HYPERSPECTRAL TRAJECTORY IMAGE FOR MULTI-MONTH TRAJECTORY ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Trajectory anomaly detection underpins applications from fraud detection to urban mobility analysis. Dense GPS preserves fine-grained evidence such as abnormal speeds and short-duration events, but its quadratic cost makes multi-month analysis intractable; sparse stay-point methods scale by discarding that evidence and require a separate modeling regime. We argue that this bottleneck is unnecessary: dense and sparse trajectories share a natural two-dimensional cyclic structure along within-day and across-day axes. TITAnD (Trajectory Image Transformer for Anomaly Detection) is the first framework to cast both in a single representation, a Hyperspectral Trajectory Image (HTI), a day $\times$ time-of-day grid whose channels encode spatial, semantic, temporal, and kinematic information. Under this formulation, agent-level detection reduces to image classification and temporal localization to semantic segmentation. The Cyclic Factorized Transformer (CFT) models the two temporal axes directly, reducing attention cost by up to two orders of magnitude. With the HTI, multi-month dense anomaly detection becomes feasible for the first time, and CFT keeps it accurate and fast. Empirically, TITAnD matches or beats every evaluated baseline in AUC-PR on all four benchmarks. It surpasses vision models such as U-Net while using as few as 6.5M parameters, and it runs 11–75$\times$ faster than a capacity-matched flat Transformer on the same trajectory images.

1 INTRODUCTION

Trajectory anomaly detection underpins applications from fraud detection and public safety to urban mobility analysis, where the goal is to identify rare or suspicious behaviors in an individual's movement history and localize when they occur (Kennedy & Züfle, 2024; Azarijoo et al., 2025). Existing methods fall into two largely disconnected regimes. Sparse approaches compress dense GPS streams into event sequences such as stay-points or visits (Zheng et al., 2009; Wen et al., 2024; Azarijoo et al., 2025; Hosseini Sereshgi et al., 2025), enabling long-horizon modeling but discarding short-duration activities and in-motion behaviors such as detours, abnormal speed profiles, and atypical routes (Kennedy & Züfle, 2024; Shao et al., 2025). Dense approaches preserve this fine-grained evidence but face a long-context bottleneck: even after discretization, multi-month trajectories yield thousands of time steps, and Transformers scale quadratically with sequence length (Vaswani et al., 2017; Liang et al., 2022). To our knowledge, no existing method performs anomaly detection over multi-month dense GPS trajectories. Recent trajectory foundation models improve scalability through adaptive resampling or efficient encoders (Zhu et al., 2024), yet they still operate on flat one-dimensional sequences and do not target anomaly detection. Likewise, prior work does not bridge dense and sparse regimes within one representation and architecture, and most existing approaches remain self-supervised, relying on reconstruction or next-event prediction as proxy objectives (Wen et al., 2024; Azarijoo et al., 2025; Mbuya et al., 2024). When labeled anomaly data is available, as in recent simulation benchmarks (Stanford et al., 2024), supervised detection remains largely unexplored.

Beyond the tension between observability and scalability, both paradigms share a deeper limitation: trajectories are modeled as flat one-dimensional sequences, ignoring the cyclic temporal structure governing human mobility (Schneider et al., 2013). Within a day, movement unfolds as coherent episodes such as commutes, errands, or leisure, each characterized by consistent motion patterns and stop-and-go dynamics. Across days, behavior at the same time-of-day tends to follow stable

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

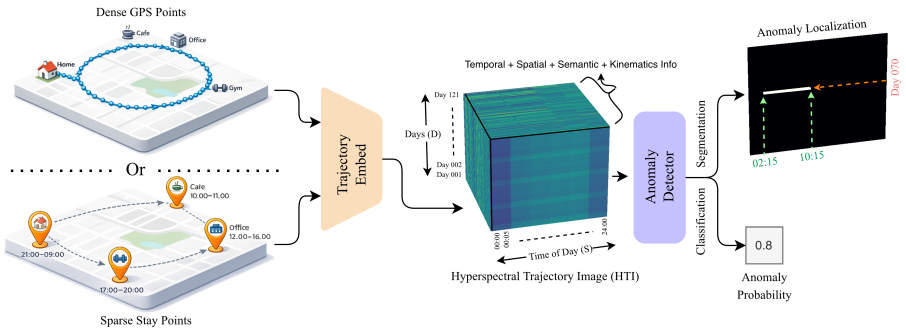


Figure 1: Overview of our TITAnD framework. Dense GPS streams or sparse stay-points are encoded into a unified day $\times$ time-of-day hyperspectral trajectory image (HTI), where each pixel represents spatio-semantic, temporal, and kinematic information.

routines, such as recurring morning commutes or evening gym visits (Herder & Siehndel, 2012; Schneider et al., 2013). Anomalies, in turn, manifest as deviations along these two axes: an activity may be atypical for the time of day at which it occurs, or a behavior that recurs across days may be skipped or shifted. Collapsing these temporal axes into a single dimension obscures this structure and forces models to rediscover periodicity from raw positional tokens.

This cyclic structure suggests a different formulation. If human mobility organizes along within-day and across-day axes, then trajectories over weeks or months naturally form an *image*: a day $\times$ time-of-day grid where each pixel encodes behavior within a fixed time window (Fig. 1). Under this formulation, agent-level anomaly detection reduces to image classification and temporal localization to semantic segmentation. Both dense GPS streams and sparse stay-point logs can be projected into the same image space through data-specific encoders that aggregate irregular observations into per-pixel embeddings, revealing that dense and sparse trajectory anomaly detection are instances of the same vision task. We instantiate this idea in **TITAnD** (**T**rajectory **I**mage **T**ransformer for **An**omaly **D**etection), an end-to-end supervised framework. Our contributions are as follows:

- We present TITAnD, to our knowledge the first framework to perform anomaly detection over multi-month dense GPS trajectories. This is enabled by two complementary ideas: a Hyperspectral Trajectory Image (HTI) that compresses GPS observations into a structured 2D representation, and a Cyclic Factorized Transformer (CFT) that processes it efficiently.
- We unify dense and sparse trajectory data under a single architecture by reformulating anomaly detection as a vision problem with the HTI representation: a day $\times$ time-of-day grid whose channels encode spatial, semantic, temporal, and kinematic information aggregated from either dense GPS streams or sparse stay-points through modality-specific encoders.
- We introduce the Cyclic Factorized Transformer (CFT), which factorizes attention along the within-day and across-day axes of the HTI to exploit the cyclic pattern of human routines while reducing the cost significantly.
- Extensive experiments across dense and sparse benchmarks show that trajectory images alone do not guarantee improved performance: gains arise when the backbone explicitly exploits their cyclic structure.

2 Related Work

2.1 Trajectory Anomaly Detection

Trajectory anomaly detection seeks to identify rare or suspicious behaviors in an individual's movement history. Existing approaches fall into two disconnected regimes by trajectory representation.

A common strategy is to compress dense GPS streams into visits or stay-points (Zheng et al., 2009). These representations scale well but discard short-duration and in-motion behavior. ICAD predicts visit sequences autoregressively (Azarijoo et al., 2025), USTAD reconstructs masked stay-point events (Wen et al., 2024), and LM-TAD applies causal language modeling to user-conditioned trajectory tokens (Mbuya et al., 2024). Behavior-aware methods compare an individual's activity against

their own normal profile (Xie et al., 2025; Azarijoo et al., 2026), and CoBAD looks for anomalies in the joint behavior of related individuals (Wen et al., 2025). These methods operate on sparse sequences and are typically self-supervised; we study supervised detection and evaluate all baselines on the same data, labels, and metrics.

When operating directly on dense GPS coordinates, existing work focuses on short horizons such as individual trips. FOTraj (Shao et al., 2025) detects route anomalies in taxi trajectories using a spatio-temporal graph, while earlier methods such as iBAT (Zhang et al., 2011) identify deviations within individual rides. More broadly, recent trajectory architectures remain limited to short sequences: TrajFormer (Liang et al., 2022) compresses attention for trip-level classification, TrajMamba (Liu et al., 2025b) applies linear-time state-space models to vehicle trips, and UniTraj (Zhu et al., 2024) pre-trains across large datasets but processes each trajectory as a flat one-dimensional sequence. Scaling these models to months-long histories would incur prohibitive quadratic cost or require local attention strategies that sacrifice cross-day context. Consequently, anomaly detection over dense, multi-month trajectories remains an open problem.

2.2 From Cyclic Mobility Structure to Image Representations

Human mobility exhibits strong regularity along two temporal dimensions. Within a day, movement follows circadian patterns with predictable episodes such as commutes, errands, and leisure (Song et al., 2010; Herder & Siehndel, 2012). Across days, individuals repeatedly visit a small set of locations and follow stable routines over months (González et al., 2008; Schneider et al., 2013). These regularities naturally define a two-dimensional temporal structure with days and time-of-day as separate axes, suggesting that long-horizon trajectories can be organized as a day $\times$ time-of-day grid rather than a flat sequence, as in eigenbehaviors (Eagle & Pentland, 2009).

The idea of converting temporal signals into images has been explored in time-series analysis. Classical transforms such as Gramian Angular Fields, recurrence plots, and Markov transition fields encode temporal relationships as images for CNN-based classification (Wang & Oates, 2015; Eckmann et al., 1987; Hatami et al., 2018). Other approaches render time series as visual representations processed by vision transformers (Li et al., 2023). These methods treat the image primarily as a visualization of a one-dimensional signal rather than a structured representation with semantically meaningful axes. Closer to our setting, TimesNet (Wu et al., 2023) and the later TimeMixer++ (Wang et al., 2024) reshape a time series along periods detected in the signal, using the two-dimensional view as a feature transformation within a general-purpose time-series backbone. In our work, the image is instead the formulation of the problem itself: it makes multi-month modeling efficient and organizes behavior along within-day and across-day axes so that our architecture can exploit the inherent two-dimensional structure of human mobility. Closest to us, Hosseini Sereshgi et al. (2025) draw stay-points as a calendar image, color each visit by venue type, shade it by distance from the agent's usual places, and ask a vision-language model to spot anomalous agents without fine-tuning. Their image has no map location, nearby POIs, or motion, and it cannot hold dense GPS. In the HTI, every pixel carries learned spatial, semantic, temporal, and kinematic features for both dense GPS and stay-points, and CFT learns from it end to end.

3 Method

3.1 Overview and Task Definition

TITAnD accepts two input modalities. A *dense* trajectory is a time-ordered GPS stream $\mathcal{R}^{\text{dense}} = \{(\text{lat}_n, \text{lon}_n, t_n)\}_{n=1}^{N}$ sampled at approximately 10-second resolution. A *sparse* trajectory is a sequence of stay-point events $\mathcal{R}^{\text{sparse}} = \{(t_k^{\text{start}}, t_k^{\text{end}}, \text{lat}_k, \text{lon}_k)\}_{k=1}^{K}$, each recording the arrival time, departure time, and location of a stationary episode. Both modalities cover multi-month observation windows with benchmark-specific past/future boundaries; PoL uses a 450-day history followed by a 14-day evaluation window.

A data-specific encoder (Sec. 3.2) projects either input into a *Hyperspectral Trajectory Image* (HTI), a $D \times S \times C$ tensor where D is the number of days, S is the number of time-of-day slots (288 five-minute slots for stay-point inputs and 96 fifteen-minute slots for dense GPS), and C is the number of learned feature channels that encode temporal, spatial, semantic and kinematics information.

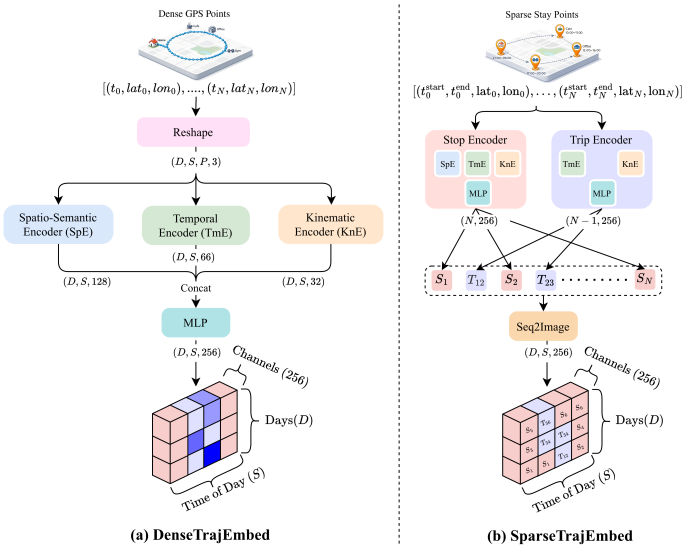


Figure 2: TITAnD's two data-specific encoders. **(a) DenseTrajEmbed** fuses per-slot spatio-semantic (SpE), temporal (TmE), and kinematic (KnE) features from a GPS stream. **(b) SparseTrajEmbed** encodes stops with spatial, temporal, and kinematic features and trips with temporal and kinematic features, and Seq2Image maps each event onto its (d, s) cells. Both produce the HTI consumed by CFT.

Slot-level ground truth $y_{d,s} \in \{0, 1\}$ is constructed from whichever label granularity is available. For dense trajectories with per-point annotations $\tilde{y}_n$, $y_{d,s} = \mathbb{I}\big[\max_{n \in \mathcal{R}^{\text{dense}}_{d,s}} \tilde{y}_n{=}1\big]$, and for sparse trajectories with per-stop labels $\tilde{y}_k$, $y_{d,s} = \mathbb{I}\big[\exists\, k : \tilde{y}_k{=}1 \wedge [t_k^{\text{start}}, t_k^{\text{end}}) \cap \text{slot}(d, s) \neq \varnothing\big]$. Both constructions produce the same binary label grid $\mathbf{Y} \in \{0, 1\}^{D \times S}$, enabling a unified loss regardless of input modality.

3.2 Trajectory Encoders

TITAnD provides two data-specific encoders—DenseTrajEmbed for dense GPS streams and SparseTrajEmbed for sparse stay-point logs—that each produce an HTI $\mathbf{X} \in \mathbb{R}^{D \times S \times C}$, where C denotes the number of channels ($C{=}256$ in our experiments). Both encoders share three feature groups—spatio-semantic, temporal, and kinematic—but instantiate them differently for their respective input modality. The resulting HTI is consumed identically by CFT (Sec. 3.3) regardless of data source.

3.2.1 Shared Components.

Spatio-semantic cell embedding. We first project GPS coordinates to a planar system (UTM) and assign each location to a grid cell. We use an adaptive QuadTree (Samet, 1984) discretization that refines regions with high POI density, improving robustness to GPS jitter while keeping the vocabulary compact. Each location is mapped to a cell id c and a discrete grid coordinate (i, j).

To inject lightweight semantics, each cell c is associated with a set of nearby POIs $\mathcal{P}(c)$ (e.g. restaurant, office). It is capped to a fixed maximum per cell (125 for Tokyo and NumoSim-LA, 30 for PoL). For POI $q \in \mathcal{P}(c)$, we embed its category and relative position within the cell, $\mathbf{h}_{c,q} = f_{\text{poi}}\big(\text{cat}_{c,q}, \Delta x_{c,q}, \Delta y_{c,q}\big) \in \mathbb{R}^{64}$. We aggregate POIs into a compact cell summary via attention, $\mathbf{s}_c = \sum_{q \in \mathcal{P}(c)} \alpha_{c,q} \mathbf{h}_{c,q}$ with $\alpha_{c,q} = \text{softmax}\big(\mathbf{w}^\top \mathbf{h}_{c,q}\big)$, so the model can emphasize functionally informative POIs when forming the cell semantics. We combine the cell semantics with a coarse absolute location code via sinusoidal positional encoding $\text{PE}(i, j) \in \mathbb{R}^{128}$ to form the cell embedding $\mathbf{e}_c^{\text{cell}} \in \mathbb{R}^{128}$ (Eq. 1). This cell embedding module is shared by both dense and sparse encoders. Implementation details of QuadTree construction and POI buffering are deferred to supplementary material.

Temporal encoder. Human mobility is highly periodic, so the same movement pattern can be typical or suspicious depending on when it occurs. We attach a single calendar embedding to each slot (d, s) that captures daily, weekly, and longer-term context. Specifically, we encode a set of calendar attributes $\{\phi_k(d,s)\}_{k\in\mathcal{K}_{\text{time}}}$ and concatenate their encodings into $\mathbf{e}^{\text{tm}}_{d,s} \in \mathbb{R}^{66}$. Together with the cell embedding, this gives the two shared embeddings:

$$\mathbf{e}^{\text{cell}}_c = \text{MLP}_{\text{sp}}\big([\mathbf{s}_c; \text{PE}(i_c, j_c)]\big), \quad (1) \qquad \mathbf{e}^{\text{tm}}_{d,s} = \text{Concat}_{k\in\mathcal{K}_{\text{time}}}\big(\text{Enc}_k(\phi_k(d,s))\big), \quad (2)$$

where $\mathcal{K}_{\text{time}}$ includes hour-of-day, day-of-week, day-of-month, part-of-day (coarse bins), week-of-month, and month-of-year. This module is shared by both encoders.

3.2.2 DENSETRAJEMBED.

DenseTrajEmbed (Fig. 2) converts a dense GPS stream into an HTI $\mathbf{X} \in \mathbb{R}^{D\times S\times C}$. The raw GPS stream is discretized into D days and S=96 slots per day (15-minute slots; D=66 days in our experiments). Each slot holds P=90 points on a 10 s grid, with gaps filled by same-day interpolation and zero padding applied at the end if necessary. Dense labels and encoder share the same 15-minute cells; 5-minute stay-point slots keep short stops and trips apart. This yields a regular tensor of shape $(D, S, P, 3)$ together with a validity mask $m_{d,s,p} \in \{0, 1\}$ for real values.

Spatio-semantic pooling. Each of the P points in slot (d, s) is mapped to a cell embedding $\mathbf{e}^{\text{cell}}_{c_{d,s,p}}$ via Eq. 1. We pool the P point embeddings into a slot-level spatial token using learned attention pooling:

$$\beta_{d,s,p} = \text{softmax}\Big(\mathbf{v}^\top_{\text{pool}} \tanh(\mathbf{W}_{\text{pool}}\mathbf{e}^{\text{cell}}_{c_{d,s,p}})\Big), \quad \tilde{\mathbf{x}}^{\text{sp}}_{d,s} = \sum_{p=1}^{P} \beta_{d,s,p}\mathbf{e}^{\text{cell}}_{c_{d,s,p}} \in \mathbb{R}^{128}. \quad (3)$$

This produces one 128-d spatial token per slot, with the pooling weights allowing the model to emphasize informative points (e.g., transitions and turns) and downweight redundant samples.

Kinematic encoder. Many anomalies are expressed in motion rather than in endpoints, such as unusual speed profiles, abrupt accelerations, or atypical turning behavior. For each slot (d, s), we summarize within-slot dynamics into a compact normalized descriptor vector $\mathbf{g}_{d,s} \in \mathbb{R}^{11}$ computed from consecutive GPS points inside the slot. It captures velocity and speed statistics, directionality (via $\sin$ / $\cos$ of bearing), turning variability, acceleration, and extreme statistics (min/max speed and acceleration). We then map this descriptor to a slot-level embedding $\mathbf{e}^{\text{kn}}_{d,s} = \text{Enc}_{\text{kn}}(\mathbf{g}_{d,s}) \in \mathbb{R}^{32}$.

Slot fusion. The final slot token is obtained by fusing the pooled spatial token with slot-level temporal and kinematic context, $\mathbf{x}_{d,s} = \text{MLP}_{\text{fuse}}\Big([\tilde{\mathbf{x}}^{\text{sp}}_{d,s}; \mathbf{e}^{\text{tm}}_{d,s}; \mathbf{e}^{\text{kn}}_{d,s}]\Big) \in \mathbb{R}^C$. Stacking all slots yields the HTI $\mathbf{X} \in \mathbb{R}^{D\times S\times C}$.

3.2.3 SPARSETRAJEMBED.

SparseTrajEmbed converts sparse stay-point logs into the same $D \times S \times C$ HTI consumed by CFT. Instead of dense GPS points, the input consists of an interleaved sequence of stop and trip events: $S_1, T_{1\to2}, S_2, T_{2\to3}, \ldots$, where each stop S_i records a location and dwell time, and each trip $T_{i\to i+1}$ records travel between consecutive stops.

Event-to-grid mapping. Each event spans a contiguous range of days and slots in the $D \times 288$ grid, determined by its start and end timestamps. Multi-slot events repeat their features across all occupied slots. When a stop and trip share the same 5-minute boundary slot, the slot keeps the stop's features, and a *stop fraction* $f \in [0, 1]$ records the stop's share of the slot minus the trip's share, clipped at zero.

Three feature groups. SparseTrajEmbed computes the same three feature groups as DenseTrajEmbed, instantiated differently for stops and trips. The spatio-semantic feature for stops is the cell embedding (Eq. 1) of the stop location; trip slots receive no spatio-semantic feature (zero vector). Attention along the slot axis lets a trip slot read its origin and destination from the stops around it, so trips need no location feature of their own.

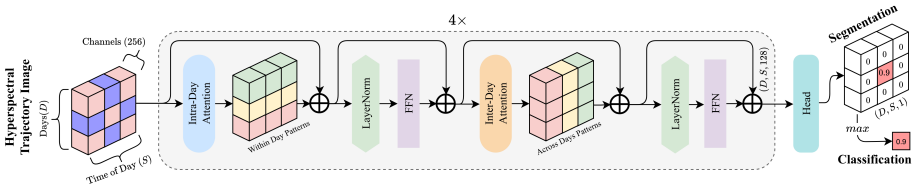


Figure 3: Cyclic Factorized Transformer (CFT) factorizes attention by interleaving intra-day and inter-day attentions. Each intra-day attention captures within-day patterns, while each inter-day attention captures cross-day routine patterns.

The kinematic features differ by event type, $\mathbf{g}_{\text{stop}} = [\log_{10}(\text{dur}+1),\ 0]$ and $\mathbf{g}_{\text{trip}} = [\log_{10}(\text{dur}+1),\ \log_{10}(\text{dist}+1)]$, where $\mathbf{g}_{\text{stop}}$ encodes only dwell duration (a zero distance signals a stationary event) and $\mathbf{g}_{\text{trip}}$ captures the travel time and straight-line distance between consecutive stops. Both are projected through a shared Enc'_{kn} to $\mathbb{R}^{32}$, analogous to Enc_{kn} in DenseTrajEmbed but with a 2-dimensional input suited to event-level kinematics rather than per-slot GPS statistics.

Slot fusion and type embedding. The three feature groups are concatenated and projected by a linear layer with layer normalization, $\mathbf{x}_{d,s} = \text{Proj}([\mathbf{e}^{\text{sp}}_{d,s};\ \mathbf{e}^{\text{tm}}_{d,s};\ \mathbf{e}^{\text{kn}}_{d,s}]) \in \mathbb{R}^C$. A learned type embedding is then added to each occupied slot to distinguish stops from trips, $\mathbf{x}_{d,s} \leftarrow \mathbf{x}_{d,s} + f \cdot \mathbf{e}_{\text{stop}} + (1-f) \cdot \mathbf{e}_{\text{trip}}$, where $\mathbf{e}_{\text{stop}}, \mathbf{e}_{\text{trip}} \in \mathbb{R}^C$ are learned embeddings and f is the stop fraction (0 for pure trip slots). The number of days D is fixed at 66 for the dense encoder but varies across agents in the sparse setting; days added as padding never produce a prediction or enter the loss.

3.3 Cyclic Factorized Transformer (CFT)

Given the HTI $\mathbf{X} \in \mathbb{R}^{D\times S\times C}$ produced by either encoder, a naive approach is to flatten it into a length-DS sequence and apply a standard Transformer. This is very expensive for multi-month horizons and, more importantly, it discards the temporal structure present in human mobility data. Specifically, human movement exhibits two complementary temporal structures: *within-day episodes* (e.g., detours, stop-and-go transitions, prolonged dwells during a trip) and *across-day routines* (e.g., recurring morning coffee, commute timing, regular gym visits). To preserve this cyclic day×time-of-day structure while scaling to long horizons, we introduce the Cyclic Factorized Transformer (CFT). As shown in Fig. 3, CFT factorizes attention along the two axes of the HTI $\mathbf{X}$, performing intra-day attention over the slot axis (S) to model daily episodes and inter-day attention over the day axis (D) to model routine consistency at the same time-of-day. This factorization reduces attention cost from $O(D^2S^2C)$ to $O((DS^2+SD^2)C)$, a $\frac{DS}{D+S} \approx 85\times$ reduction for D=120, S=288.

CFT is related to axial attention (Ho et al., 2019; Wang et al., 2020) and to factorized video and multi-agent attention (Bertasius et al., 2021; Arnab et al., 2021; Ngiam et al., 2022), but its axes are neither spatial nor agent dimensions: slots encode within-day episodes and days encode repeated routines. We therefore encode each token's slot and day with a rotary position encoding (Su et al., 2024). As with the position embedding of ViT (Dosovitskiy et al., 2020), it is applied once to the input tokens before the first CFT block, not to the queries and keys of every layer.

Intra-Day Attention. Each intra-day layer applies multi-head self-attention over all S slots within a single day, modeling short-horizon episode structure such as prolonged stays and unusual speeding. For a fixed day d, the row $\mathbf{X}_{d,:} \in \mathbb{R}^{S\times C}$ is treated as a length-S sequence; collecting all days gives output in $\mathbb{R}^{D\times S\times C}$.

Inter-Day Attention. Each inter-day layer mirrors the intra-day design but attends over the D days at a fixed time-of-day slot s, capturing routine consistency and deviations across days such as changes in habitual commute or activity timing. The column $\mathbf{X}_{:,s} \in \mathbb{R}^{D\times C}$ is treated as a length-D sequence; stacking across all slots yields output in $\mathbb{R}^{D\times S\times C}$.

Interleaved Blocks. CFT stacks L interleaved blocks (Fig. 3; L=4 for stay-point inputs and L=6 for dense GPS), each applying IntraAttn → FFN → InterAttn → FFN with Pre-LN residual connections for stable optimization of the long temporal axes. Interleaving lets within-day evidence inform cross-day comparisons at every depth. The final representation $\mathbf{Z} = \mathbf{X}^{(L)} \in \mathbb{R}^{D\times S\times C}$ jointly encodes both scales.

3.4 PREDICTION HEAD

A two-layer MLP produces per-slot anomaly logits from the final representation: $\hat{y}_{d,s} = \sigma(\mathrm{MLP}_{\mathrm{clf}}(\mathbf{Z}_{d,s}))$, yielding a segmentation map $\hat{\mathbf{Y}} \in [0,1]^{D\times S}$. Temporal localization reads directly from this map, while agent-level detection takes $\hat{y}_{\text{agent}} = \max_{d,s} \hat{y}_{d,s}$ over observed slots. We optimize $\mathcal{L} = \mathcal{L}_{\mathrm{BCE}} + \lambda \mathcal{L}_{\mathrm{Dice}}$ with a positive-class weight in the BCE term to address class imbalance; Appendix D lists the per-benchmark values.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate TITAnD on four benchmarks (full statistics in Appendix A). The public sparse benchmarks are NumoSim-LA (Stanford et al., 2024) (about 80K agents, 55 days) and Urban Patterns of Life (PoL) (Züfle et al., 2023; Amiri et al., 2023; 2024) (Atlanta, 3K agents, 464 days), which labels anomalous agents but not anomalous slots. No public dense trajectory anomaly benchmark exists, so Dense (Tokyo) uses private GPS from an agent-based simulator (about 18K agents, 66 days at 10 s resolution), with anomalies injected by an independent red team whose rules were hidden from us. Dense2Sparse (Tokyo) uses a larger, distinct cohort from the same source converted to stay-points (about 80K agents, 120 days). All benchmarks use agent-disjoint splits.

Evaluation and Metrics. We report area under the Precision–Recall curve (AUC-PR) and mean Intersection-over-Union (mIoU); mIoU is computed separately for each model and granularity at the F1-maximizing threshold on the evaluated predictions. We evaluate at two granularities: *temporal* (per-slot anomaly localization) and *agent* (per-agent anomaly detection). Sparse baselines (USTAD, ICAD) are originally self-supervised; for fair comparison we keep their published encoders and inputs but train them with the same supervision, using the same data, labels, and evaluation.

Training Details. All benchmarks use the loss defined in Sec. 3.4. Sparse baselines replace their self-supervised heads with a classification head and are fine-tuned end-to-end without an auxiliary loss. Training uses mixed precision on NVIDIA H200 GPUs; Appendix D gives the per-benchmark training settings.

4.2 COMPARISON TO PRIOR METHODS

Core comparison. Table 1 reports AUC-PR and mIoU at temporal and agent granularity; PoL supplies agent-level metrics only. Sparse methods use stop or stay-point sequences, whereas the HTI backbones use the same trajectory images with a flat Transformer, U-Net, SegFormer, DINOv2+Adapter, or CFT.

Sparse methods such as USTAD (Wen et al., 2024) and ICAD (Azarijoo et al., 2025) rely on stop-level representations. On NumoSim-LA, USTAD reaches 0.53 agent AUC-PR and ICAD reaches 0.30; at the temporal level they fall to 0.46 and 0.23, respectively. Both see the same stay-points as TITAnD, but as one long list of visits, where visits at the same time on other days can be far apart. The HTI puts them in one column, and inter-day attention compares them directly.

All HTI backbones use the same cell embedding, isolating the contribution of the backbone. The flat Transformer treats the day$\times$slot image as a one-dimensional sequence; U-Net (Ronneberger et al., 2015) uses an encoder–decoder CNN with skip connections; SegFormer (Xie et al., 2021) uses hierarchical attention with a lightweight decoder; and CFT factorizes attention along the day and slot axes. CFT is strongest or tied on every AUC-PR column: 0.59/0.65 temporal/agent AUC-PR on NumoSim-LA, 0.20/0.78 on Dense2Sparse Tokyo, and 0.84/1.00 on Dense Tokyo. The largest margin is temporal localization on dense GPS (0.84 versus 0.60 for the capacity-matched flat Transformer and 0.50 for U-Net), where global routine context and fine motion evidence both matter.

On the sparse benchmarks the flat Transformer is the weakest HTI backbone despite its more expensive full attention, suggesting that, even with calendar features, it must infer the cyclic day$\times$slot structure from positions in a long one-dimensional sequence. CNN backbones exploit a useful local inductive bias but have no explicit mechanism for comparing the same time of day across days. This distinction is clearest on Dense Tokyo: U-Net is close to CFT for coarse agent detection (0.95 versus

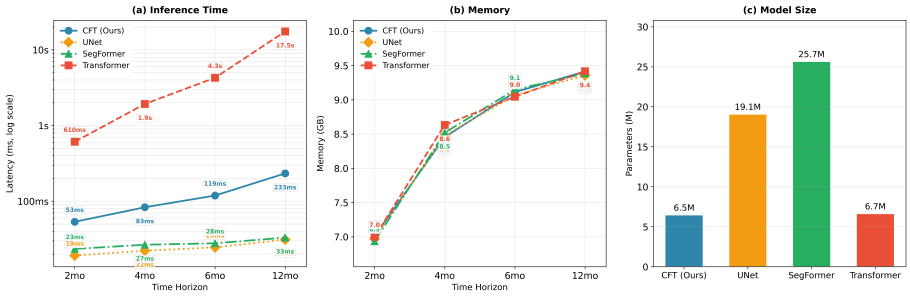


Figure 4: Scaling over 2–12 months on one NVIDIA H200 (batch size 4) with sparse stay-point HTIs of shape $D \times 288 \times 256$: (a) inference latency, (b) peak GPU memory, and (c) parameters.

Table 1: Core comparison across dense and sparse benchmarks. Sparse methods use stop/stay-point sequences; HTI backbones use both input types. $---$: unsupported or unavailable.

		PoL		**Sparse (NumoSim-LA)**				**Dense2Sparse (Tokyo)**				**Dense (Tokyo)**			
		Agent		Temporal		Agent		Temporal		Agent		Temporal		Agent	
Type	Backbone	AUC-PR	mIoU	AUC-PR	mIoU	AUC-PR	mIoU	AUC-PR	mIoU	AUC-PR	mIoU	AUC-PR	mIoU	AUC-PR	mIoU
Sparse	USTAD	0.06	0.09	0.46	0.67	0.53	0.70	0.03	0.52	0.41	0.66	—	—	—	—
	ICAD	0.06	0.36	0.23	0.59	0.30	0.62	0.01	0.51	0.36	0.65	—	—	—	—
	LM-TAD	0.34	0.60	0.03	0.50	0.06	0.49	—	—	—	—	—	—	—	—
HTI	Transformer	0.10	0.44	0.06	0.56	0.16	0.57	0.01	0.52	0.24	0.60	0.60	0.70	**1.00**	**1.00**
	U-Net	0.44	0.69	0.29	0.63	0.57	0.75	0.03	0.52	0.60	0.72	0.50	0.67	0.95	0.77
	SegFormer	0.68	0.76	0.33	0.66	0.62	**0.76**	0.03	0.53	0.52	0.70	0.21	0.59	0.95	0.91
	DINOv2+Adp.	0.25	0.56	0.12	0.59	0.46	0.57	0.07	0.54	0.67	0.73	0.19	0.61	**1.00**	0.98
	CFT (ours)	**0.71**	**0.78**	**0.59**	**0.74**	**0.65**	**0.76**	**0.20**	**0.58**	**0.78**	**0.80**	**0.84**	**0.84**	**1.00**	**1.00**
Time series	Chronos-2	—	—	—	—	—	—	—	—	—	—	0.69	0.74	0.82	0.61

1.00 AUC-PR), which is near ceiling for most backbones, while CFT has its largest advantage on the harder temporal-localization task (0.84 versus 0.50).

PoL, Chronos-2, and DINOv2. On PoL, TITAnD reaches 0.71 agent AUC-PR compared with 0.68 for SegFormer and 0.34 for LM-TAD (Mbuya et al., 2024), carrying the sparse comparison to a second public benchmark. On Dense Tokyo, TITAnD also exceeds both Chronos-2 (Ansari et al., 2025) and DINOv2+Adapter (Oquab et al., 2024) in temporal AUC-PR.

Efficiency. CFT is the smallest evaluated HTI backbone, with 6.5M parameters on NumoSim-LA and PoL versus 19–26M for U-Net and SegFormer, and it scales favorably with the observation horizon (Fig. 4). At two months it is already 11× faster than the similarly sized flat Transformer (53 ms versus 610 ms); over 2–12 months, CFT's latency grows roughly linearly while the flat Transformer's grows quadratically, widening the gap to 75× at 12 months (233 ms versus 17.5 s). Peak memory remains comparable across backbones (about 7–10 GB) with flash attention (Dao et al., 2022); U-Net and SegFormer are faster than CFT but less accurate (Table 1).

4.3 Ablations

We ablate both encoder features and attention axes while keeping the other component fixed (Table 2); a single-axis model doubles its layers to keep the parameter count.

Encoder features. On Dense, temporal features alone are insufficient (0.03), and adding spatial features yields the largest jump (0.03→0.62), confirming that location context is the dominant signal in dense GPS. On Dense2Sparse, each feature group adds to the previous one (0.18→0.19→0.20). Kinematic features give a further gain on Dense (0.62→0.84), reflecting that dense GPS retains rich motion dynamics.

Attention axes. On Dense (Tokyo), intra-day attention alone achieves 79% of full AUC-PR, indicating that most anomaly signal in the dataset arises from within-day deviations. Inter-day attention alone captures 39%, reflecting that cross-day comparison provides weaker signal when anomalies are locally coherent within individual days. On Dense2Sparse, intra-day and inter-day attention alone reach 0.12 and 0.07, below their factorized combination (0.20). Full interleaved attention is best among the attention variants, and the Full (CFT) entries in both parts of Table 2 correspond to the same model architecture.

Table 2: Ablation of encoder features and attention axes with temporal AUC-PR.

	Encoder features		Attention axes		
	Temporal	Spatial + Temporal	Intra-day	Inter-day	**Full (CFT)**
Dense (Tokyo)	0.03	0.62	0.66	0.33	**0.84**
Dense2Sparse (Tokyo)	0.18	0.19	0.12	0.07	**0.20**

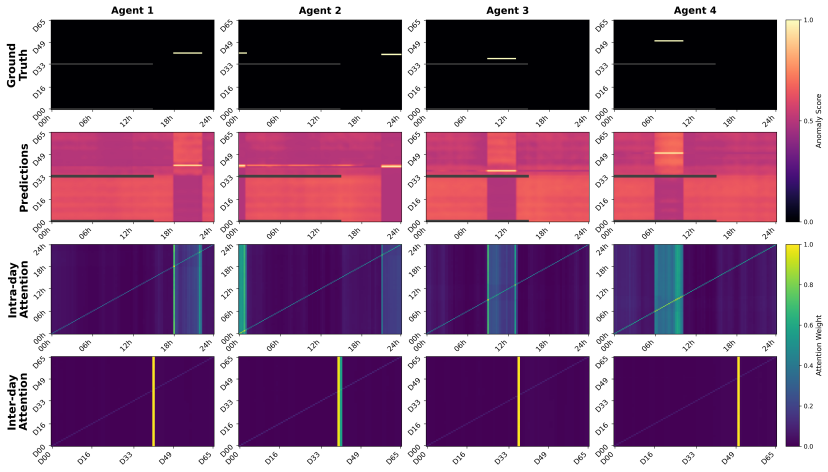


Figure 5: Qualitative analysis. Rows show, from top to bottom, ground truth (gray denotes missing data), CFT predictions, intra-day (slot×slot) attention, and inter-day (day×day) attention.

Qualitative analysis. Attention-rollout visualizations in Figure 5 show complementary localization along the two CFT axes: inter-day attention concentrates on anomalous days, while intra-day attention concentrates on the corresponding within-day slots. The same behavior appears for anomalies driven by wide-area movement, atypical dwelling, and prolonged inactivity.

5 Limitations

Because verified real-world mobility anomaly labels are scarce, our evaluation uses synthetically injected anomalies. Constructing verified labels requires access to sensitive mobility traces and ground-truth incident verification, which makes large public benchmarks difficult to obtain. Synthetic injection therefore provides a controlled way to compare detection performance across diverse anomaly types. For dense trajectories, no public dataset currently provides multi-month GPS traces with anomaly annotations, so our Dense (Tokyo) experiments use private data from an agent-based simulator; an independent red team injected anomalies using rules withheld from the modeling team. Consequently, two of our four benchmarks share the same private Tokyo source, which limits the geographic and demographic diversity of the evaluation. Results on additional independent real-world datasets would strengthen the generalization claims. Finally, USTAD and ICAD were originally designed for self-supervised learning. We adapt their encoders to our supervised protocol to compare representations under the same labels and evaluation setting, while recognizing that this differs from their original training objectives.

6 Conclusion

We showed that sparse and dense trajectory anomaly detection are instances of the same vision problem. Through hyperspectral trajectory images and structure-aware attention along the cyclic axes of human mobility, TITAnD makes multi-month dense anomaly detection feasible for the first time and unifies dense and sparse trajectories in a single representation. TITAnD matches or beats every evaluated baseline in AUC-PR on all four benchmarks, at as few as 6.5M parameters and up to 75× lower latency than a flat Transformer at 12 months. Our ablations show that the two attention axes complement each other: neither alone matches their factorized combination, on dense and sparse inputs alike.

AI USE STATEMENT

Generative AI tools were used in a limited assistive role for proofreading and code assistance.

ETHICS STATEMENT

Human mobility traces are sensitive and can reveal habits, associations, and frequently visited locations. Trajectory anomaly systems could cause privacy harm or be misused for surveillance if deployed without appropriate safeguards. We therefore advocate data minimization, access controls, retention limits, human oversight, and explicit documentation of dataset provenance, labeling assumptions, intended uses, and known limitations. Our experiments study research benchmarks and do not claim that model predictions alone are suitable for consequential decisions about individuals.

REPRODUCIBILITY STATEMENT

The appendix reports dataset statistics, preprocessing and label aggregation, pseudocode for HTI construction and CFT, encoder dimensions, training hyperparameters, and robustness analyses. The public NumoSim and PoL benchmarks support independent evaluation of the sparse setting. The private dense benchmark cannot be redistributed; the implementation, preprocessing pipeline, and trained model artifacts are included in the supplementary material so that the method can be evaluated on comparable authorized datasets.

REFERENCES

Hossein Amiri, Shiyang Ruan, Joon-Seok Kim, Hyunjee Jin, Hamdi Kavak, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. Massive trajectory data based on patterns of life. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pp. 1–4, 2023.

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*, 2024. URL https://arxiv.org/abs/2410.01844.

Abdul Fatir Ansari et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025. URL https://arxiv.org/abs/2510.15821.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *IEEE/CVF International Conference on Computer Vision*, 2021.

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 (SIGSPATIAL '25)*, pp. 595–606, 2025. doi: 10.1145/3748636.3762774.

Bita Azarijoo, John Krumm, and Cyrus Shahabi. IBAD: Interpretable behavioral anomaly detection on human mobility data. *arXiv preprint arXiv:2606.16023*, 2026.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, 2021.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

Jean-Pierre Eckmann, S Oliffson Kamphorst, and David Ruelle. Recurrence plots of dynamical systems. *Europhysics Letters*, 4(9):973–977, 1987.

Marta C González, César A Hidalgo, and Albert-László Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

Nima Hatami, Yann Gavet, and Johan Debayle. Classification of time-series images using deep convolutional neural networks. In *Tenth International Conference on Machine Vision (ICMV 2017)*, volume 10696, pp. 106960Y. SPIE, 2018.

Eelco Herder and Patrick Siehndel. Daily and weekly patterns in human mobility. In *UMAP Workshops*, pp. 338–340, 2012.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Erfan Hosseini Sereshgi, Mauryan Uppalapati, Yueyang Liu, Lance Kennedy, Andreas Züfle, and Carola Wenk. Semantic anomaly detection in human trajectories: Preserving behavioral patterns through calendar representations. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection*, pp. 33–42, 2025.

Lance Kennedy and Andreas Züfle. Kinematic detection of anomalies in human trajectory data. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection*, pp. 94–97, 2024.

Zekun Li, Shiyang Li, and Xifeng Yan. Time series as images: Vision transformer for irregularly sampled time series. *Advances in Neural Information Processing Systems*, 36:49187–49204, 2023.

Yuxuan Liang, Kun Ouyang, Yiwei Wang, Xu Liu, Hongyang Chen, Junbo Zhang, Yu Zheng, and Roger Zimmermann. Trajformer: Efficient trajectory classification with transformers. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 1229–1237, 2022.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025a.

Yichen Liu, Yan Lin, Shengnan Guo, Zeyu Zhou, Youfang Lin, and Huaiyu Wan. TrajMamba: An efficient and semantic-rich vehicle trajectory pre-training model. *arXiv preprint arXiv:2510.17545*, 2025b.

Jonathan Mbuya, Dieter Pfoser, and Antonios Anastasopoulos. Trajectory anomaly detection with language models. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, 2024. URL https://arxiv.org/abs/2409.15366.

Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David Weiss, Ben Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260, 1984.

Christian M Schneider, Vitaly Belik, Thomas Couronné, Zbigniew Smoreda, and Marta C González. Unravelling daily human mobility motifs. *Journal of The Royal Society Interface*, 10(84): 20130246, 2013.

Wei Shao, Ziquan Fang, Lu Chen, and Yunjun Gao. Towards trajectory anomaly detection: a fine-grained and noise-resilient framework. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2490–2501, 2025.

Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

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*, pp. 68–78, 2024.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *European conference on computer vision*, pp. 108–126. Springer, 2020.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. *arXiv preprint arXiv:2410.16032*, 2024.

Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3939–3945, 2015.

Haomin Wen, Shurui Cao, Zeeshan Rasheed, Khurram Hassan Shafique, and Leman Akoglu. Uncertainty-aware human mobility modeling and anomaly detection. *arXiv preprint arXiv:2410.01281*, 2024.

Haomin Wen, Shurui Cao, and Leman Akoglu. CoBAD: Modeling collective behaviors for human mobility anomaly detection. *arXiv preprint arXiv:2508.14088*, 2025.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations (ICLR)*, 2023.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

Junyi Xie, Jina Kim, Yao-Yi Chiang, Lingyi Zhao, and Khurram Shafique. Bestad: Behavior-aware spatio-temporal anomaly detection for human mobility data. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection*, pp. 56–59, 2025.

Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. ibat: detecting anomalous taxi trajectories from gps traces. In *Proceedings of the 13th international conference on Ubiquitous computing*, pp. 99–108, 2011.

Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*, pp. 791–800. Association for Computing Machinery, 2009. doi: 10.1145/1526709.1526816.

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. *arXiv preprint arXiv:2411.03859*, 2024.

Andreas Züfle, Carola Wenk, Dieter Pfoser, Andrew Crooks, Joon-Seok Kim, Hamdi Kavak, Umar Manzoor, and Hyunjee Jin. Urban life: a model of people and places. *Computational and Mathematical Organization Theory*, 29:20–51, 2023.

This document provides supplementary details for the main paper. We include dataset statistics and provenance (Sec. A), pseudocode for TITAnD, CFT, and QuadTree (Sec. B), encoder implementation details (Sec. C), per-benchmark training hyperparameters (Sec. D), robustness and anomaly-type analyses (Sec. E), extended qualitative analysis (Sec. F), and the relation to period-folding time-series models (Sec. G).

A Dataset Details

Table 3: Dataset statistics.

	Scale			Splits			Anomaly Prevalence	
Benchmark	Modality	Agents	Duration	Train	Val	Slots/Agent	Slot Rate	Agent Rate
Sparse (NumoSim-LA)	Stay-points	80,381	55 days	50,238	30,143	15,840	0.024%	0.47%
Dense (Tokyo)	GPS (10 s)	18,469	66 days	14,772	3,697	6,336	0.20%	46.4%
Dense2Sparse (Tokyo)	Stay-points	80,463	120 days	50,416	30,047	34,560	0.009%	0.57%
Sparse (PoL, Atlanta)	Stay-points	3,000	464 days	2,400	600	133,632	—	5.0%

Slot-level anomalies are 20–230$\times$ rarer than anomalous agents on every benchmark with slot labels (Table 3). Because the chance level of AUC-PR equals the positive rate, temporal AUC-PR is lower than agent AUC-PR for every model in Table 1, and the two granularities should not be compared on raw values. For the same reason, temporal mIoU stays near 0.5 even for weak detectors: the normal-class IoU is close to 1 when almost every slot is normal.

Sparse (NumoSim-LA). NumoSim (Stanford et al., 2024) provides natively sparse stay-point trajectories from the Los Angeles region. Each event records arrival time, departure time, and a POI location; no dense GPS is available. Trajectories span $\sim$55 days (28 past + 27 future). Anomaly labels are aggregated to 5-minute slots via the sparse labeling rule of Sec. 3.1; a trip slot takes the trip's own label where the benchmark provides one (Dense2Sparse) and otherwise the label of either adjacent stay.

Dense (Tokyo). A private dense GPS dataset from the Tokyo metropolitan region sampled at $\sim$10-second resolution. Each agent spans 66 days (33 past + 33 future), yielding $\sim$570K GPS points per agent. Anomaly labels are provided at per-point granularity and aggregated to 15-minute slots via OR aggregation (Sec. 3.1). The trajectories come from an agent-based patterns-of-life simulator. Synthetic agents are sampled from public census microdata, activity choices are produced by a random-utility model calibrated on household-travel and time-use surveys, and mobility is realized on an OpenStreetMap road network using map-constrained routing. An independent red team injected anomalies without disclosing the rules to the modeling team.

Dense2Sparse (Tokyo). A larger, distinct cohort from the same Tokyo source is converted to stay-point sequences spanning 120 days. The shared geography and collection process permit a modality comparison, but the different agents and horizons mean this is not a paired dense-versus-sparse ablation.

Patterns of Life (PoL). PoL (Züfle et al., 2023; Amiri et al., 2023; 2024) is a public sparse benchmark simulated on a map of Atlanta, with 3,000 agents over a 450-day history and a 14-day evaluation window. Its release identifies 150 anomalous agents over a 14-day evaluation window but does not provide per-slot anomaly labels for that window, so we report agent AUC-PR rather than temporal localization metrics. Because this leaves only 30 anomalous validation agents, agent AUC-PR varies across training runs; for TITAnD and SegFormer, the two strongest methods, PoL entries are the mean over six training seeds. Each stay-point pixel additionally encodes the category of the visited venue (apartment, workplace, restaurant, or pub), the semantic information also available to LM-TAD, and stay-points are split at calendar-day boundaries. Because PoL labels agents rather than stays, TITAnD is trained with weak slot targets: every stay of an anomalous agent within the 14-day window is positive, trip slots inherit the label of their adjacent stays, and the loss covers that window only. Table 4 lists the remaining settings. The supervised LM-TAD adaptation uses place-token BCE with positive class weight 70, learning rate 3×10^{-4}, and batches of 256 daily sequences. DINOv2+Adapter uses the same PoL input as TITAnD.

B PSEUDOCODE

B.1 TITAND: END-TO-END FORWARD PASS

Listing 1 shows the high-level three-stage pipeline. Encoder internals (spatial encoding, temporal calendar encoding, kinematic features) are detailed in Sec. 3.2; the pseudocode focuses on inter-stage data flow.

Listing 1: TITAnD forward pass.

```
def titand_forward(batch):
    B, D, S = batch.mask.shape

    # Stage 1: Modality-specific encoder
    # (DenseTrajEmbed or SparseTrajEmbed)
    sp = encode_spatial(batch)      # [B,D,S,128]
    tm = encode_temporal(batch)     # [B,D,S,66]
    kn = encode_kinematic(batch)    # [B,D,S,32]
    x = mlp_fuse(cat(sp, tm, kn)) # [B,D,S,C]
    x += type_emb(batch.stop_frac)
    x += venue_emb(batch.venue)   # PoL

    # Stage 2: CFT
    x = cft(x)

    # Stage 3: Prediction head
    y = mlp_clf(x).squeeze(-1)     # [B,D,S]
    y_agent = y[observed].max()    # [B]
    return y, y_agent
```

B.2 CYCLIC FACTORIZED TRANSFORMER (CFT)

Listing 2 details the interleaved block structure described in Sec. 3.3. Each block applies intra-day attention (over S slots within each day) followed by inter-day attention (over D days at the same time-of-day), both with LayerNorm and residual connections.

Listing 2: CFT forward pass.

```
def cft_forward(x):
    B, D, S, C = x.shape
    # position encoding, once on the input (as in ViT)
    x = rope_slot(x)  # slot position
    x = rope_day(x)   # day position

    for i in range(L):
        # Intra-day: self-attn over S per day
        x = x + intra_attention[i](layer_norm(x))
        x = x + ffn_intra[i](layer_norm(x))

        # Inter-day: self-attn over D per slot
        x = x + inter_attention[i](layer_norm(x))
        x = x + ffn_inter[i](layer_norm(x))
    return x # [B, D, S, C]
```

B.3 QUADTREE SPATIAL DISCRETIZATION

Listing 3 shows the adaptive grid construction referenced in Sec. 3.2. Cells with more than `cap` POIs are recursively split into quadrants until reaching the minimum resolution.

Listing 3: QuadTree construction.

```
def build_quadtree(bounds, pois,
        base=200, cap=125, min_sz=10):
```

```
    # base: grid pitch (m) for (i,j)
    # cap: POI split threshold
    # min_sz: smallest cell (m)
    root = Node(bounds)
    queue = [(root, all_pois)]
    leaves = []

    while queue:
        node, idx = queue.popleft()
        half = min(node.w, node.h) / 2

        if len(idx) > cap and half >= min_sz:
            for c in split_quadrants(node):
                c_idx = idx[in_bounds(pois, c)]
                queue.append((c, c_idx))
        else:
            leaves.append(node)

    for i, leaf in enumerate(leaves):
        leaf.cell_id = i
    lut = build_lookup(leaves, res=5)
    return leaves, lut
```

B.4 SPATIO-SEMANTIC CELL EMBEDDING (Enc_{SP})

Listing 4 shows the shared `cell_embed` core (GPS → QuadTree → 128-d). The dense encoder refines each of the P=90 cell embeddings in a slot with the point's offset inside its cell and pools them via learned attention. The sparse encoder embeds stops by the cell at their location; trip slots carry no spatial feature and take their spatial context from neighbouring stops through attention.

Listing 4: Spatio-semantic cell embedding.

```
def cell_embed(cell_id, i, j):
    # Semantic: attention-pool POIs in cell
    h = poi_mlp(cat(cat_emb(poi_cat[cell_id]),
                    pos_mlp(dxy[cell_id]))) # [N,64]
    alpha = softmax(w_sem @ h, poi_mask)
    s = dot(alpha, h) if n_poi[cell_id] else e_empty # [64]
    # e_empty: learned embedding for cells without POIs
    # Spatial: sinusoidal grid PE
    pe = sinusoidal_pe(i, j)  # [128]
    return mlp_sp(cat(s, pe)) # [128]

# Dense: pool P cell embeddings per slot
def enc_sp_dense(gps):          # [P, 2]
    cells, ij, off = quadtree(gps)
    e = vmap(cell_embed)(cells, ij) # [P, 128]
    e = refine(e, off)          # point offset in cell
    beta = softmax(v_pool @ tanh(W_pool @ e))
    return dot(beta, e)         # [128]

# Sparse stop: cell embedding at location
def enc_sp_stop(stop):
    cell, ij = quadtree(stop.gps)
    return cell_embed(cell, *ij)     # [128]

# Sparse trip: no spatial feature
def enc_sp_trip(trip):
    return zeros(128)                # [128]
```

B.5 TEMPORAL ENCODER (Enc_{T})

Listing 5 shows the shared temporal encoder used identically for both dense and sparse encoders. Six calendar attributes are encoded independently and concatenated to a 66-d vector: hour, day of week, day of month, and month use fixed sinusoidal encodings, while part of day and week of month use learned embeddings.

Listing 5: Temporal encoder.

```
def enc_temporal(t):
    # 6 calendar attributes → 66-d total
    h = cat(
        sincos(t.hour, 24),        # 8-d
        sincos(t.day_of_week, 7), # 6-d
        sincos(t.day_of_month, 31), # 8-d
        sincos(t.month, 12),       # 12-d
        emb_pod(t.part_of_day),    # 16-d, learned
        emb_wom(t.week_of_month), # 16-d, learned
    )
    return h                       # [66]
```

B.6 KINEMATIC ENCODERS (Enc_{KN})

Listing 6 shows both variants. The dense encoder derives an 11-d descriptor from raw GPS within a 15-minute slot. The sparse encoder uses a shared 2-d input for stops and trips: stops set distance to zero (signalling a stationary event), while trips carry travel duration and distance. Stops and trips share one linear projection; the dense encoder uses its own two-layer MLP.

Listing 6: Kinematic encoders (dense and sparse).

```
# Dense: 11-d from ~90 GPS points per slot
def enc_kn_dense(xy, t):  # [P, 2] metres, [P] s
    v = diff(xy) / diff(t)         # [P-1, 2]
    spd = norm(v, dim=-1)          # [P-1]
    brg = atan2(v[:,0], v[:,1])    # [P-1]
    ms, mc = sin(brg).mean(), cos(brg).mean()
    b = atan2(ms, mc)              # circular mean
    acc = norm(diff(v), dim=-1) / diff(t)[:-1]
    f = cat([v.mean(0)/200, sin(b), cos(b),
             spd.mean()/200, 1 - hypot(ms, mc),
             acc.mean()/50,
             spd.min()/200, spd.max()/200,
             acc.min()/50, acc.max()/50]) # 11-d
    return mlp_kn(f)  # Linear-ReLU-Linear, [32]

# Sparse stop: duration only, distance = 0
def enc_kn_stop(stop):
    f = [log10(1+stop.dur), 0]  # 2-d
    return gelu(linear_kn(f))   # [32]

# Sparse trip: duration and distance
def enc_kn_trip(trip):
    f = [log10(1+trip.dur),
         log10(1+trip.dist)]    # 2-d
    return gelu(linear_kn(f))   # [32]
# linear_kn shared across stops and trips
```

C Encoder Implementation Details

C.1 QuadTree Grid Statistics

The QuadTree starts from the bounding box of the region (the AOI polygon for Tokyo, the buffered POI extent for NumoSim-LA and PoL) and recursively splits any cell containing more than 125 POIs into four quadrants, stopping when children would be smaller than 10 m (Tokyo and NumoSim-LA; PoL, a 4 km area, uses 100 m base cells and a 25-venue threshold). This yields fine resolution in dense commercial districts and coarse cells in sparse residential areas.

Property	Value
Total leaf cells (Tokyo)	$\sim$78,000
Split threshold	125 POIs
Minimum cell size	10 m
Lookup resolution	5 m (for $O(1)$ vectorized queries)

C.2 POI Categories

For Tokyo we use 45 semantic POI categories grouped from raw OpenStreetMap tags (NumoSim-LA uses 16, and PoL its 4 venue categories):

Category group	Examples
Residential (6)	apartment, house, dormitory, ...
Commercial (8)	store, mall, supermarket, ...
Food & Drink (5)	restaurant, cafe, bar, ...
Office (4)	office building, coworking, ...
Healthcare (4)	hospital, clinic, pharmacy, ...
Education (5)	school, university, library, ...
Transportation (4)	station, parking, bus stop, ...
Recreation (5)	park, gym, cinema, ...
Other (4)	religious, industrial, ...

C.3 Feature Dimensions

Spatio-semantic cell embedding. Each POI is encoded by its category embedding (32-d) concatenated with a relative position MLP (32-d), yielding a 64-d POI feature. Attention pooling over all POIs in a cell produces a 64-d cell summary, which is concatenated with a 128-d sinusoidal grid position encoding $\mathrm{PE}(i, j)$ and projected via $\mathrm{MLP}_{\mathrm{sp}}$ to a 128-d cell embedding. For the dense encoder, learned attention pooling over the $P{=}90$ refined cell embeddings per slot produces a 128-d spatial token.

Temporal encoder. Six calendar attributes are independently encoded and concatenated: hour-of-day (8-d), day-of-week (6-d), day-of-month (8-d), month-of-year (12-d), part-of-day with 6 four-hour bins (16-d), and week-of-month (16-d), totaling 66 dimensions.

Kinematic encoder. For the dense encoder, $\mathrm{Enc}_{\mathrm{kn}}$ takes an 11-d descriptor (velocity components, direction, speed, turning variability, acceleration, min/max speed and acceleration) and projects it to 32 dimensions. For the sparse encoder, $\mathrm{Enc}'_{\mathrm{kn}}$ takes a 2-d input (log-duration, log-distance) and projects it to 32 dimensions.

Slot fusion. The three feature groups are concatenated ($128 + 66 + 32 = 226$ dimensions) and projected to $C{=}256$, by a linear layer with layer normalization in the sparse encoder and by a two-layer MLP in the dense encoder.

D Training Details

All benchmarks use a positively weighted binary cross-entropy (plus a Dice term for the benchmarks with $\lambda{>}0$ in Table 4), a linear learning-rate warmup followed by cosine decay, gradient clipping at 1.0, mixed precision, and SpecAugment with probability 0.5. Table 4 lists the main settings that

Table 4: Per-benchmark training hyperparameters. †Muon (Liu et al., 2025a) for weight matrices and AdamW for the remaining parameters.

Hyperparameter	Dense (Tokyo)	Sparse (NumoSim-LA)	Dense2Sparse (Tokyo)	PoL
Optimizer	Muon†	AdamW	AdamW	AdamW
Learning rate	5×10^{-3}	7×10^{-5}	2×10^{-4}	2×10^{-4}
LR warmup epochs	10	5	5	5
Epochs	40	20	20	20
Batch size per GPU	32	64	32	8
Gradient accumulation	1	1	1	8
Weight decay	0.1	0.01	0.01	0.01
Positive-class weight	400	70	70	5
Dice weight λ	0	1.0	0	0.3
Label smoothing	0.02	0	0	0

differ across benchmarks. For every method, Table 1 reports the checkpoint with the best validation score. The dense CFT has 16.2M parameters.

The comparison methods share the data splits, labels, and metrics. U-Net and SegFormer start from ImageNet-pretrained encoders and DINOv2+Adapter from a pretrained DINOv2 backbone, each fine-tuned end-to-end; CFT and the flat Transformer are trained from scratch. Chronos-2 uses the Chronos-2 (small) architecture trained from scratch, with a per-slot classification head in place of its forecasting head; on Dense Tokyo it reads coordinates and POI channels at 5-minute resolution. USTAD, ICAD, and LM-TAD are trained from scratch as supervised variants of their published architectures.

E Robustness and Anomaly-Type Analysis

NumoSim anomaly types. NumoSim-LA distinguishes unfamiliar-location visits from familiar-location revisits at shifted times. TITAnD obtains 0.51 temporal AUC-PR on unfamiliar-location anomalies and 0.50 on shifted-time revisits. The two are detected at nearly the same rate even though shifted-time revisits are spatially and semantically familiar, leaving timing relative to the agent's routine as the only signal. Tokyo does not expose anomaly-type labels because the injection scheme was withheld by the red team, so the same quantitative stratification is not available there.

Routine drift. HTI explicitly retains month and time-of-day features, while its semantic channels can map changed physical locations to related activity categories. These features may help with timing or location drift, but the present benchmarks do not provide a controlled drift annotation. We therefore treat robustness to seasonal schedule changes, relocation, and evolving routines as an open evaluation question rather than a demonstrated guarantee.

F Extended Qualitative Analysis

Figure 5 compares the ground truth, CFT predictions, and attention rollout for four Dense (Tokyo) agents. Attention is averaged over heads and layers; inter-day attention is additionally averaged over anomalous time slots, while intra-day attention is averaged over anomalous days. Across cases, inter-day attention concentrates on anomalous days and intra-day attention highlights the corresponding within-day windows.

For each agent, we compare the anomaly period against the established pattern of life during the 33-day baseline (Days 0–32). Throughout this analysis, *unique locations* refers to distinct grid cells visited; *novel locations* are cells not visited during the baseline; *POI enrichment* is the ratio of POI category exposure during anomaly versus baseline; and *velocity* is normalized per agent using min-max scaling.

F.1 AGENT 1 (DAY 41, 18:00–22:00)

During the baseline period, this agent follows a compact routine with limited variation. On Day 41 evening, the behavior shifts abruptly: the agent traverses a substantially wider region and exhibits different stop-and-go dynamics within a four-hour window. Inter-day attention peaks on Day 41, while intra-day attention concentrates on the ground-truth interval. The window combines a surge in distinct visited locations with extended stationary periods at atypical areas.

F.2 AGENT 3 (DAY 37, 09:00–13:00)

This anomaly is not dominated by spatial displacement. On Day 37 morning, the agent is stationary for most of the four-hour interval and exhibits repeated prolonged dwells, deviating from the baseline weekday pattern of regular movement and brief stops. Inter-day attention isolates Day 37 and intra-day attention concentrates on 09:00–13:00, showing how fine-grained temporal dynamics can support localization even when the visited region remains similar.

F.3 AGENT 2 (DAYS 40–41, 00:00–23:45)

This agent exhibited the longest anomaly in our evaluation: 24 continuous hours. During baseline, the agent frequented only 37 unique locations with strong residential preference (66%) and essentially zero engagement with entertainment venues ($<$0.5%). Starting midnight on Day 40, the agent visited 86 locations (83% novel) with a dramatic semantic shift: 6.4$\times$ enrichment in night club proximity and 2.4$\times$ enrichment near theaters—POI categories absent from their pattern of life. Between 21:15–22:30 on Day 40, the agent made five consecutive 15-minute stops in unpopulated areas, behavior inconsistent with normal evening activity. The 24-hour duration, spatial novelty, and entertainment venue engagement collectively constitute a comprehensive behavioral discontinuity.

F.4 AGENT 4 (DAY 50, 06:00–10:00)

This case presents the most geographically dramatic anomaly. Baseline morning activity showed a consistent 5 km commute with 2–3 brief stops. On Day 50, the agent traversed a 31$\times$14 km area—spanning the metropolitan region north-to-south—visiting 267 locations (81% novel). The trajectory formed an approximately linear path, suggesting deliberate transit rather than exploration. Most notably, the five longest dwells (15 minutes each, 07:00–08:15) occurred at education and religious facilities—institutional POI categories with negligible baseline presence. Despite covering 6$\times$ normal distance, velocity decreased 81%, indicating methodical movement with frequent stops. This combination of extreme spatial displacement, institutional engagement, and paradoxically reduced speed triggered high-confidence detection.

G RELATION TO PERIOD-FOLDING TIME-SERIES MODELS

TimesNet (Wu et al., 2023) and TimeMixer++ (Wang et al., 2024) fold a regularly sampled multivariate series into a period $\times$ phase grid whose period is estimated with the Fourier transform, and model within- and across-period variation with 2D convolutions (TimesNet) or axis-wise attention (TimeMixer++). TITAnD shares the idea of a two-dimensional view of time, but differs in motivation, data, representation, architecture, and task.

Role of the two-dimensional view. In TimesNet and TimeMixer++, the 2D view is a feature transformation applied inside a general-purpose backbone, with periods discovered from each signal; TimesNet re-estimates them at every layer. In TITAnD, the image is the task formulation. It maps dense and sparse trajectories into one representation, so the same architecture serves both; it compresses months of observations into a grid of $D \times S$ slots on which attention is factorized along the two axes, reducing cost from $O(D^2S^2C)$ to $O((DS^2+SD^2)C)$; and it aligns those axes with within-day and across-day human routines, so slots within a day capture episodes and days at the same time of day capture routines.

Data. These models take a single regularly sampled series. TITAnD takes trajectories in two forms, dense GPS streams and sparse stay-point events, and maps both into the same image.

Multi-modal pixels. In these models, a grid cell holds a projection of the measured values at one time step. In HTI, a pixel fuses spatial (map cell), semantic (nearby points of interest), temporal (calendar), and kinematic (speed, acceleration, bearing, and turning on dense GPS; duration and distance on sparse events) information aggregated from all observations inside its time-of-day slot.

Architecture that exploits the structure. The image is designed together with its backbone. CFT applies intra-day attention over the slots of each day to model episodes and inter-day attention across days at the same time of day to model routines, with separate rotary position encodings per axis. TimeMixer++ also uses axis-wise attention, but over estimated periods to separate seasonal and trend components; TimesNet uses local 2D convolutions, the operator class evaluated on HTI through the U-Net backbone in Table 1, where CFT outperforms it. Sec. 4.3 further shows that neither axis alone recovers full performance.

Task. TITAnD is trained with supervision for per-slot anomaly localization and agent-level detection over multi-month horizons of tens of thousands of cells, whereas TimesNet detects abnormal time points by reconstructing 100-step windows of sensor readings.