpeng Qi, Guiyuan Jiang, and Yanwei Yu. Scalable trajectory-user linking with dual-stream representation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13224–13232, 2025.

[73] Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*, 2022.

[74] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

[75] Fan Zhou, Qiang Gao, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. Trajectory-user linking via variational autoencoder. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3212–3218, 2018.

[76] Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional Hawkes processes. In *International Conference on Machine Learning*, pages 1301–1309. PMLR, 2013.

[77] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1893–1902, 2020.

[78] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained LM. *Advances in Neural Information Processing Systems*, 36:43322–43355, 2023.

[79] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Xun Zhou, Liang Han, Xuetao Wei, and Yuxuan Liang. UniTraj: Learning a universal trajectory foundation model from billion-scale worldwide traces. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[80] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In *International Conference on Machine Learning*, pages 11692–11702. PMLR, 2020.

A Formal description of TraXion

This section restates the architecture and pre-training losses of §4 in self-contained mathematical form. All notation is inherited from §3 and §4; symbols introduced here are local to the appendix and italicized in the index below. Figure 1 in §4.4 gives a schematic of the full pre-training pipeline; this section gives the same content in equations.

Notation. We write $u \in \mathcal{U}$ for an entity, $\mathcal{X}$ for the substrate with embedding $\phi : \mathcal{X} \to \mathbb{R}^{d_x}$, and $e_{u,t} = (u, x_{u,t}, \tau_{u,t}, a_{u,t}, \delta_{u,t})$ for the t-th event of u, with $a_{u,t} \in \Delta^{n_a-1}$ a probability vector over n_a activity categories and $\delta_{u,t} \in \mathbb{R}_{\geq 0} \cup \{\perp\}$ an optional duration. A pre-training example consists of a chronologically ordered window $E_u = (e_{u,1}, \ldots, e_{u,T})$ of length T and, for each event t, a tuple of co-occuring peers from other entities (defined below). Let d denote the model dimension and F the number of feature tokens per event, with $d_f = d/F$ the per-token width. The encoder f_θ produces contextualized embeddings $\mathcal{H}_{u,t} \in \mathbb{R}^d$ as in Eq. 1; the prototype table is $P \in \mathbb{R}^{|\mathcal{U}| \times d_f}$ with rows p_u.

A.1 Per-event feature embedding

Each event $e_{u,t}$ is mapped to feature tokens $\big(\mathbf{z}^{\text{loc}}_{u,t}, \mathbf{z}^{\text{start}}_{u,t}, \mathbf{z}^{\text{stop}}_{u,t}, \mathbf{z}^{\text{ent}}_{u,t}, \mathbf{z}^{\text{act}}_{u,t}\big) \in (\mathbb{R}^{d_f})^F$ that are stacked along a feature axis. We use $F = 5$ on every instance except the two EHR corpora, where the backbone width d=128 (with H=4 heads and 2 layers) is not divisible by 5; on those two we drop $\mathbf{z}^{\text{ent}}_{u,t}$ from the input stack, leaving $F = 4$. The prototype p_u is still drawn from the same table P and used as the supervisory target in Eq. 3, so the contrastive coupling between the encoder and entity identity is preserved; the consequence is that A2's input-side entity-feature clause (§3.2) is unmet on the two EHR corpora while the supervisory-side clause holds. The smallest width that retains the entity token at the same head count is d=160 ($\sim 25\%$ more parameters at the same depth), which was not run within our compute envelope; we record this as a follow-up in Appendix G. Point events ($\delta_{u,t} = \perp$) keep both time tokens with $\mathbf{z}^{\text{stop}}_{u,t} = \mathbf{z}^{\text{start}}_{u,t}$ rather than collapsing to a single token.

Spatial token (Space2Vec). Let $\{a_1, a_2, a_3\}$ be the unit vectors of a planar hexagonal lattice with $a_1 = (1, 0)$, $a_2 = (-\frac{1}{2}, \frac{\sqrt{3}}{2})$, $a_3 = (-\frac{1}{2}, -\frac{\sqrt{3}}{2})$, and let

$$\lambda_s = \lambda_{\min} \left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{s/(N_s-1)}, \qquad s = 0, \ldots, N_s - 1, \tag{4}$$

be a geometric sweep of spatial scales over $[\lambda_{\min}, \lambda_{\max}]$. Define the multi-scale phase $\rho_{j,s}(x) = \langle a_j, \phi(x) \rangle / \lambda_s$ for $j \in \{1, 2, 3\}$. Stacking the $6N_s$ values $(\cos \rho_{j,s}, \sin \rho_{j,s})$ into a vector $\text{PE}(x) \in \mathbb{R}^{6N_s}$ and applying a learned linear-ReLU map gives

$$\mathbf{z}^{\text{loc}}_{u,t} \;=\; \text{ReLU}\big(W_{\text{loc}}\, \text{PE}(x_{u,t})\big) \in \mathbb{R}^{d_f}, \qquad W_{\text{loc}} \in \mathbb{R}^{d_f \times 6N_s}. \tag{5}$$

Temporal tokens (Time2Vec). Let $\Pi : \mathbb{R} \to [0, P)$ denote modular wrapping by the dataset-specific period P (`daily`, `weekly`, or `none`; Appendix B.1.5). With learned $w \in \mathbb{R}^{d_f}$, $b \in \mathbb{R}^{d_f}$, the Time2Vec encoding is the channel-wise map

$$\text{T2V}(\tau)_k = \begin{cases} w_0\, \Pi(\tau) + b_0, & k = 0, \\ \sin\big(w_k\, \Pi(\tau) + b_k\big), & k = 1, \ldots, d_f - 1, \end{cases} \tag{6}$$

giving $\mathbf{z}^{\text{start}}_{u,t} = \text{T2V}(\tau_{u,t})$ and $\mathbf{z}^{\text{stop}}_{u,t} = \text{T2V}(\tau_{u,t} + \delta_{u,t})$, with the convention $\delta_{u,t}$=0 when $\delta_{u,t} = \perp$ (point events have $\mathbf{z}^{\text{stop}}_{u,t} = \mathbf{z}^{\text{start}}_{u,t}$).

Entity token (factored prototype). The prototype table is factored as $p_u = W_P\, q_u$ with a small lookup $q_u \in \mathbb{R}^h$ and a shared projection $W_P \in \mathbb{R}^{d_f \times h}$, so that

$$\mathbf{z}^{\text{ent}}_{u,t} \;=\; p_u \;=\; W_P\, q_u \;\in\; \mathbb{R}^{d_f}. \tag{7}$$

The same p_u is used as input feature here and as supervisory target in Eq. 3 (§4.3); only $\{q_u\}_{u \in \mathcal{U}}$ and W_P are stored. Setting $h \ll d_f$ controls the parameter cost when $|\mathcal{U}|$ is large.

Activity token. With $W_a \in \mathbb{R}^{d_f \times n_a}$,

$$\mathbf{z}^{\text{act}}_{u,t} \;=\; W_a\, a_{u,t} \;\in\; \mathbb{R}^{d_f}. \tag{8}$$

Per-event matrix. We collect the F tokens into a per-event matrix $Z_{u,t} = [\mathbf{z}^{\text{loc}}_{u,t}, \mathbf{z}^{\text{start}}_{u,t}, \mathbf{z}^{\text{stop}}_{u,t}, \mathbf{z}^{\text{ent}}_{u,t}, \mathbf{z}^{\text{act}}_{u,t}]^\top \in \mathbb{R}^{F \times d_f}$ and write $\mathbf{Z}_u = (Z_{u,1}, \ldots, Z_{u,T}) \in \mathbb{R}^{T \times F \times d_f}$ for the window.

A.2 Co-occurrence retrieval

For each focal event $e_{u,t}$ we form a peer set

$$\mathcal{N}_{u,t} \;=\; \operatorname{top}^{(C-1)}_{e\in\mathcal{P}(u,t)} \big|\tau-\tau_{u,t}\big| + \big|(\tau+\delta)-(\tau_{u,t}+\delta_{u,t})\big|, \tag{9}$$

where $\mathcal{P}(u,t)=\{e_{u',t'} : x_{u',t'}=x_{u,t},\ u'\neq u\}$ is the set of training events from *other* entities at the same substrate context (taking $\delta=0$ when $\delta=\perp$), $\operatorname{top}^{(C-1)}$ keeps the $C-1$ smallest values, and $\mathcal{N}_{u,t}$ is padded to size $C-1$ with masked dummies when fewer than $C-1$ candidates exist (Appendix B.1.4). Let $Z^{(c)}_{u,t}$ be the per-event feature matrix of the c-th peer for $c=1,\dots,C-1$, and write the focal slot as $Z^{(0)}_{u,t}=Z_{u,t}$. Stacking peers along a co-occurrence axis yields the rank-5 tensor

$$X^{(0)}_u \;=\; \big[\, Z^{(0)}_{u,t}, Z^{(1)}_{u,t}, \dots, Z^{(C-1)}_{u,t} \,\big]_{t=1}^{T} \;\in\; \mathbb{R}^{T\times F\times C\times d_f}, \tag{10}$$

together with a peer mask $M_u\in\{0,1\}^{T\times C}$ ($M_{u,t,0}=0$ always; $M_{u,t,c}=1$ for padded slots) and a sequence padding mask $m_u\in\{0,1\}^T$ from window padding.

A.3 Factorized attention block

Let $\mathrm{TF}^{(\mathrm{ax})}_{H,d_{\mathrm{ff}}}$ denote a standard pre-LN Transformer encoder layer (multi-head attention with H heads followed by a position-wise MLP of width d_{ff}, with residual connections and LayerNorm) applied to a tensor by treating axis ax as the token axis and all remaining non-embedding axes as independent batch axes. The TraXionbackbone stacks L blocks $X^{(\ell+1)}_u=\mathrm{Block}_{\ell+1}(X^{(\ell)}_u)$, $\ell=0,\dots,L-1$, where each block consists of three axis-restricted sub-layers acting on $X\in\mathbb{R}^{T\times F\times C\times d_f}$:

$$Y \;=\; \mathrm{TF}^{(F)}_{H,d}(X), \qquad \text{(feature-axis attention; all } T\cdot C \text{ slots)} \tag{11}$$

$$\widetilde{Y}_{t,f,0,:} \;=\; \mathrm{XAttn}\big(\, Y_{t,f,0,:};\ \{Y_{t,f,c,:}\}_{c=0}^{C-1};\ M_{u,t,:}\,\big), \quad \text{(focal} \to \text{peer cross-attention)} \tag{12}$$

$$\widetilde{Y}_{t,f,c,:} \;=\; Y_{t,f,c,:} \quad (c\geq 1),$$

$$X'_{:,:,0,:} \;=\; \mathrm{TF}^{(T)}_{H,d}\big(\widetilde{Y}_{:,:,0,:};\ m_u\big), \qquad \text{(sequence-axis attention; focal slot only)} \tag{13}$$

$$X'_{:,:,c,:} \;=\; \widetilde{Y}_{:,:,c,:} \quad (c\geq 1).$$

$\mathrm{XAttn}(q;\,K;\,M)$ is a single multi-head cross-attention sub-layer (with residual, LayerNorm, and position-wise MLP) where q is the focal query, $\{K_c\}$ are the peer keys/values, and M is the key-padding mask; the asymmetric query-only update rewrites only the focal slot's representation in this sub-layer (peer slots retain $Y_{t,f,c,:}$ for $c\geq 1$); each peer event is itself the focal slot in its own training window, so a symmetric self-attention here would be redundant and $C\times$ more expensive.

After L blocks, the contextualized event embedding is the per-event flattening of the focal slot's F tokens:

$$\mathcal{H}_{u,t} \;=\; \mathrm{vec}\big(X^{(L)}_{u,t,:,0,:}\big) \;\in\; \mathbb{R}^d, \qquad H_u=(\mathcal{H}_{u,1},\dots,\mathcal{H}_{u,T})\in\mathbb{R}^{T\times d}. \tag{14}$$

This $\mathcal{H}_{u,t}$ is the same vector referenced by Eq. 1–3.

A.4 Pre-training heads and the perturbation operator

Noise-detection head. A linear head $g_\psi:\mathbb{R}^d\to\mathbb{R}$ scores each event for being perturbed:

$$g_\psi(\mathcal{H}_{u,t}) \;=\; w_\psi^\top \mathcal{H}_{u,t} + b_\psi. \tag{15}$$

Eq. 2 is then the masked binary cross-entropy of $g_\psi(\mathcal{H}_{u,t})$ against the per-event target $y_{u,t}$ produced by η, averaged over the non-padded events $\mathcal{V}=\{(u,t) : m_{u,t}=0\}$.

Prototype-contrastive projection. The InfoNCE anchor $\bar{\mathcal{H}}_{u,t}$ in Eq. 3 is obtained by passing $\mathcal{H}_{u,t}$ through a two-layer MLP $\pi_\theta : \mathbb{R}^d \to \mathbb{R}^{d_f}$ and ℓ_2-normalizing,

$$\pi_\theta(h) \;=\; W_2\,\mathrm{ReLU}(W_1 h + b_1) + b_2, \qquad \bar{\mathcal{H}}_{u,t} \;=\; \frac{\pi_\theta(\mathcal{H}_{u,t})}{\|\pi_\theta(\mathcal{H}_{u,t})\|_2}, \qquad \bar{p}_u \;=\; \frac{p_u}{\|p_u\|_2}, \tag{16}$$

with $W_1 \in \mathbb{R}^{h_{\mathrm{proj}} \times d}$, $W_2 \in \mathbb{R}^{d_f \times h_{\mathrm{proj}}}$. Anchors are restricted to $\mathcal{V}^+ = \{(u,t) \in \mathcal{V} : y_{u,t} = 0\}$ (Appendix B.1.2); the denominator of Eq. 3 ranges over the prototypes of all minibatch entities $B \subseteq \mathcal{U}$ that have at least one anchor.

Perturbation operator. The operator η of §4.2 is a stochastic mapping $\eta : \mathcal{E}^T \to \mathcal{E}^T \times \{0,1\}^T$ on event windows. Let $\mathrm{Bern}(p)$ denote a Bernoulli draw and Unif a uniform draw on the indicated set. Sample $b \sim \mathrm{Bern}(1 - p_{\mathrm{norm}})$; if $b = 0$ return $(E_u, \mathbf{0}_T)$. Otherwise draw an indicator vector $y_u \in \{0,1\}^T$ with $y_{u,t} \overset{\mathrm{iid}}{\sim} \mathrm{Bern}(r)$ at a fixed rate $r = 0.3$, forcing $\sum_t y_{u,t} \geq 1$. For each flagged t (i.e. $y_{u,t} = 1$), draw $\kappa_{u,t} \sim \mathrm{Unif}\{\mathrm{loc}, \mathrm{time}, \mathrm{both}\}$ and apply

$$\eta_{\mathrm{loc}}(e_{u,t}) :\; x'_{u,t} \;=\; \arg\min_{x \in \mathcal{X}} \big\|\phi(x) - \tilde{\phi}\big\|_2, \quad \tilde{\phi} \sim \mathrm{Unif}(\mathrm{AOI}), \quad a'_{u,t} = a(x'_{u,t}), \tag{17}$$

$$\eta_{\mathrm{time}}(e_{u,t}) :\; \tau'_{u,t} \;\sim\; \mathrm{Unif}\big(\tau_{u,t-1} + \delta_{u,t-1},\; \tau_{u,t+1}\big), \quad x'_{u,t} = x_{u,t}, \tag{18}$$

applied separately or in composition $\eta_{\mathrm{loc}} \circ \eta_{\mathrm{time}}$ depending on $\kappa_{u,t}$. When the interval in Eq. 18 is empty — e.g. concurrent events with $\tau_{u,t-1} + \delta_{u,t-1} > \tau_{u,t+1}$, or boundary positions $t \in \{1, T\}$ where a neighbor is undefined — η_{time} falls back to the identity on that event and $y_{u,t}$ is set to 0. AOI $\subset \mathbb{R}^{d_x}$ is the dataset's area-of-interest in ϕ-space, and $a(x)$ denotes the activity associated with substrate context x. The output of η is the corrupted window $\tilde{E}_u$ and label vector y_u used in Eq. 2.

A.5 Total objective

For a minibatch of windows $\{E_u\}_{u \in B}$, sample $(\tilde{E}_u, y_u) = \eta(E_u)$ independently per entity and compute $\{\mathcal{H}_{u,t}\}$ via Eq. 14. With Eq. 2 and Eq. 3 the joint pre-training loss of Eq. 1 reads

$$\mathcal{L}(\theta, \psi) \;=\; \underbrace{\frac{1}{|\mathcal{V}|} \sum_{(u,t) \in \mathcal{V}} \mathrm{BCE}\big(g_\psi(\mathcal{H}_{u,t}),\, y_{u,t}\big)}_{\mathcal{L}_{\mathrm{noise}}(\theta,\psi)} \;+\; \gamma \cdot \underbrace{\frac{1}{|\mathcal{V}^+|} \sum_{(u,t) \in \mathcal{V}^+} -\log \frac{\exp\big(\bar{\mathcal{H}}_{u,t}^\top \bar{p}_u / \beta\big)}{\sum_{u' \in B} \exp\big(\bar{\mathcal{H}}_{u,t}^\top \bar{p}_{u'} / \beta\big)}}_{\mathcal{L}_{\mathrm{prototype}}(\theta)}. \tag{19}$$

Optimization uses AdamW with cosine decay; concrete values of $T, F, C, L, H, d, d_f, h, h_{\mathrm{proj}}, N_s, [\lambda_{\min}, \lambda_{\max}], P, \lambda, \beta, p_{\mathrm{norm}}$ are listed in Appendix B.2.

B Implementation details of TraXion

B.1 Pre-training mechanics

B.1.1 Selection and composition of pre-training perturbation

Within a corrupted window we flag a random proportion of events for perturbation; each flagged event is then assigned context perturbation, time perturbation, or both, with equal probability. If no event ends up flagged (for example, because rejection sampling fails on a short window), we force at least one so that the denoising objective is non-trivial on every corrupted example. If both perturbation types are disabled the operator degenerates to the identity and Eq. 2 is masked out.

B.1.2 Anchoring the prototype loss on unperturbed events only

Negative examples in Eq. 3 are drawn from minibatch entities, but anchors — the event representations $\bar{\mathcal{H}}_{u,t}$ that appear in the InfoNCE numerator — are restricted to the events the denoising operator η left unchanged ($\mathcal{V}^+$). A corrupted anchor would pull $\bar{p}_u$ toward an event the entity did not actually produce, blurring the prototype's role as a summary of

the entity's normal behavior. Restricting the anchor set is what keeps the two pre-training objectives compatible: the denoising objective uses corrupted events as positive supervision for detecting perturbations, while the prototype objective uses unperturbed events as positive supervision for binding events to their entities.

B.1.3 Why the shared-prototype design does not collapse to an identity shortcut

With p_u as an input token and residual connections in the backbone, a trivial identity path could drive Eq. 3 to zero; joint training with $\mathcal{L}_{\text{noise}}$ (Eq. 2) prevents this by forcing the encoder to model the joint context/time/activity distribution rather than copy p_u.

B.1.4 Co-occurrence retrieval

The co-occurrence axis is assembled by a non-parametric lookup over a context-keyed inverted index of substrate events. At load time, for each focal event $e_{u,t}$ we query the index for co-occurring events at context $x_{u,t}$ (excluding events of the focal entity u), optionally filter to those whose $[\tau, \tau + \delta]$ interval overlaps $[\tau_{u,t}, \tau_{u,t} + \delta_{u,t}]$ by at least a threshold fraction, and rank the remainder by the Eq. 9 distance; the top $C-1$ are kept and the rest are padded and masked out of attention. Each retrieved row contributes the substrate tuple of §3 into the cross-attention as keys/values, so the focal stream sees only event-tuple features from peers; the index is built once per data partition (separately for train, val, and test; a one-shot precompute) and the padding mask is carried through every co-occurrence sub-layer so events with no co-occurring peers are handled gracefully.

Cost and scaling of the retrieval. The inverted index is a hash table mapping each context $x \in \mathcal{X}$ to the row positions of the training events at x, built in one linear pass over the partition with $\mathcal{O}(N)$ time and $\mathcal{O}(N)$ memory (4 bytes per event), where N denotes the number of training events; on NUMOSIM-LA ($N \approx 3.4 \times 10^7$ events over $|\mathcal{X}| = 920{,}360$ contexts; Table 9) the index itself occupies $\sim$140 MB and is paid once at dataset construction. A single peer query for focal event $e_{u,t}$ costs an $\mathcal{O}(1)$ hash lookup of the bucket at $x_{u,t}$, an $\mathcal{O}(k_x)$ self-exclusion and optional overlap filter, and an $\mathcal{O}(k_x \log k_x)$ partial sort over the bucket to select the $C-1$ peers minimizing the Eq. 9 distance, where k_x denotes the per-bucket fan-in (the number of training events at context x). Per-query cost therefore scales with the per-bucket fan-in k_x: the average fan-in $N/|\mathcal{X}|$ across our datasets is ≈ 37 (NUMOSIM-LA), ≈ 10 (Foursquare-Tokyo), and ≈ 86 (LANL-auth). Retrieval runs inside CPU DataLoader workers in parallel with GPU forward/backward, keeping the lookup off the model's critical path at our settings; at far larger scale ($N = 10^9$ events, $|\mathcal{X}| = 10^7$ contexts) the index reaches $\sim$4 GB and per-query cost continues to scale with bucket fan-in, and standard mitigations (sharding $\mathcal{X}$ across workers, replacing the partial sort with $\mathcal{O}(k_x)$ partitioning, or capping each bucket to a fixed reservoir) preserve the structure of Eq. 9.

B.1.5 Design rationale

Location and temporal perturbations. Before encoding, we wrap the raw timestamp t (hours from the dataset origin) by a per-dataset period: $t \leftarrow t \bmod 24$, $t \leftarrow t \bmod 168$, or no wrapping, selected by validation loss (Table 7). Location perturbation dominates the pre-training denoising signal because it samples city-scale coordinates, producing a much larger Euclidean shift in Space2Vec space than time perturbation's feasible-window resample.

Empirical validation. In a matched UA-Berlin pre-train (identical recipe, 300 epochs), final validation total-loss was 0.052 at $\bmod 24$, 0.201 at $\bmod 168$, and 0.192 without wrapping (Figure 2); the per-dataset choices in Table 7 come from analogous sweeps.

B.2 Hyperparameters

We report the settings used to produce Tables 1, 2, and 3. All runs share the architecture and loss defaults below, except for the backbone width and depth of the two ICU corpora; the dataset- and task-specific settings are listed in Tables 7 and 8.

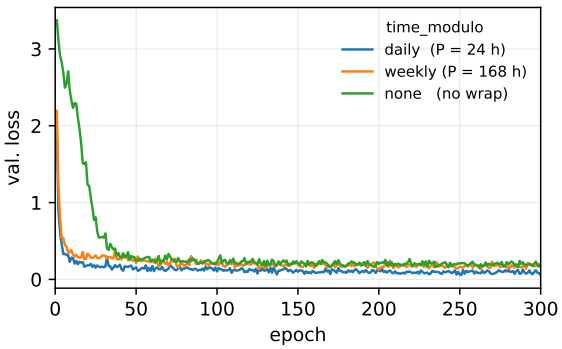

Figure 2: UA-Berlin pre-training validation loss under three wrap settings (identical recipe, 300 epochs). Wrapping at 24 h converges to roughly 4× lower loss than wrapping at 168 h or no wrapping on this dataset; the wrap period is a per-dataset hyperparameter.

Shared architecture and loss. We set γ in Eq. 1 to 0.5, $p_{\text{norm}} = 0.7$, β in Eq. 3 to 0.07, N_s in §4.4 to 32, and h in §4.4 to 32. The factorized attention in §4.4 stacks $L = 6$ blocks with $H = 4$ heads and a feed-forward width of 1024; we use $d = 1040$ (auto-adjusted from a target of 1024 so that d/F is integral). The two ICU corpora use $d = 128$ and $L = 2$ (Appendix A.1). Sequences are chunked into non-overlapping windows of $T = 32$ events; co-occurrence retrieval uses no minimum overlap filter. The two ICU corpora are the one exception: they pre-train at $T = 32$ but fine-tune at $T = 128$, because the mortality label is defined over a whole stay rather than a window of it.

Per-dataset pre-training settings. Pre-training uses AdamW [39] with peak learning rate 2×10^{-4}, gradient-norm clipping at 1.0, and a cosine schedule decaying to $\eta_{\min}=10^{-6}$ over the full budget. Early stopping monitors an EMA-smoothed total validation loss (EMA factor 0.1). Table 7 lists the dataset-varying settings: co-occurrence window size C, Space2Vec scale range $[\lambda_{\min}, \lambda_{\max}]$, the wrap period applied to timestamps before Time2Vec [29] (selected by validation loss), weight decay, maximum epoch budget, and patience.

Dataset	C	$[\lambda_{\min}, \lambda_{\max}]$	modulo	weight decay	max epochs	patience
NUMOSIM-LA	8	$[10^{-6}, 2]$	weekly	10^{-3}	200	20
UA-Atlanta	8	$[10^{-6}, 2]$	daily	10^{-3}	200	40
UA-Berlin	4	$[10^{-6}, 2]$	daily	10^{-2}	800	80
Foursquare-Tokyo	8	$[10^{-3}, 360]$	daily	10^{-2}	400	40
Gowalla-Stockholm	4	$[10^{-3}, 360]$	daily	10^{-2}	500	80
Gowalla-Austin	4	$[10^{-3}, 360]$	daily	10^{-2}	500	80
LANL	8	$[10^{-3}, 2]$	daily	10^{-2}	200	40
eICU-CRD	4	$[10^{-6}, 2]$	daily	10^{-3}	80	15
MIMIC-IV	4	$[10^{-6}, 2]$	daily	10^{-3}	80	15

Table 7: Per-dataset pre-training hyperparameters. All other settings are shared across datasets.

Fine-tuning. Every fine-tuning run starts from the pre-trained checkpoint and inherits its architecture arguments. Fine-tuning uses AdamW with weight decay 10^{-2}, a cosine schedule, and gradient-norm clipping at 1.0. The learning rate is 10^{-4} on the mobility tasks, 5×10^{-5} on eICU-CRD, and 2×10^{-4} on MIMIC-IV (Table 8). Early stopping monitors an EMA (factor 0.1) of the validation criterion in Table 8, except on Gowalla-Austin social-link prediction and the two ICU corpora, which monitor the raw value. Task-specific settings are summarized in Table 8 and detailed below.

Anomaly detection. A three-layer MLP head on $\mathbf{H}_{u,t}$ produces a per-visit anomaly logit, trained with the BCE term of Eq. 2; the user-prototype loss is disabled at fine-tune and we monitor BCE validation loss for early stopping. Prototype sharing (§4.4) keeps the visit features user-aware. Perturbations at fine-tune time are dataset-specific: NUMOSIM-LA uses inserted-visit anomalies [53]; UA-Berlin and UA-Atlanta use the "behavioral anomaly" regime [1] (hunger/interest/social/work). For both NUMOSIM and Urban Anomalies, finetuning is unsupervised as the perturbations are generated on-the-fly. NUMOSIM and Urban Anomalies are simulators whose anomaly definitions are design choices of their

respective papers, and η's fine-tune perturbations follow each benchmark's own generation regime. For LANL we report only the unsupervised regime (the LANL benchmark convention — every log-anomaly baseline is itself unsupervised; Appendix D.5): the pre-trained backbone is scored directly, with no LANL fine-tuning, by rank-fusing its noise-detection and prototype-contrastive heads (Appendix D.6).

Next-POI recommendation. A two-layer projection maps $\mathbf{H}_{u,t}$ to a query vector that is scored against a learned POI embedding table via sampled-softmax cross-entropy with n_{neg}=256 uniform POI negatives per query at temperature 0.1; head dropout 0, label smoothing 0, no LR warmup.

Next-visit prediction. Joint location and time. The location head is identical to the next-POI head above. The time head is a 2-layer MLP that maps $[\,\mathbf{q}_{u,t};\ \mathbf{e}_{\text{POI}}(x_{u,t+1});\ \text{Time2Vec}(\tau_{u,t})\,]$ — the encoder query, the embedding of the ground-truth next POI, and a Time2Vec encoding of the current visit's start time — to the parameters of a K=8-component 1D Gaussian mixture over $\log(1+\Delta_{u,t})$, where $\Delta_{u,t} = \tau_{u,t+1} - \tau_{u,t}$ in hours, with softplus mixture weights and scales and real-valued means. Training adds the GMM NLL, with weight 0.3, to the next-POI sampled-softmax loss. The GMM mode is the point prediction; T±60 is the mass of the predicted density within ±60 min of $\Delta_{u,t}$.

Social-link prediction. Masked mean- and max-pooling over each user's valid visits yields a per-user embedding $\mathbf{e}_u$; a symmetric pair scorer over $[\mathbf{e}_u{+}\mathbf{e}_v,\ |\mathbf{e}_u{-}\mathbf{e}_v|,\ \mathbf{e}_u{\odot}\mathbf{e}_v]$ concatenated with 8 handcrafted co-visitation features predicts whether (u, v) is a social tie. The objective combines BPR (weight 1.0) with a supervised InfoNCE term (weight 0.3, temperature 0.1) over n=8 in-batch negatives per positive edge, for 200 edge-centric batches per epoch; per-user embedding dim is 128 and the pair-head hidden width is 128 with dropout 0.1. The 8 handcrafted co-visitation features are normalized with LayerNorm on Foursquare-Tokyo and Gowalla-Stockholm and with per-feature z-scores on Gowalla-Austin; per-dataset variants, tuned on the validation AUC of each city subset, adjust the pair head: Foursquare-Tokyo uses a split pair-head (separate embedding-MLP and linear handcrafted-feature branches), Gowalla-Austin adds an auxiliary random-negative BCE loss on the handcrafted features (weight 1.0, batch 256), and Gowalla-Stockholm uses the default joint pair-head. At test time the social logits are late-fused with a logistic-regression baseline on the 8 handcrafted features: $s = (1{-}\alpha)\,\sigma(z_{\text{model}}) + \alpha\,\sigma(z_{\text{LR}})$ with $\alpha \in \{0, 0.1, \ldots, 1\}$ picked per seed on validation AUC (0.1 on every Gowalla seed; 0.4 or 0.5 on Foursquare-Tokyo).

ICU mortality prediction. The mortality fine-tune head pools the per-event TraXion representations of the stay's events with masked mean+max pooling, concatenates the result with a 52-dim per-stay tabular vector, and passes the concatenation through a two-layer MLP and a sigmoid for binary BCE training (`pos_weight`= 5.0). The tabular vector carries: admission demographics (age, sex one-hot, height, admit/discharge weight, BMI, unit-visit number), one-hot indicators for the top-6 ethnicities, top-8 `unittype`s, top-4 `unitstaytype`s, and top-15 APACHE admission diagnoses, and per-table event aggregates (log-counts per source table, distinct-context count, total event count, stay duration in hours). Z-score statistics are fit on the train+val slice and applied to all stays. Because mortality is a 9–12% minority in both corpora, fine-tuning uses a class-balanced `WeightedRandomSampler` that balances positives and negatives in each batch, a positive class weight of 5 in the BCE loss, and a deterministic 4-window evaluation in which each stay's score is the mean predicted probability across 4 evenly-spaced sub-windows; the entity input is zeroed at fine-tune time, because the patient-level split leaves test patients without a prototype. Early stopping monitors the raw validation AUROC.

B.3 Hyperparameter selection procedure

The settings reported in Tables 7 and 8 (and in the per-baseline appendices) are the outcome of a validation-guided iterative search rather than a pre-committed grid or random sweep. For each (method, dataset, task) cell we initialize from the published defaults — or, when a baseline ships no defaults for the cohort in question, from sensible literature priors — and then run successive small batches of configurations (typically 4–8 runs each) that vary one or two hyperparameters at a time. Each batch is scored by the same validation criterion used for early stopping (BCE validation loss for anomaly fine-tunes, total fine-tune loss for

Table 8: Per-task, per-dataset fine-tuning budgets. "criterion" is the validation quantity monitored for early stopping: *BCE* is the denoising BCE loss, *total* is the full fine-tuning loss, and AUC/AUROC are the validation ranking metrics. Train/val batch sizes are 64/256 for anomaly, next-POI, and next-visit, 64/64 for social, and 32/64 for mortality.

Task	Dataset	lr	max epochs	patience	criterion
Anomaly	NUMOSIM-LA	10^{-4}	100	10	BCE
	UA-Atlanta	10^{-4}	250	50	BCE
	UA-Berlin	10^{-4}	500	50	BCE
Next-POI	Foursquare-Tokyo	10^{-4}	100	15	total
	Gowalla-Stockholm	10^{-4}	100	15	total
	Gowalla-Austin	10^{-4}	200	80	total
Next-visit	Foursquare-Tokyo	10^{-4}	200	40	total
	Gowalla-Stockholm	10^{-4}	200	40	total
	Gowalla-Austin	10^{-4}	200	40	total
Social	Foursquare-Tokyo	10^{-4}	50	15	AUC
	Gowalla-Stockholm	10^{-4}	50	15	AUC
	Gowalla-Austin	10^{-4}	50	15	AUC
Mortality	eICU-CRD	5×10^{-5}	25	8	AUROC
	MIMIC-IV	2×10^{-4}	25	8	AUROC

next-POI / next-visit, validation AUC for social-link, validation AUROC for ICU mortality, EMA-smoothed total validation loss for pre-training; Tables 7–8), and the next batch's settings are chosen from the run-by-run trajectory of the preceding batch. The reported configuration per cell is the highest-validation point reached under this procedure.

The selection loop is orchestrated by a Claude Code agent [2], in the spirit of Karpathy [28]: the agent reads the prior batch's logs and validation curves, proposes the next batch's hyperparameter settings under our search criterion, and launches the runs via the same training scripts a human would use. This replaces the bookkeeping of a sequential coordinate-descent search with an automated proposer but does not change the search space, the validation criterion, or the protocol's outcome — every reported number comes from the unmodified training/evaluation code in the released code. We apply the same selection procedure to TraXion and to all retuned baselines (Appendix D) to keep the comparison balanced.

The standing prompt — used identically for TraXion and every retuned baseline — instructs the agent to tune on the validation criterion only, document each change and its effect rather than running random or grid search, and log the best configuration; the released code ships the verbatim prompt.

C Data and evaluation setup

C.1 Dataset statistics

Table 9 reports per-dataset counts. What counts as an *entity* differs by instance: the simulated agent for NUMOSIM-LA and Urban Anomalies, the user for the three check-in datasets (Foursquare-Tokyo, Gowalla-Stockholm, Gowalla-Austin), the `src_user` for LANL, and the patient (`uniquepid` in eICU-CRD, `subject_id` in MIMIC-IV) for the two EHR corpora. Mobility rows are reported *after* the ≥ 5 train-split check-in filter that every model is trained and evaluated under. For the EHR corpora the prediction unit is the ICU stay while the entity is the patient — 197,307 stays across 138,176 patients for eICU-CRD and 94,407 stays across 65,360 patients for MIMIC-IV — so the *anomalous* column counts patients who died in hospital (14,592 and 9,465); the 17,673 and 11,333 deaths in §5.5 count ICU stays, since every stay of a fatal hospitalization is labeled positive. Social links are listed only for datasets that ship an explicit social relationship graph.

C.2 Dataset licenses and provenance

We use only publicly released datasets, each cited in the bibliography and consumed under its published terms. Table 10 summarizes the source repository and license of each dataset.

Table 9: Dataset statistics.

Dataset	Events	Contexts	Entities		Social Links
			Total	Anomalous	
NUMOSIM-Los Angeles	34,328,711	920,360	200,000	381	—
Urban Anomalies-Berlin	298,476	1,204	1,000	120	—
Urban Anomalies-Atlanta	310,441	1,222	1,000	120	—
Foursquare-Tokyo	1,512,647	153,762	7,456	—	48,047
Gowalla-Stockholm	416,539	52,349	7,759	—	56,808
Gowalla-Austin	334,476	20,687	5,397	—	52,586
Non-mobility					
LANL-auth cohort	190,944	2,214	1,598	57	—
eICU-CRD	120,786,921	507	138,176	14,592	—
MIMIC-IV	54,560,796	520	65,360	9,465	—

We do not redistribute any raw dataset; the released code contains preprocessing scripts that read each dataset from its original repository.

Table 10: Dataset sources and licenses. "Citation required" indicates the dataset is publicly released for research with a citation requirement to the original paper but no formally named open license; we cite each in the references and use the data only as released.

Dataset	Source	License	Citation
NUMOSIM-LA	OSF `osf.io/sjyfr`	Public release; no formal license	Stanford et al. [53]
Urban Anomalies-Berlin	OSF `osf.io/dg6t3`	CC-BY 4.0 (per OSF record)	Amiri et al. [1]
Urban Anomalies-Atlanta	OSF `osf.io/dg6t3`	CC-BY 4.0 (per OSF record)	Amiri et al. [1]
Foursquare-Tokyo	D. Yang's homepage	Citation required	Yang et al. [67]
Gowalla-Stockholm	SNAP (Stanford)	Citation required	Cho et al. [8]
Gowalla-Austin	SNAP (Stanford)	Citation required	Cho et al. [8]
LANL Auth. Log	`csr.lanl.gov/data/cyber1`	CC0 / Public domain	Kent [30]
eICU-CRD	PhysioNet	PhysioNet Credentialed Health Data License	Pollard et al. [47]
MIMIC-IV	PhysioNet	PhysioNet Credentialed Health Data License	Johnson et al. [26]

Code-side baselines are run from each authors' released implementation under the licenses those repositories ship with (Appendix D); we cite each in the corresponding subsection and the bibliography. Repository licenses, where the upstream repository ships an explicit LICENSE file or in-README license declaration, are: UniTraj (Apache 2.0); LoTNext, TAPT, THP, A-NHP, TrajGPT, Raindrop, WaveGNN, and STraTS (MIT); NuLog (BSD 3-Clause); and LogGPT (CC-BY-NC-4.0). The repositories of GETNext, MobTCast, Mobility-LLM, EBM, USRC, LBSN2Vec, H^{3}GNN, CoBAD, DeepBayesic, USTAD, ICAD, and SimMTM do not ship an explicit LICENSE file or in-README license declaration; we use each upstream repository unmodified except for cohort/split adapters (and, for ICAD, the score computation of Appendix D.2), cite the original paper, and would defer to any forthcoming license the upstream authors publish. COAST has no upstream code release; our re-implementation (Appendix D.3) follows the published paper. Tuor-AAAI17, DeepLog, and LogBERT are PyTorch re-implementations from the published descriptions (Appendix D.5), with the original papers cited.

C.3 Dataset splits

Splits follow each dataset's released protocol. NUMOSIM and Urban Anomalies ship a test split with visit- or agent-level anomaly labels; the three LBSN city subsets use a temporal train/test split (first 90% / last 10% of the visit stream) for next-POI recommendation and next-visit prediction, and GETNext, LoTNext, MobTCast, and COAST hold out the latest 20% of the training window as validation; social-link prediction uses a 70/15/15 edge split (seed 42) with 100 sampled non-edges per test positive for ranking metrics. LANL uses a temporal split at t=950,400 s (approximately day 11 of the 58-day log) yielding 45,806 train events (316 red-team) and 145,138 test events (197 red-team); the split is chosen to balance positives between train and test since red-team activity is concentrated in the first month. Both EHR corpora use a patient-level 70/15/15 stratified split (seed 42) — on `uniquepid` for eICU-CRD — so all stays of the same patient land in a single fold; the per-stay binary mortality label is taken from the `patient.hospitaldischargestatus` field for eICU-CRD and from in-hospital death for MIMIC-IV. Pre-training and TraXion's anomaly, next-POI, and next-visit fine-tuning use the earliest 20% of each training set as validation. Pre-training consumes only the events in each dataset's training partition.

C.4 Next-POI and next-visit prediction targets

All methods in Table 2 are evaluated on the same test visits, but each method's evaluation code selects its own prediction targets from them and ranks each target against its own POI vocabulary, built from its training data. Table 11 lists each method's target rule, vocabulary, and number of targets. Targets are restricted to POIs in the method's vocabulary, and every method ranks against its full vocabulary.

Table 11: Prediction targets in Table 2.

Method	Targets	Vocabulary	Tokyo	Stockholm	Austin
TraXion (both tasks)	each test visit after the agent's first	latest 80% of train	132,224	23,983	21,957
UniTraj†	each test visit after the first, for agents with ≥4 test visits; one random 128-visit window per agent with ≥128	train ∪ test	120,112	25,134	23,270
GETNext	each visit after the first in 24-hour test sessions with ≥3 visits, for users present in the earliest 80% of train	earliest 80% of train	57,534	8,541	9,798
LoTNext	every test visit, including the first, for users with ≥20 check-ins in the earliest 80% of train	earliest 80% of train	123,183	12,240	12,993
MobTCast	each test visit after the agent's first	earliest 80% of train	128,059	20,543	20,766
COAST	each visit after the first in 24-hour test sessions with ≥2 visits, within the last 200 test visits (150 on Tokyo)	earliest 80% of train	72,192	14,587	15,270
TAPT	every test visit, including the first	all of train	137,966	27,461	23,806
TrajGPT	every test visit in 128-visit windows, the first predicted from padding	latest 80% of train	136,666	27,317	23,736
Mobility-LLM	each visit after the first among the agent's last 100 test visits; the last partial batch is dropped	earliest 80% of train	102,225	19,718	19,964
THP	each visit after the first among the agent's first 256 test visits	latest 80% of train	125,493	23,896	21,734
A-NHP	each visit after the first among the agent's last 256 test visits (128 on Tokyo)	earliest 80% of train	108,710	19,973	20,507

On the targets it shares with the strongest next-POI baseline (127,107, 10,736, and 12,048 targets), TraXion's next-POI H@10 is .4443 against MobTCast's .3827 on Foursquare-Tokyo, .3358 against LoTNext's .2362 on Gowalla-Stockholm, and .2875 against LoTNext's .2034 on Gowalla-Austin. On the targets it shares with each next-visit baseline, TraXion's next-visit H@10 and H@20 differ from its full-set values by at most .014 and T±60 by at most .04 (Foursquare-Tokyo on Mobility-LLM's targets: .1776 against .2181), and every comparison with that baseline in Table 2 keeps its direction.

C.5 LANL cohort and host coordinates

The raw `auth.txt.gz` log (1.05 B events) is impractical to pretrain on directly. We subsample a cohort of 4,000 `src_user`s consisting of all 104 users that appear as the source in at least one red-team event plus 3,896 uniformly sampled benign users; after applying a per-user event cap (500 benign events retained per user; *all* red-team rows kept regardless of cap) and dropping users with fewer than 10 surviving events, 1,598 users remain. POI (host) coordinates are obtained by running 2D UMAP [41] ($n_{\text{neighbors}}$=30, cosine distance) on the sparse user-by-host access-count matrix restricted to the cohort; the resulting 2,214 unique hosts are min-max scaled to $[0, 1]^2$. The UMAP is fit on the full cohort (train+val+test events). Each host's category is the dominant `(auth_type, logon_type)` pair observed at that host in the cohort, yielding 13 distinct categories. Anomaly labels are the real red-team events: an event is labeled 1 iff its (time, src_user, src_computer, dst_computer) tuple matches a row in `redteam.txt.gz`. Of the 749 raw red-team events, 513 survive the cohort and per-user filters (316 in train, 197 in test); 57 `src_user`s have at least one red-team row in the evaluated split, which we report as "Anomalous" in Table 9. Red-team events make up 197/145,138 = 0.136% of the evaluated test split, against 749/1.05B = 0.00007% of the raw log, an enrichment of roughly 1,900×. The event- and user-level metrics of Table 4 are computed against these red-team labels. TraXion pre-trains under the perturbation operator η (§4.2) and is scored without fine-tuning (Appendix D.6); each baseline trains under its own self-supervised objective.

C.6 EHR corpora and context coordinates

We use the PhysioNet releases of eICU-CRD [47] and MIMIC-IV [26], retaining stays with at least one event in our source tables: 197,307 stays from 138,176 patients for eICU-CRD and 94,407 stays from 65,360 patients for MIMIC-IV (Table 9). For eICU-CRD, events come from `lab`, `medication`, `treatment`, `infusiondrug`, and `vitalAperiodic`, retaining the top-K most frequent items per table (K=$\{150, 200, 100, 60, 12\}$ respectively), normalizing free-text drug names, and dropping rows with non-positive duration; medication strings are normalized by lowercasing, stripping dose/route tokens, and matching against an alias table built from the corpus. MIMIC-IV fills the same five streams, with the same K per stream, from `hosp/labevents`, `hosp/emar` (administered doses), `icu/procedureevents`, `icu/inputevents`, and a curated item subset of `icu/chartevents`. The resulting event streams cover 120,786,921 events over 507 unique contexts for eICU-CRD and 54,560,796 events over 520 contexts for MIMIC-IV. Each event is a $(\text{entity}, \text{context}, \text{time}, \text{category})$ tuple: the *patient* as entity, `<table>::<item>` as context, the offset minute from unit-admit as time (with stay-relative duration on $[t_{\text{start}}, t_{\text{stop}}]$), and the source table name as category (5 categories). Mortality is predicted *per stay*, the unit the label is defined on. Context coordinates are obtained by running 2D UMAP [41] ($n_{\text{neighbors}}$=30, cosine distance) on the (context $\times$ stay) co-occurrence matrix and min-max scaling to $[0, 1]^2$; the UMAP is fit on all stays (train+val+test). Of the 197,307 eICU-CRD stays, 17,673 have `hospitaldischargestatus`=Expired; 11,333 of the 94,407 MIMIC-IV stays end in in-hospital death. That per-stay binary label is the fine-tune target reported in Table 9 and Table 5.

D Baseline implementation

Unless noted otherwise, baselines are retrained on the same cohorts and splits as TraXion (Appendix C.3) using the released code. Where a baseline's published settings transfer to our cohorts we keep them unchanged; where they do not, we retune under the same validation-guided procedure applied to TraXion (Appendix B.3), and the retuned settings are the ones reported in the per-baseline sections below. This appendix documents the exceptions: a backbone, UniTraj, that we adapt across all tasks (§D.1); a next-POI baseline, COAST, whose authors do not release code and which we re-implement from the published paper (§D.3); and a task — enterprise authentication — of which some baselines we re-implement from scratch (§D.5).

D.1 UniTraj

We use the released UniTraj [79] encoder–decoder as a baseline backbone, but the published recipe assumes dense GPS traces ($\sim$1 Hz, $\geq$36 points per trajectory). Our datasets are sparse visit/event streams, so we make the following minimal, fairness-preserving modifications and apply them identically across all nine datasets.

Tokens are visits, not GPS samples. For each agent we sort their visits by start time and treat the resulting stream as one trajectory. Each token's spatial features are the visit's (lon, lat), and the time channel is $\Delta t_i = t_i - t_{i-1}$ in seconds, log-transformed via $\log(1 + \Delta t)$ to compress the large dynamic range (seconds to days) into the same scale UniTraj's $\text{Linear}(1, d)$ interval embedding was tuned for.

Per-dataset normalization on anchored deltas. UniTraj's released `Normalize` statistics were fit on dense 1 Hz traces and are not transferable. We refit per dataset on $(\text{lon} - \text{lon}_0, \text{lat} - \text{lat}_0)$ deltas (i.e. already anchored at the first visit of each agent), using a 0.5%/99.5% trimmed standard deviation to be robust to occasional bad coordinate rows; we also clip normalized values to $\pm 10\sigma$.

Pre-training objective and masking. We retain UniTraj's masked auto-encoding loss with mask ratio 0.5 and the original 1D-conv patch tokenizer (patch_size = 1 since visits are already aggregated). Of the four masking strategies in the original paper we keep *random*, *block*, and *last-N* with the same 70/15/15 mixture used in the released code; we drop *key-points (RDP)* masking because the Ramer–Douglas–Peucker simplification degenerates on the

geometry of sparse visit streams (every visit is "key"). The Adaptive Trajectory Resampling step from the original paper is skipped for the same reason: visits are not over-sampled, so there is nothing to downsample.

Splits, finetune protocol, evaluation. Pre-training and finetuning use the *same* train/test partitions, validation slices, real anomaly labels (NUMOSIM, UA-$*$, LANL), POI vocabularies, and social-edge splits as the rest of the paper (Appendix C.3). At fine-tune time we discard the MAE decoder, take the per-token features from the encoder (CLS dropped), and attach minimal task heads: a single Linear$(d, 1)$ + BCE for anomaly detection on NUMOSIM/UA-$*$ (synthetic perturbations from η); a MLP$(d{\to}d{\to}d)$ query projection scored by dot-product against a learned POI embedding table for next-POI recommendation (sampled-softmax with K=256 negatives plus in-batch negatives); and mean-pooled per-agent embeddings fed to a $|u-v|, u{\odot}v$ pair head with BPR loss for social-link prediction. We deliberately omit the co-occurrence sub-layer, the user-prototype contrastive loss, agent-id embeddings, and the late-fusion handcrafted-feature ensemble used by TraXion: UniTraj has no native mechanism for any of these, and the contributions of the omitted components are separately isolated by C1/C3 in Table 6 (user-prototype) and, for the co-occurrence sub-layer, by ablation row C7 together with the per-seed dispatch analysis of Appendix G.

Architecture and optimizer. The released UniTraj checkpoint uses d=128, which is dramatically smaller than TraXion's d=1040, so a literal off-the-shelf comparison would conflate *objective* with *capacity*. For a capacity-matched comparison we set the encoder hidden size to d=1024 (closest power-of-two to 1040 that the UniTraj 1D-conv tokenizer accepts cleanly) with depth L=6 and H=4 heads, decoder depth 2, RoPE attention — about 100M parameters total, $\sim$5$\times$ larger than TraXion's backbone ($\approx$18M params for the same depth and width: TraXion is parameter-efficient because its factorized attention shares projections across the sequence and co-occurrence axes, and its feature encoder reuses agent/category embeddings that UniTraj has no analog of). Any remaining performance gap therefore cannot be attributed to UniTraj being under-parameterized; if anything, UniTraj is heavily over-parameterized relative to TraXion. The visit window is T=128, larger than TraXion's window of T=32 on the mobility and authentication tasks and equal to TraXion's fine-tuning window on the two ICU corpora.

Pre-training uses AdamW with lr=$2{\times}10^{-4}$, weight decay 10^{-3}, cosine schedule to $\eta_{\min}$=10^{-6}, gradient-norm clipping at 1.0, and batch 64 (128 on NUMOSIM-LA). The per-dataset epoch budget and early-stopping patience are 30/10 for NUMOSIM-LA, 300/80 for UA-Berlin, 200/40 for UA-Atlanta, 200/80 for Gowalla-Stockholm, 500/80 for Gowalla-Austin, and 200/40 for Foursquare-Tokyo, LANL, eICU-CRD, and MIMIC-IV. Early stopping is on validation reconstruction loss.

Fine-tuning uses AdamW with lr=10^{-4}, weight decay 10^{-4} (10^{-3} on the two ICU corpora), cosine schedule, and gradient-norm clipping at 1.0. The per-task, per-dataset epoch budget and patience are 100/10 for NUMOSIM-LA anomaly, 500/50 for UA-$*$ anomaly, 100/15 for next-POI on Foursquare-Tokyo and Gowalla-Stockholm, 200/80 for next-POI on Gowalla-Austin, 100/15 for next-visit, 50/15 for social, and 60/10 for mortality. UniTraj's train batch sizes are 64 for anomaly and next-visit, 8/32/16 for next-POI on Foursquare-Tokyo/Gowalla-Stockholm/Gowalla-Austin, and 32 for social and mortality; the social run uses 200 edge-centric batches per epoch, matching the per-epoch step count of TraXion's social-link recipe (TraXion's own batch sizes are listed in Table 8). Early-stopping criterion: visit/agent AP for anomaly, MRR for next-POI, recall@10 for next-visit, AUC for social, and AUROC for mortality, all computed on the validation slice. On LANL, UniTraj is not fine-tuned: each test event in a length-128 window is masked alone, and the reconstruction error at that position is its anomaly score.

D.2 Human mobility anomaly detection

The four task-specific mobility anomaly detectors (CoBAD, DeepBayesic, USTAD, ICAD) are run from their released code on the cohorts and splits of Appendix C.3. USTAD and ICAD use their published settings unchanged; CoBAD and DeepBayesic were retuned under the procedure of Appendix B.3. ICAD's anomaly score is computed as defined in its paper,

as the joint likelihood of each visit's region, travel time, and departure time (region and travel time on Urban Anomalies, which has no departure times), accumulated in log space.

D.3 Next-POI recommendation

The five next-POI / next-visit baselines (TAPT, GETNext, TrajGPT, MobTCast, LoTNext) are run from their released code on the LBSN subsets and temporal splits of Appendix C.3. All five were retuned under the procedure of Appendix B.3, since their published settings target cohorts other than ours. Mobility-LLM [19] is also run from its released code on the same subsets and splits, with the GPT-2 backbone (the smallest LLM in the upstream enumeration) and inputs shifted by one position so that the autoregressive LLM cannot read the target POI when predicting it. COAST [62] is re-implemented in PyTorch from the paper (the upstream authors do not ship reference code), using the architecture and contrastive recipe of Sections III-B / III-C of the paper. The only adaptation is to make the sequence-head Transformer autoregressive, so the model emits one prediction per non-first position of each 24-hour session with at least two visits; the original COAST predicts only the last held-out check-in per user. All five augmentations (location replacement, time transformation, point masking, data pruning, RDP-based sequence simplification), the Two-Head Self-Attention Encoder over a kNN spatial POI graph (k=10, D_{graph}=2 km), the attention-based feature fusion, the LSTM short-term head, and the InfoNCE objective on sequence-averaged user representations are implemented as in the paper. We tune learning rate $\in \{3\times10^{-4}, 5\times10^{-4}\}$, batch size $\in \{16, 32\}$, and contrastive weight α=0.1 on validation cross-entropy. For the joint next-visit task (§5.2), Mobility-LLM is trained as a single model with both heads under one combined loss — POI cross-entropy plus α times an L1 loss on the next check-in time delta in minutes — so the same checkpoint produces H@k and T@60. The time head is augmented to also take an embedding of the ground-truth next POI as input, matching the TraXion protocol of conditioning the time prediction on the true next location. Because the time loss is on minutes (typical magnitudes in the hundreds) and the POI cross-entropy is in single digits, the joint loss balance is sensitive to α; we picked α=10^{-3} on the Gowalla-Austin validation set after sweeping $\{1, 10^{-2}, 10^{-3}\}$ and choosing the value that maximized validation H@20 without collapsing T@60.

D.4 Social link inference

The four social-link baselines (EBM, USRC, LBSN2Vec, H^{3}GNN) are run from their released code on the same 70/15/15 edge splits used by TraXion (Appendix C.3). All four were retuned under the procedure of Appendix B.3; H^{3}GNN required a per-city architecture, reported in its own paragraph below.

D.4.1 Handcrafted-feature parity ablation

TraXion's social-link recipe in Table 3 concatenates the per-user embedding with 8 handcrafted co-visitation features and late-fuses the test-time logit with a logistic-regression baseline on those same 8 features (Appendix B.2, "Social-link prediction"). The adapted UniTraj baseline in Table 3 does not use these features. To rule out the concern that our advantage over UniTraj on this task derives entirely from this asymmetric input, we run two parity studies. Both use the same 70/15/15 edge splits, the same train/val/test partitions, and the same per-agent embedding pipeline as the rest of the paper.

Parity 1: same handcrafted-feature + LR-fusion adapter on UniTraj. We attach the exact pair head and test-time fusion that TraXion uses to the UniTraj backbone — per-agent mean-pool embedding → symmetric pair scorer over $[\mathbf{e}_u + \mathbf{e}_v,\ |\mathbf{e}_u - \mathbf{e}_v|,\ \mathbf{e}_u \odot \mathbf{e}_v]$ concatenated with the 8 handcrafted features, late-fused at test time with the same logistic-regression baseline at α tuned on validation AUC. The only difference from the TraXion row in Table 3 is the backbone.

Parity 2: no handcrafted features and no late fusion, on either backbone. Both backbones are fine-tuned with a shared adapter that has no access to the 8 handcrafted features and does no late fusion: per-agent embedding → 1-hop graph refinement over the train-edge friend graph → symmetric pair scorer trained with BPR plus a hard-random-negative BCE

term on 256 uniform train-edge / non-edge pairs per step. The graph refinement reads each agent's train-friend neighbors from a per-agent embedding cache (refreshed at the start of each epoch and re-L2-normalized after a learned gated linear projection); it consumes train edges only, the same access H^3GNN, USRC, and LBSN2Vec consume natively.

Single-seed test results are in Table 12.

Table 12: Single-seed parity results on social-link prediction. "HC+LR" = the 8-handcrafted-feature pair head with test-time logistic-regression late fusion (the recipe used by TraXion in Table 3, applied here unchanged to UniTraj as well). "GNN+HN" = the no-handcrafted-feature, no-late-fusion adapter described above.

	Foursquare-Tokyo			Gowalla-Stockholm			Gowalla-Austin		
Method	AP	H@10	MRR	AP	H@10	MRR	AP	H@10	MRR
UniTraj [79] (Table 3 row, no HC, no LR fusion)	.5126	.1570	.0736	.4337	.1273	.0637	.4965	.1549	.0766
UniTraj + HC + LR fusion (parity 1)	.8962	.7241	.5522	.8818	.7945	.6052	.9261	.7278	.4495
UniTraj + GNN + HN (parity 2)	.7684	.3496	.1404	.7236	.3379	.1338	.8167	.4238	.1692
TraXion + GNN + HN (parity 2)	.8208	.6163	.3404	.8572	.6187	.4025	.8645	.5359	.2891
TraXion (Table 3 row, HC + LR fusion)	**.9344**	**.7801**	.5347	**.9276**	.7426	.5870	**.9373**	**.7423**	**.4997**

Reading. (i) Under the no-HC, no-LR-fusion adapter (parity 2), TraXion + GNN + HN outperforms UniTraj + GNN + HN on all 9 cells — a clean backbone-level win that does not depend on handcrafted features or late fusion. (ii) Under the HC+LR-fusion adapter (parity 1), the gap narrows sharply and the outcome is mixed: TraXion leads on 6 of the 9 cells — AP on all three cities and the full Gowalla-Austin column — while UniTraj takes Foursquare-Tokyo MRR and both Gowalla-Stockholm ranking metrics, where its handcrafted pair head reaches H@10 .7945 against our .7426. UniTraj reaches AP $\geq$ .88 on every dataset under this adapter. The combined picture is that the handcrafted co-visitation features carry a large share of the social-link signal regardless of backbone — enough to overturn a backbone deficit on ranking metrics in two of three cities — while at backbone parity (no HC, no LR) the TraXion representation is stronger on every cell.

D.5 Enterprise authentication logs

The LANL anomaly baselines are trained on the same cohort and split as TraXion. Every row of Table 4 is a mean over 5 seeds, except LogGPT and UniTraj†, which use 3. Tuor-AAAI17, DeepLog and LogBERT use AdamW at learning rate 10^{-3}, train batch size 128 (LogBERT 64), window length 64, and early stopping on validation cross-entropy with patience 8 (max 80 epochs). NuLog and LogGPT do not share that recipe: both were retuned under the procedure of Appendix B.3 and selected on validation NLL — NuLog at learning rate 3×10^{-4} with its architecture swept, and LogGPT at 12 layers, learning rate 3×10^{-4}, weight decay 0.01, dropout 0.2 and a 120-epoch budget. Per-event inputs for every baseline are restricted to the same categorical stream TraXion sees — the destination host identifier and the `(auth_type, logon_type)` pair — so no baseline receives side information unavailable to TraXion. All five log-anomaly baselines are trained on LANL as self-supervised density estimators, with early stopping on the same self-supervised objective; TraXion is pre-trained on the same partition and scored from the pre-trained backbone without fine-tuning (Appendix D.6). At test time each baseline scores an event by its model-specific anomaly proxy (next-event NLL for the autoregressive baselines, masked-token cross-entropy for the masked-LM baselines) and we compare those scores directly against the real `redteam.txt.gz` labels at evaluation time only. **Tuor-AAAI17 [56]** is a single-layer LSTM (hidden 64) that takes a `src_user`-id embedding (d_u=16) concatenated with three categorical embeddings (d=32 each for `dst_computer`, `auth_type`, `logon_type`) and predicts the next event's three-way token tuple via independent heads; the event-level anomaly score is the summed cross-entropy across the three fields. **DeepLog [13]** is a two-layer LSTM (hidden 128) over a 1,689-token vocabulary of distinct `dst_computer` identifiers; the anomaly score is the next-token negative log-likelihood. **LogBERT [20]** is a four-layer Transformer encoder (d=128, H=4) with masked-token prediction over the same 1,689-token vocabulary (mask probability 0.15); we score each event by masking it alone and reading the cross-entropy at that position, and omit the hypersphere objective since the original paper shows the MLM term carries most of the signal. For the two autoregressive baselines, the first position

of each non-overlapping window has no preceding context and is excluded from scoring (uniformly across classes); user-level metrics max-pool event scores per `src_user`, identical to the TraXion protocol.

D.6 LANL scoring without fine-tuning: rank fusion of the two pre-training heads

TraXion is not fine-tuned on LANL: the pre-trained backbone is scored directly, so no LANL-specific training objective enters the reported row.

At test time we use both pre-training heads. Each event x owned by user u receives two scores: the sigmoided noise-detection logit $s_{\text{noise}}(x) = \sigma(g_\psi(x))$, and the negative cosine $s_{\text{proto}}(x) = -\cos(\phi(x), p_u)$ between the visit projection $\phi(x)$ and the entity prototype p_u (high = event does not match u's pretraining-time prototype). We convert each to a global rank in $[0, 1]$ across all test events and sum: $s(x) = \text{rank}(s_{\text{noise}}(x)) + \text{rank}(s_{\text{proto}}(x))$. Rank fusion is robust to the different score scales of the two heads and avoids any tunable mixing weight. No real `anomaly` labels are seen at any stage.

D.7 ICU mortality prediction

The four ICU-mortality baselines (STraTS, Raindrop, WaveGNN, SimMTM) are run from their released code on the corpora and patient-level 70/15/15 splits of Appendix C.3. All four were retuned under the procedure of Appendix B.3, since none ships defaults for corpora of this size.

E Variance quantification and cohort sensitivity

E.1 Percentile bootstrap confidence intervals

This appendix reports percentile bootstrap confidence intervals over test instances for the anomaly-detection results of Tables 1 and 4.

All intervals are 95% percentile intervals over 10,000 resamples (1,000 for NUMOSIM-LA). Event- and visit-level metrics resample events or visits; user- and agent-level metrics resample users or agents *after* the scorer's own max-over-events aggregation.

Table 13 gives per-seed intervals for the LANL row of Table 4, where the cohort is thinnest (197 positive events in 145,138, and 38 positive users in 1,559) and the intervals are correspondingly wide. Tuor-AAAI17's event-level AP of .0112 falls *inside* the interval of three of the five seeds, so event-level AP is not separable, and §5.4 claims only same-range parity. The weaker baselines are cleanly excluded — DeepLog (.0066), NuLog (.0059) and LogGPT (.0055) lie outside all five intervals, LogBERT (.0079) and UniTraj† (.0075) outside four of five.

Table 13: Per-seed percentile bootstrap intervals for TraXion's event-level AP on LANL (the pre-train-only row of Table 4), 10,000 resamples over events. The final column is interval width relative to the point estimate.

Seed	Event AP	95% interval	width / point
0	.0106	[.0068, .0185]	1.1×
1	.0118	[.0088, .0173]	0.7×
2	.0123	[.0084, .0189]	0.9×
3	.0157	[.0120, .0220]	0.6×
4	.0361	[.0278, .0477]	0.6×

Table 14 places the two variance sources side by side for the TraXion cells of Table 1. The last column is 2σ across seeds divided by the mean instance-interval width over the five seeds: above 1 means the training seed moves the number more than test-set resampling does, below 1 the reverse. On every cell the instance interval is the wider of the two, by 1.4–6.0×; the widest intervals are at agent level on Urban Anomalies, which has 1,000 agents per city.

Table 14: Across-seed spread (n=5) against the mean instance-resampling interval width for the TraXion cells of Table 1. Ratio is 2σ divided by the interval width.

Dataset	Level	Metric	mean $\pm$ σ	interval width	ratio
NUMOSIM-LA	visit	AP	.1950 $\pm$ .0098	.0279	0.70×
NUMOSIM-LA	visit	Max F1	.2901 $\pm$ .0089	.0352	0.50×
NUMOSIM-LA	agent	AP	.5764 $\pm$ .0155	.0989	0.31×
NUMOSIM-LA	agent	Max F1	.6485 $\pm$ .0114	.0866	0.26×
UA-Berlin	agent	AP	.4132 $\pm$ .0142	.1710	0.17×
UA-Berlin	agent	Max F1	.4261 $\pm$ .0237	.1802	0.26×
UA-Atlanta	agent	AP	.4141 $\pm$ .0231	.1802	0.26×
UA-Atlanta	agent	Max F1	.4214 $\pm$ .0217	.1696	0.26×

E.2 Wall-clock cost

All timings are from the runs behind the reported numbers, on one NVIDIA RTX 6000 Ada (48 GB). Pre-training is once per dataset (once per seed on LANL) and is reused by every downstream task on that dataset; each seed fine-tunes both co-occurrence routes (§5), and the table reports the selected one.

Table 15: Wall-clock time, one RTX 6000 Ada. Pre-train times are each backbone's own accumulated runtime (LANL: mean over its five backbones); fine-tune times are the mean over the reported seeds of the selected route, for the most expensive task on that dataset. Pre-training early-stops, so "epochs" is what ran, not the budget.

Dataset	Pre-train (h)	Epochs run	Fine-tune (h)
NUMOSIM-LA	16.21	36	49.67
UA-Berlin	3.23	586	0.89
UA-Atlanta	2.45	200	2.53
Foursquare-Tokyo	19.01	211	3.99
Gowalla-Stockholm	8.61	399	2.65
Gowalla-Austin	4.52	235	1.11
LANL	0.62	121	—
eICU-CRD	2.46	29	2.80
MIMIC-IV	2.78	80	1.06

LANL has no fine-tune time because TraXion is not fine-tuned there (Appendix D.6). NUMOSIM-LA has the longest end-to-end time, 16.21 h of pre-training and 49.67 h of fine-tuning; every other dataset completes end-to-end in under one day.

E.3 Ablation protocol

The protocol of Table 6 differs from that of the per-task tables of §5 as follows, so the Full TraXion row carries different numbers in the two.

1. Each column family uses one fixed co-occurrence route for all rows: active for mobility anomaly detection, social link, and the authentication log, and bypass for next-visit. The per-task tables select the route per seed on the validation criterion (§5).
2. Cells with $\pm$ report the mean over five pre-training seeds with the fine-tuning seed fixed, except in row C6, which has no pre-training and varies the fine-tuning seed over five values, and for Full TraXion on the authentication log, which is not fine-tuned (item 4). For Full TraXion on mobility anomaly detection and next-visit, $\pm$ is $\sqrt{\sigma_{\text{pre}}^2 + \sigma_{\text{fine}}^2}$, with σ_{fine} taken over three and five fine-tuning seeds, respectively; elsewhere $\pm$ is the standard deviation over the varied seed. Cells without $\pm$ are single runs. The per-task tables vary the fine-tuning seed from one pre-trained checkpoint per dataset, except Table 4, which uses five pre-trained backbones.
3. Social link is reported from the model score alone. Table 3 late-fuses the model score with a logistic regression on handcrafted co-visitation features at a validation-selected mixing weight (Appendix D.4).
4. The authentication-log columns are scored at user level. Full TraXion is scored without fine-tuning from its five pre-trained backbones, as in Table 4 (Appendix D.6); every other row is fine-tuned on LANL, C6 from scratch, so in these columns each arm also differs from Full in fine-tuning.

5. Rows C4 and C5 replace the denoising objective with next-token and masked-token prediction, respectively. For C4 on mobility anomaly detection and the authentication log, and for C5 on the authentication log, the backbone was pre-trained without $\mathcal{L}_{\text{prototype}}$ as well, and so differs from Full TraXion in two components; the other C4 and C5 cells keep $\mathcal{L}_{\text{prototype}}$ and differ in the objective only. On the three arm–task pairs evaluated both ways, keeping $\mathcal{L}_{\text{prototype}}$ changes the variant's score by $-.003$ AP / $-.001$ H@10 (C4, social link), $-.003$ AP / $-.002$ H@10 (C5, social link), and $-.027$ AP / $-.024$ MaxF1 (C5, mobility anomaly detection).

E.4 LANL cohort-construction sensitivity

The LANL cohort of Appendix C.5 fixes a sampling seed and a per-user event cap. Table 16 scores TraXion without fine-tuning, as in Table 4, on two perturbations of that construction: a different benign-user sample at the same cap, and the same sample at twice the cap. Rows 2–4 each use one backbone pre-trained on that cohort with the same recipe; row 1 repeats Table 4. Relative to row 2, doubling the cap lowers all six metrics, and re-sampling the benign users lowers the three event-level metrics and user Max F1 while raising user AP and AUROC. Every value in rows 2–4 lies within two standard deviations of the mean in Table 4, except user Max F1 at twice the cap (.2826, 3.2 standard deviations below).

Table 16: LANL cohort-construction sensitivity, TraXion scored without fine-tuning. Row 1 is Table 4 (mean ± standard deviation over five backbones); rows 2–4 use one backbone each.

Sample seed / cap	Event AP	Event AUROC	Event Max F1	User AP	User AUROC	User Max F1
42 / 500 (Table 4)	$.0173_{\pm.011}$	$.8594_{\pm.052}$	$.0683_{\pm.030}$	$.2392_{\pm.039}$	$.8611_{\pm.033}$	$.4151_{\pm.042}$
42 / 500	.0264	.9235	.0858	.2057	.8668	.3617
202 / 500	.0089	.8543	.0367	.2727	.9146	.3385
42 / 1000	.0036	.8233	.0224	.1730	.8323	.2826

E.5 Capacity-matched next-POI baselines

To separate the prototype mechanism from user-embedding capacity, we set the user-embedding dimensionality of LoTNext and MobTCast to $\dim(p_u)$=208 on Gowalla-Austin, with three runs per method and a separately pre-trained TraXion backbone. TraXion reaches H@10 $.2579_{\pm.0026}$ against LoTNext $.1951_{\pm.0059}$ and MobTCast $.1595_{\pm.0048}$. Replacing the InfoNCE of Eq. 3 with a full 5,397-way user-classification cross-entropy at the same capacity gives $.2502_{\pm.0023}$.

F Extended related work

Each downstream task in §5 has a bespoke baseline lineage. Table 17 compares each baseline against TraXion along three axes: (i) *input* — what the model consumes per training example; (ii) *supervision* — the loss it trains under; (iii) *cross-entity coupling* — whether and how information from other entities enters the model. TraXion consumes the per-event tuple (u, x, τ, a, δ) of §3, supervises through the dual denoising-plus-entity-prototype objective of §4, and routes cross-entity signal through the co-occurrence attention axis (§4.4) and the shared prototype table (§4.3).

CoBAD [59] is the closest baseline in spirit on the cross-entity axis, but its coupling enters through a per-event collective-behavior likelihood rather than a pre-training objective, so the cross-entity signal cannot be reused across downstream tasks. GETNext [68] routes cross-entity signal through a global trajectory flow map computed once from the training set, whereas TraXion's co-occurrence axis recomputes attention weights per visit from the current event window. THP and A-NHP, the temporal-point-process baselines for next-visit prediction, model continuous-time event streams via attention-driven marked Hawkes intensities (with auxiliary mark-CE and time-MSE heads in THP) but operate per-user; TraXion shares the same prediction target (next-event time and mark) yet learns its representations under a discriminative denoising objective rather than an event-likelihood loss, which the next-visit results show transfers to high H@k. In the social-link family, EBM, LBSN2Vec, H^{3}GNN, and USRC all consume cross-entity structure as the central

Table 17: Per-baseline comparison along three axes. Family blocks group baselines benchmarked under each downstream task in §5; the bottom row summarizes TraXion for reference.

Method	Input	Supervision	Cross-entity coupling
Mobility anomaly detectors			
ICAD [3]	per-user visit sequence	SSL autoregressive next-event; anomaly from likelihood	none (per-user)
CoBAD [59]	per-user visits + collective context	likelihood under a collective-behavior model	explicit, per-event likelihood
USTAD [60]	per-user spatio-temporal visits	supervised prediction with an uncertainty head	none (per-user)
DeepBayesic [16]	per-user visits + population density	statistical-likelihood + neural prediction	shared density prior
Next-POI rankers			
TAPT [64]	per-user visit history	joint next-POI + next-timestamp CE	none
MobTCast [65]	per-user visit history	next-POI CE + auxiliary trajectory forecasting	none
LoTNext [63]	per-user visit history	long-tail-aware re-weighted next-POI CE	none
GETNext [68]	per-user history + global flow map	next-POI CE over flow-map-augmented representation	pre-committed flow graph
COAST [62]	per-user check-ins + spatial POI graph	next-POI CE + InfoNCE over augmented sequence pairs	spatial kNN POI graph (GAT)
Next-visit predictors (joint location + time)			
TrajGPT [25]	per-user visit history	autoregressive POI CE + GMM NLL on next-visit time delta	none (per-user)
Mobility-LLM [19]	per-user visit history wrapped as LLM prompt	POI CE + L1 on next-visit time delta (min), joint-α weighted	none (per-user)
THP [80]	per-user marked event stream	marked-Hawkes log-likelihood + next-mark CE + scaled time-MSE	none (per-user)
A-NHP [66]	per-user marked event stream	attentive neural Hawkes log-likelihood (single term)	none (per-user)
Social-link inferers			
EBM [46]	co-visitation tuples	closed-form entropy-based pair score (no learned parameters)	central observable
LBSN2Vec [67]	check-ins as (u, t, x, a) hyperedges	hypergraph-embedding loss with social-edge supervision	hyperedge structure
H^{3}GNN [34]	user-POI heterogeneous hypergraph (hyperbolic)	friend-recommendation supervision over the hypergraph	hypergraph topology
USRC [9]	per-user mobility + pair features	cross-city social-relationship classification	pair-level coupling
Enterprise-log anomaly detectors			
DeepLog [13]	per-host log-key sequence	LSTM next-token CE; anomaly from likelihood	none (single-stream)
Tuor-AAAI17 [56]	per-user multi-field log sequence	summed CE across categorical fields	none (single-stream)
LogBERT [20]	per-host log-key sequence	masked-token reconstruction (+ hypersphere)	none (single-stream)
NuLog [43]	per-host log-token sequence	masked-token reconstruction with [CLS]-pooled anomaly head	none (single-stream)
LogGPT [22]	per-host log-key sequence	GPT autoregressive next-token CE; anomaly from likelihood	none (single-stream)
ICU mortality predictors			
STraTS [55]	per-stay irregular event triples	self-supervised forecasting pre-train + supervised binary CE	none (per-stay)
Raindrop [73]	per-stay sensor multivariate time series	dependency-graph message passing, supervised binary CE	per-stay sensor graph
WaveGNN [21]	per-stay irregular multivariate time series	wavelet-based GNN, supervised binary CE	per-stay sensor graph
SimMTM [12]	per-stay dense $[C,T]$ time-series grid	masked-reconstruction + series-wise contrastive pre-train + supervised binary CE	none (per-stay)
TraXion	per-event tuple (u, x, τ, a, δ)	SSL pre-train (denoising + entity-prototype) + task-specific fine-tune (self-supervised on anomaly tasks; supervised on next-POI / social / ICU mortality)	learned co-occurrence attention + shared prototype

training signal; TraXion instead learns its per-user prototypes during pre-training without any social-edge supervision, and the social-link head in §5.3 is a thin pair scorer over those prototypes rather than a graph network. LogBERT [20] is the closest baseline to TraXion's pre-train-plus-fine-tune paradigm; its single-stream restriction is the structural gap, since cross-entity coupling between users sharing hosts is an explicit signal on LANL that single-stream models cannot use.

Prototype-contrastive learning beyond mobility. Prototype-anchored contrastive losses appear elsewhere with a different supervisory commitment. PCL [31] alternates clustering of unsupervised image embeddings with a contrastive loss whose positive class is the cluster centroid; SWCC [18] augments InfoNCE on event sentences with a clustering branch whose centroids regularize semantically related events from being pulled apart by the contrastive negatives. In both, the prototype is a *latent* cluster centroid discovered from the data, and the assignment of an example to a prototype is itself an estimand. TraXion's p_u is instead supervised by entity identity — the assignment is given for free in every MESES dataset (§3, A2) — and the same vector enters the input side of f_θ, so contrastive shaping at the output and feature use at the input are coupled by construction. On the place side of the (entity, context) pair, Siampou et al. [52] learn POI-side prototypes from mobility data, the dual problem to ours.

Summary. Across all six families, TraXion differs in two consistent ways: (i) supervision decomposes into a self-supervised pre-train (denoising plus entity-prototype) and a task-specific fine-tune — self-supervised on the anomaly tasks (synthetic perturbations of η at fine-tune; the LANL row skips the fine-tune and is scored from the pre-trained backbone, Appendix D.6) and supervised on the labeled tasks (next-POI, social link, ICU mortality) — rather than a single end-to-end supervised stage; (ii) cross-entity coupling enters through learned attention and a shared prototype table rather than a pre-committed graph topology, a hand-built collective-behavior likelihood, or single-stream per-entity processing.

G Limitations and outlook

Discriminative vs. likelihood-based supervision. TraXion pre-trains representations for classification and ranking; on next-visit time prediction (Table 2) it tops H@10/H@20 across all three LBSN city subsets but trails the LLM-based Mobility-LLM [19] and the marked-Hawkes A-NHP [66] on T±60 — the metric that rewards a calibrated density over event times. For next-event time, hazard estimation, or any task whose primitive is a likelihood over the event distribution, TPP-style supervision remains the natural fit; a hybrid that adds a likelihood head on top of the TraXion backbone is a clean direction for extending the framework to those tasks without losing the discriminative gains documented here.

Corruption operator is intentionally simple. The corruption operator η resamples each flagged event's context $x_{u,t}$ uniformly from the bounding box of $\phi(\mathcal{X})$ (snapped to the nearest $x' \in \mathcal{X}$, with the activity $a_{u,t}$ replaced by the activity associated with x') and resamples its timestamp $\tau_{u,t}$ uniformly within the window between adjacent events (§4.2); the activity $a_{u,t}$ thus shifts as a side-effect of the context-snap, and the duration $\delta_{u,t}$ shifts as a side-effect of the gap-preservation step on adjacent events. This is a deliberately MESES-universal choice that requires only the (context, time) pair guaranteed by every MESES instance (§3); nonetheless it leaves A1's joint plausibility only partially constrained, since a context-conditional resampler — e.g., time-of-day-aware context priors, near-but-atypical contexts as hard negatives, or marked corruption of the activity and duration channels — would more tightly target the joint density over (x, τ, a, δ) and is the most direct lever we are aware of for sharpening A1 alignment. We did not run a corruption-design ablation along this axis, and view a systematic study of harder negatives — including their interaction with $\mathcal{L}_{\text{prototype}}$ and the co-occurrence sub-layer — as future work.

Co-occurrence route selected on validation, not learned. The co-occurrence route is chosen per seed on the validation criterion (§5). It is active on every seed of NUMOSIM-LA anomaly detection and of social-link prediction, and bypassed on every seed of next-POI and next-visit prediction; it is bypassed on 4 of 5 seeds for UA-Atlanta and MIMIC-IV, 3 of 5 for eICU-CRD, and 2 of 5 for UA-Berlin. Learning the choice end-to-end, for example as a gate with a per-task or per-event coefficient, would remove the validation step.

Per-instance pre-training, not yet cross-instance. We pre-train one checkpoint per dataset — six mobility datasets, the LANL authentication log, and the two EHR corpora — which validates that the same recipe transfers across MESES instances but does not yet exploit the obvious next gain: a single backbone trained on a unified corpus of MESES instances. Characterizing when a shared substrate is rich enough for cross-instance transfer (mobility-only, mobility+auth, mobility+EHR), and identifying whether the prototype table or the substrate embedding is the cross-instance bottleneck, are the directions we view as most likely to extend the framework beyond the per-instance checkpoints studied here. A related deployment-time limitation is cold start on entities unseen during pre-training: the prototype table is keyed by training-set users, so a freshly observed user must either inherit a generic initialization or accumulate enough events to be fine-tuned in, and we have not characterized the few-shot regime where this matters most.

Single-run cells in the ablation table. In Table 6, 30 of the 64 cells are means over five seeds (Appendix E.3); the other 34 are single runs, carry no error bar, and are not tested for significance.

Scope of the clinical evaluation. Table 5 reports eICU-CRD (197,307 stays) and MIMIC-IV (94,407 stays); both are retrospective US intensive-care databases, and both are scored on the same binary outcome, in-hospital mortality. We therefore do not test whether the recipe transfers to non-ICU care, to non-US health systems, to longitudinal outcomes, or across sites within a corpus: our splits are patient-level but not site-held-out, so a hospital seen in training may also appear at test.

Substrate embedding ϕ is fixed, not learned. For the two non-mobility instances we obtain ϕ by running 2D UMAP on the user–host (LANL) and context–stay (eICU) co-occurrence matrix at fixed hyperparameters ($n_{\text{neighbors}}$=30, cosine distance; Appendices C.5, C.6) and freezing the result before pre-training. The framework treats ϕ as part of the schema (§3), so any unsupervised geometry over $\mathcal{X}$ that respects co-occurrence would satisfy the formal requirements; we did not sweep $n_{\text{neighbors}}$, distance metric, or the embedding dimensionality, and we did not test an end-to-end learned ϕ that is updated jointly with the encoder under $\mathcal{L}$ (Eq. 1). A learned ϕ would close the loop between the substrate geometry that η's context perturbation samples from (§4.2) and the representations the encoder builds on top of it, and is the most direct extension for the non-mobility instances.

Evidence for the identity-shortcut constraint. The shared-prototype design relies on $\mathcal{L}_{\text{noise}}$ to prevent an identity shortcut from the input-side p_u to its supervisory copy (Appendix B.1.3); ablation row C2 (Table 6) shows the constraint is empirically binding, but the evidence is behavioral rather than structural. We did probe the representations directly — linear probes recovering the entity identity from each layer's $\mathcal{H}_{u,t}$, with frozen prototypes and layer by layer, on NUMOSIM-LA and UA-Berlin — and found *no* evidence that the prototype embeddings encode behavioral structure beyond identity. What remains untried is an *architectural* barrier: masking the residual on the entity-token path for selected layers, or a stop-gradient on the p_u pathway past a chosen depth, would block the trivial copy operationally rather than by training pressure, and an ablation that toggles that barrier would distinguish "the loss happens to suppress the copy" from "the architecture forbids it."

H Broader impacts

Positive impacts. TraXion consolidates anomaly detection, next-visit prediction, social-link inference, and clinical mortality prediction under a single self-supervised pre-training framework, reducing the engineering cost of building task-specific models for sparse multi-entity event streams. On the mobility side, better next-visit and next-POI prediction supports downstream applications such as demand forecasting for transit planning, congestion aware routing, and location-based recommendations that surface relevant places to users without requiring them to specify their intent in advance. Label-free anomaly detection also recovers lateral-movement-shaped activity in enterprise authentication logs (§5.4), a useful capability for organizations that can deploy it on their own logs without requiring labelled red-team events. On the clinical side, the AP and Sens@Sp.9 gains on the rare-positive eICU mortality task (§5.5) land at the operating point most relevant to early-warning use, where the cost of a missed positive dominates.

Negative impacts. TraXion improves pre-trained representations for sparse multi-entity event streams, and like any improvement in general-purpose representations, its gains are not intrinsically tied to any specific use. The Axiom A2 and A3 mechanisms that make the framework accurate (entity-distinguishing information, co-occurrence structure) are also the mechanisms that any downstream task exploiting entity- or population-level structure would benefit from, including uses beyond those we evaluated. As with other pre-training advances, practitioners deploying TraXion on new data should apply the oversight standards appropriate to that setting.

Mitigations. All experiments use datasets publicly released by their original owners under their own anonymization and licensing decisions (Appendix C.1); we introduce no new identifiers and redistribute no raw data. Practitioners deploying TraXion on new data should validate its behavior on that data before use and apply the access controls that govern the underlying records to the model's per-entity outputs.

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 abstract and Section 1 state three contributions: formalizing MESES via three axioms, introducing TraXion, and empirically evaluating it across mobility and non-mobility MESES instances. These map directly to the axiom definitions in Section 3.2, the framework in Section 4, and the benchmarks in Section 5, which substantiate every quantitative gain claimed in the abstract (next-POI, next-visit, social link, anomaly detection, LANL authentication, eICU mortality).

   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 "Limitations" paragraph in Section 6 states three limitations: TraXion is discriminative and trails TPP/LLM baselines on likelihood-style next-visit timing (T±60), the co-occurrence route (active or bypass) is selected per seed on validation rather than learned end-to-end, and we still pre-train one checkpoint per dataset rather than a single shared backbone across MESES instances. Appendix G expands each point and discusses scaling TraXion to a unified MESES corpus as the most likely extension.

   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: [N/A]

   Justification: The paper contains no theorems, lemmas, or proofs. The three axioms in Section 3.2 are stated as design desiderata that any MESES pre-training framework should satisfy, not as theoretical claims with formal proofs; the paper's contributions are a data-class formalization and an empirically validated framework.

   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: Section 4 describes the TraXion framework in full, and Appendix A gives formal definitions of the per-event embeddings, co-occurrence retrieval, factorized attention block, pre-training heads and perturbation operator, and total

objective; Appendix B reports pre-training mechanics and per-dataset hyperparameters, and Appendix C documents dataset statistics, splits, the LANL cohort construction, and the EHR event-stream construction (including the UMAP-derived context coordinates). Baseline implementations (including the UniTraj adaptation and the LANL rank-fusion scoring recipe) are detailed in Appendix D, and the protocol paragraph of Section 5 fixes the seed count, hardware, and significance test.

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: [Yes]

   Justification: All nine datasets used in the experiments are publicly available: NUMOSIM-LA [53], Urban Anomalies-Berlin/-Atlanta [1], Foursquare-Tokyo [67], Gowalla-Stockholm and Gowalla-Austin [8], the LANL unified-host-and-network authentication log [30], and the PhysioNet eICU-CRD [47] and MIMIC-IV [26] critical-care databases. The two EHR corpora require a credentialed PhysioNet account and a signed data use agreement; the other seven are open downloads. The code is released at `https://github.com/ktxlh/TraXion` with a README giving environment setup, data preparation, and the exact commands needed to reproduce the pre-training, fine-tuning, and evaluation runs reported in Section 5; dataset access and split construction are documented in Appendix C.

   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: Section 5 fixes the protocol (datasets, seeds, hardware, significance test); Appendix B.2 reports the optimizer (AdamW), shared architecture and loss defaults, per-dataset pre-training settings (Table 7), and per-task fine-tune budgets (Table 8); Appendix C.3 documents dataset splits. Appendix B.3 describes how hyperparameters were chosen — a validation-guided iterative search orchestrated by a Claude Code agent, applied uniformly to TraXion and every retuned baseline, with the verbatim agent prompt shipped with the released code.

   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: Tables 1–5 report mean $\pm$ 1 standard deviation (sample std) over 5 random seeds for TraXion and for every baseline that leads a column; the remaining baselines are at 3–5 seeds, and EBM is deterministic by construction (Section 5, "Implementation and reporting"). The reported variability captures seed-to-seed variation under the same data splits and hyperparameters; significance against the strongest per-column baseline is assessed by a one-sided Wilcoxon rank-sum (Mann–Whitney U) test on the per-seed scores at $p \leq 0.05$ and is marked with $*$ in the tables.

   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: The protocol paragraph in Section 5 states the compute worker (one NVIDIA RTX 6000 Ada, 48 GB) and the per-experiment wall-clock budget (each end-to-end pre-training plus per-task fine-tuning completes within 5 days on a single GPU); the iterative hyperparameter search described in Appendix B.3 implies additional compute beyond the reported final runs.

   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 research uses only publicly released, de-identified or simulated datasets — NUMOSIM-LA and Urban Anomalies-Berlin/-Atlanta are agent-based simulations with no real-person identifiers, Foursquare-Tokyo and Gowalla-Stockholm/-Austin are widely used anonymized LBSN releases, the LANL unified-host-and-network log is published with anonymized user/host identifiers, and the PhysioNet eICU-CRD and MIMIC-IV databases are HIPAA-compliant de-identified critical-care data accessed under signed data use agreements — and we use each dataset under its published license; no new data is collected, no crowdsourcing or human-subjects interaction is involved, and the paper contains no application targeting weapons, surveillance, or discriminatory decision-making. Uses beyond those we evaluate are discussed in the broader-impacts response below.

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: Appendix H discusses positive impacts (a single pre-training framework in place of task-specific models; mobility applications such as transit demand forecasting, congestion-aware routing, and location recommendations; label-free anomaly detection on enterprise authentication logs) and negative impacts (the entity- and co-occurrence mechanisms of Axioms A2 and A3 would equally benefit downstream uses beyond those we evaluate), and lists mitigations: only publicly released datasets are used under their owners' anonymization and licensing, no new identifiers are introduced and no raw data is redistributed, and deployments should validate the model on their own data and apply the access controls of the underlying records to its per-entity outputs.

    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: [N/A]

    Justification: The paper releases no pre-trained language models, image generators, or scraped datasets; the released code is a method implementation, and all datasets used are pre-anonymized public releases consumed under their existing licenses

(Appendix C.1). Deployment-time considerations for the per-entity prototype table on sensitive cohorts are discussed as part of the broader-impact mitigations in Appendix H.

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: Every dataset and baseline implementation used in the paper is cited in the bibliography, and Appendix C.2 (Table 10 plus the surrounding paragraphs) lists each dataset's source repository and license — LANL is CC0, eICU-CRD and MIMIC-IV are under the PhysioNet Credentialed Health Data License, Urban Anomalies-Berlin/-Atlanta is CC-BY 4.0, Foursquare-Tokyo and the two Gowalla city subsets are released for research with a citation requirement, and NUMOSIM-LA is publicly released by the authors with no formally named license — and itemizes the upstream license of each baseline repository (Apache 2.0 for UniTraj; MIT for LoTNext, TAPT, THP, A-NHP, TrajGPT, Raindrop, WaveGNN, and STraTS; BSD 3-Clause for NuLog; CC-BY-NC-4.0 for LogGPT) along with the repositories that ship no explicit license (GETNext, MobTCast, Mobility-LLM, EBM, USRC, LBSN2Vec, H^3GNN, CoBAD, DeepBayesic, USTAD, ICAD, SimMTM) and the baselines we re-implemented from their papers (COAST, Tuor-AAAI17, DeepLog, and LogBERT). We use each asset only as released, do not redistribute any raw dataset, and consume each baseline repository under its upstream license.

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 paper releases the implementation of TraXion as a new code asset at `https://github.com/ktxlh/TraXion`. The repository ships with a `README` documenting the environment setup, data preparation steps, and the exact commands needed to reproduce the pre-training, fine-tuning, and evaluation runs reported in Section 5, and a `LICENSE` file (MIT) under which the code is released. No new datasets or pre-trained model checkpoints are released; all datasets used are pre-existing public releases (Appendix C.2).

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: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects

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: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects

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: The paper does not use LLMs as any important, original, or non-standard component of the core method development. We use a Claude Code agent to orchestrate the hyperparameter search (Appendix B.3); this is a tuning procedure, not a component of the method, and is described for transparency. We also use Claude Code as a coding assistant for routine engineering tasks: dispatching training and evaluation runs against existing scripts as shared GPUs become available, and implementing common baselines (e.g., EBM) when no official reference implementation is available.

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.