

Towards Predicting Urban Land Use Changes: A Dynamic Graph Alignment Perspective

YU FAN, Beijing University of Posts and Telecommunications, China

XINJIANG LU[✉], Baidu Research, China

HAO LIU, Hong Kong University of Science and Technology, China

PENGFEI WANG, Beijing University of Posts and Telecommunications, China

LIANG LIU, Beijing University of Posts and Telecommunications, China

HUADONG MA, Beijing University of Posts and Telecommunications, China

JINGBO ZHOU[✉], Baidu Research, China

Urban land use, intrinsically linked to people's daily activities, undergoes continuous evolution, presenting a complex interplay that remains partially understood. To bridge this gap, our study leverages fine-grained human mobility data to predict these changes, adopting a novel approach that conceptualizes "community-level" land use shifts as a regression problem and represents citywide changes through dynamic graphs. We harness recent advancements in graph neural networks (GNNs), which, despite their success in various applications, face challenges in directly predicting land use changes due to the temporal mismatch between the slow evolution of urban land and the immediacy of human mobility data. Our research stands out by introducing a temporal skeleton for dynamic GNNs to synchronize human activity graphs with urban land use changes, a dynamic heterogeneous GNN approach for integrating diverse human activity data to capture essential temporal dependencies, and a novel algorithm powered by causal inference to elucidate the primary factors influencing land use predictions at the community level, all of which contribute to a training process informed by the generated causal graph. Empirically validated on three real-world datasets, our model demonstrates a performance leap over state-of-the-art baselines, marking a pivotal step toward understanding and predicting the dynamics of urban land use.

CCS Concepts: • Information systems → Spatial-temporal systems.

Additional Key Words and Phrases: Land Use Change Prediction, Dynamic Graph, Graph Neural Networks.

ACM Reference Format:

Yu Fan, Xinjiang Lu[✉], Hao Liu, Pengfei Wang, Liang Liu, Huadong Ma, and Jingbo Zhou[✉]. 2018. Towards Predicting Urban Land Use Changes: A Dynamic Graph Alignment Perspective. 1, 1 (January 2018), 24 pages. <https://doi.org/XXXXXX.XXXXXXXX>

This work was done when the first author was an intern at Baidu Research. This work was supported in part by the National Natural Science Foundation of China under No.62225204, the A3 Foresight Program of NSFC under Grant 62061146002, the NSFC under No.62372059.

Authors' addresses: Yu Fan, fany@bupt.edu.cn, Beijing University of Posts and Telecommunications, Beijing, China; Xinjiang Lu[✉], luxinjiang@baidu.com, Baidu Research, Beijing, China; Hao Liu, liuh@ust.hk, Hong Kong University of Science and Technology, China; Pengfei Wang, wangpengfei@bupt.edu.cn, Beijing University of Posts and Telecommunications, Beijing, China; Liang Liu, liangliu@bupt.edu.cn, Beijing University of Posts and Telecommunications, Beijing, China; Huadong Ma, mhd@bupt.edu.cn, Beijing University of Posts and Telecommunications, Beijing, China; Jingbo Zhou[✉], zhoujingbo@baidu.com, Baidu Research, Beijing, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM XXXX-XXXX/2018/1-ART

<https://doi.org/XXXXXX.XXXXXXXX>

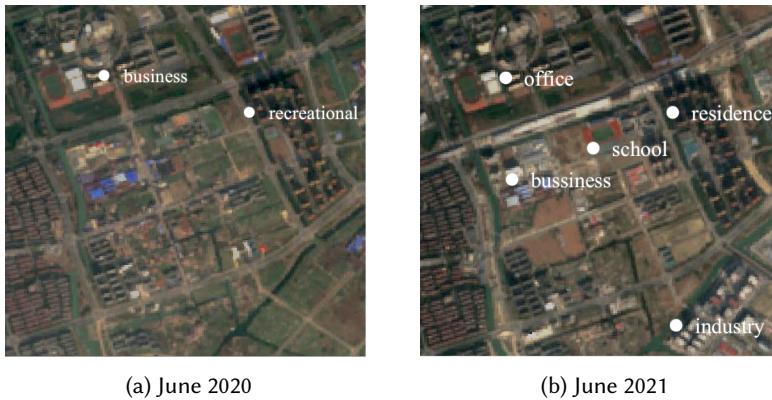


Fig. 1. An illustrative example of urban land use change.

1 INTRODUCTION

Predicting urban land use change is vital for anticipating future land use patterns and crafting sustainable development strategies. This task requires an analysis of land use transformations between consecutive time points to project these changes into the future, as highlighted by Eastman [9]. Understanding a city's historical land development trends is essential for city planners to formulate plans that support sustainable growth. Stanilov and Batty [32] underscored the significance of recognizing both rapid and gradual dynamics influencing urban land use alterations, advocating for an enhanced comprehension of the underlying mechanisms. Given its profound impact on human habitation and productivity, producing accurate land use predictions has become increasingly urgent.

Over recent decades, numerous studies have focused on modeling land use change, traditionally employing physical models like cellular automata [19, 42] to simulate and forecast land use dynamics. With the advent of deep learning, artificial neural network (ANN)-based methods [24, 34] have gained popularity. These methods typically approach land-use change prediction as a classification task, using single-label or multi-label classification to predict changes over long periods. However, this “coarse-grained” approach often falls short in predicting short-term, community-level land use changes, which are becoming increasingly common with the development of smart cities, as illustrated in Fig. 1. Additionally, the static nature of classified land use labels fails to capture the nuanced dynamics of urban land use changes. In response, our work redefines community land use as a real-valued portfolio, enabling us to model community-level land use changes as a regression problem, thereby capturing the more granular changes.

Urban development is recognized to be influenced by human activities, a relationship that has become clearer with the advent of big data over recent decades [13, 31, 43]. Traditional urban models use numerical functions to simulate human mobility, but these methods often fall short in predicting land use changes driven by mobility. In response, we propose a data-driven approach utilizing multifaceted, fine-grained human mobility data to enhance land use predictions. Integrating this intricate and heterogeneous human activity data into land use change models presents significant challenges, yet our approach aims to address these by harnessing the latest advancements in data analytics to improve the accuracy and efficiency of urban planning models.

In this work, we introduce a novel data-driven approach, dubbed ExHAGN, which features distinctive methodologies such as temporal skeletonization through dynamic graph clustering, heterogeneous dynamic graph aggregation, and causal inference to elucidate prediction results.

Specifically, we develop a temporal skeletonization mechanism that adaptively summarizes fast-evolving dynamic graphs, effectively capturing the informative dependencies found in numerous snapshots of citywide human activities. This process ensures that the dynamics of various human activities are seamlessly aligned with land use changes, maintaining the integrity of inherent “timing rules” of human behaviors. Furthermore, we introduce a heterogeneous dynamic graph neural network (HDGNN) that explores the diversity and temporal evolution of dynamic graphs. Additionally, our model includes an algorithm leveraging causal inference to explain predicted land use changes, with the generated causal graphs also aiding in the training process. Empirical validation on three real-world datasets demonstrates that ExHAGN significantly surpasses state-of-the-art baselines. We highlight our contributions as follows:

- We reformulate the land use change prediction as a regression problem and introduce a data-driven approach, ExHAGN, to address this task.
- We propose a temporal skeletonization mechanism in ExHAGN that automatically aligns varying human activities with target land use changes, respecting the inherent “timing rules” of daily human life.
- Leveraging mined high-quality human activity patterns, we design a graph neural network module in ExHAGN to extract informative properties for prediction, complemented by a causal inference module to elucidate key factors influencing the outcomes.
- Empirically, we demonstrate that ExHAGN significantly outperforms state-of-the-art (SOTA) baselines across three real-world datasets.

2 RELATED WORK

2.1 Land Use Change Modeling

2.1.1 Traditional Methods. Previous studies on urbanization and land use changes have primarily utilized satellite imagery and GIS data, broadly categorizing the approaches into static (operational) models and dynamic models like cellular automata. Operational models view the urban system as a network of market interactions, aiming to link land use with various driving forces and often simulate interactions among different industrial sectors. However, these models lack a solid theoretical foundation, which limits their ability to capture the underlying dynamics of urban development. In contrast, dynamic models, particularly cellular automata, have gained prominence for their effectiveness in modeling the dynamics of land use changes [2, 3, 36], though their simulation of macroscopic urban changes poses challenges for empirical validation.

2.1.2 Machine Learning Methods. With its universal approximation capabilities, machine learning has been widely applied to model land-use changes. SVM [46], classification trees [35], and KNN [25] have been employed for predictive tasks. However, the emergence of deep learning has introduced more sophisticated models for urban land use change prediction. Free from the constraints of stationary assumptions, deep learning approaches are adept at handling nonlinearities. Initially, DNN-based models became standard in this domain, exemplified by Tayyebi and Pijanowski [34]. Later, RNN-based and CNN-based models were introduced to better leverage correlated information, such as the model by Mu et al. [24] that combines RNNs with cellular automata for simulating land cover changes. Despite their effectiveness, these deep learning models still struggle with capturing the complex temporal dynamics of land use changes.

2.2 Clustering Algorithms

Clustering techniques can be categorized into three types: partition methods [16, 26], density-based methods [10, 29], and hierarchical methods [7, 33]. Partition-based methods, like K-means [16], segment data into distinct subsets where each data point belongs to one subset, optimizing the

centroids of these clusters. Density-based methods, exemplified by DBSCAN [10], identify clusters as dense regions of points, delineated by areas of lower density. Hierarchical methods structure data into a tree of nested clusters, with approaches like the one proposed by Szekely [33], which minimizes joint between-within cluster distances. Recently, the integration of deep learning for enhancing data representations has become prevalent in clustering [51], with learned representations being combined with traditional models for improved clustering outcomes [27, 39]. Despite these advancements, limitations persist: partition-based methods require a predetermined number of clusters, density-based approaches are sensitive to parameter settings, and hierarchical methods demand high computational resources. These constraints make them less adaptable to scenarios that involve varying rhythm time-series data.

2.3 Temporal Graph Neural Networks

Temporal graph neural networks can be categorized into static and dynamic methods. Static methods use graph neural networks (GNNs) to model static graphs and capture node features through convolutional neural networks (CNNs) or recurrent neural networks (RNNs). For example, Li et al. [20] integrate diffusion graph convolutions with RNNs, while Yu et al. [48] employ graph convolutions alongside temporal gated convolutions to capture temporal dependencies. More recently, dynamic methods have been developed for modeling evolving temporal graph scenarios. VGRNN [14] introduces a node embedding method for dynamic graphs using a variational autoencoder. Similarly, Rossi et al. [30] develop a generic inductive GNN that incorporates memory mechanisms for dynamic graphs. Despite their effectiveness, these approaches struggle to align the slow dynamics of urban land use changes with the rapid fluctuations in human mobility data, a critical issue in our research context.

2.4 Explanation for Deep Models

In the realm of deep learning interpretability, most existing methods are either feature- or sample-based. For instance, self-explanatory models like attention-based methods highlight the significance of inputs from the model’s perspective [1, 17, 23, 44], while model-independent interpreters examine changes in variables under certain assumptions [12, 40, 47, 49]. Unlike traditional land-use change prediction, which relies on expert-defined rules [3, 32], these methods do not provide a holistic understanding. Motivated by conventional approaches, we aim to automatically discover globally defined rules (i.e., concept-based explanations), as simply identifying key features or samples for specific predictions does not reflect the model’s overall behavior. Although causal inference models, which consider counterfactuals, have gained popularity for concept-based explanations [5, 45], applying these effectively in land use scenarios, where causal relationships may not be strictly defined, remains challenging.

3 PRELIMINARIES

This section formally introduces the urban land use change prediction problem and defines essential concepts to facilitate understanding of our proposed approach.

Let \mathcal{R} represent a set of regions within a city, defined as $\mathcal{R} = \{r_1, r_2, \dots, r_N\}$, and C denote the set of urban land use categories, $C = \{c_1, c_2, \dots, c_\Gamma\}$. In this study, we focus on 10 volatile and commonly observed urban land use categories, represented by C , which include: *business*, *commercial*, *administrative*, *residential*, *recreational*, *health care*, *public facility*, *municipal utility*, *industrial*, and *educational*.

DEFINITION 1 (LAND USE). *Given a region $r \in \mathcal{R}$, the land use of r is a portfolio vector $\mathbf{x} = (x_1, x_2, \dots, x_\Gamma)$ where x_y ($y \in [1, \Gamma]$) denotes the weight of y -th land use category within region r .*

Urban land use is characterized by the activities it accommodates and their spatial intensity. Historically, when human movement in cities was primarily on foot, urban activity nodes were densely agglomerated, resulting in compact urban forms with mixed land uses. Moreover, urbanization and subsequent land use changes are influenced by evolving social, economic, and technological factors. Let $\mathbf{X} \in \mathbb{R}^{N \times T}$ represent the matrix of land use in a city, encapsulating the notion that the land use \mathbf{X} evolves over time.

DEFINITION 2 (LAND USE CHANGE). *At time point t_m , the land use matrix of the region set \mathcal{R} is denoted as $\mathbf{X}(t_m)$. Then, the sequence $\mathcal{X} = \{\mathbf{X}(t_1), \mathbf{X}(t_2), \dots, \mathbf{X}(t_m), \dots, \mathbf{X}(t_M)\}$ denotes the observed land use change of the set of regions \mathcal{R} in a city where M is the length of the observation sequence.*

A multitude of functions, such as production, consumption, and distribution, are tied to diverse urban activities occurring at specific locations within the urban activity system. These include routine and occasional human behaviors related to passenger mobility, and crucial freight-related activities like manufacturing and distribution. The geographical dispersal of these activities drives the movements of both passengers and freight, supported by modern location-based services such as transportation systems and web services. In this context, we analyze the impact of these activities on land use changes through the dynamic graph.

DEFINITION 3 (DYNAMIC GRAPH). *A dynamic graph \mathcal{G} evolves across time with varying nodes and edges, which is denoted by $\mathcal{G} = \{G(s_1), G(s_2), \dots, G(s_l), \dots, G(s_L)\}$, and $G(s_l) = (\mathcal{V}(s_l), \mathcal{E}(s_l))$ is the snapshot of \mathcal{G} at time point s_l with $\mathcal{V}(s_l)$ and $\mathcal{E}(s_l)$ being the corresponding node and edge sets, respectively.*

In this work, to thoroughly explore the diverse and complex activities associated with land use change, we utilize multiple dynamic graphs, each representing different aspects of the urban activity system. We construct dynamic graphs from three distinct perspectives: movement, (travel) demand, and traffic, denoted by \mathcal{G}^μ , \mathcal{G}^δ , and \mathcal{G}^τ . It is important to note that these graphs evolve at a finer temporal resolution than land use changes, reflecting the more immediate and dynamic nature of these activities compared to the gradual shift in land use.

Essentially, given the land use change \mathcal{X} and related dynamic activity graphs \mathcal{G}^μ , \mathcal{G}^δ and \mathcal{G}^τ observed within the region set \mathcal{R} in a city, the objective of land use change prediction is to forecast future land use at subsequent time steps. Formally, we define this as:

PROBLEM 1 (LAND USE CHANGE PREDICTION). *Given land use change $\mathcal{X}(t_{1:M})$, and dynamic activity graphs \mathcal{G}^μ , \mathcal{G}^δ and \mathcal{G}^τ , the goal of land use change prediction is to forecast land use at a time step t_{M+1} via a learned model \mathcal{F} :*

$$\mathcal{F}(\mathcal{X}(t_{1:M}), \mathcal{G}^\mu, \mathcal{G}^\delta, \mathcal{G}^\tau) \rightarrow \widehat{\mathbf{X}}(t_{M+1}).$$

4 METHODOLOGY

As illustrated in Fig. 2, ExHAGN processes dynamic graphs \mathcal{G} and land use changes \mathcal{X} as inputs. It consists of four primary modules: activity graph segmentation, temporal skeletonization, DHGNN, and explanation.

- **Activity Graph Segmentation.** Activity graphs track daily human activity changes, but these patterns may not align with monthly land use changes due to weekly routines or events that shift peak days. To better capture urban rhythms, we use a data-driven approach to match activity graphs with land use timelines by identifying natural shifts in activity patterns. This module identifies the start and end points of activity graphs, ensuring alignment with land use changes. By segmenting these graphs, we synchronize human activity timing with urban development.

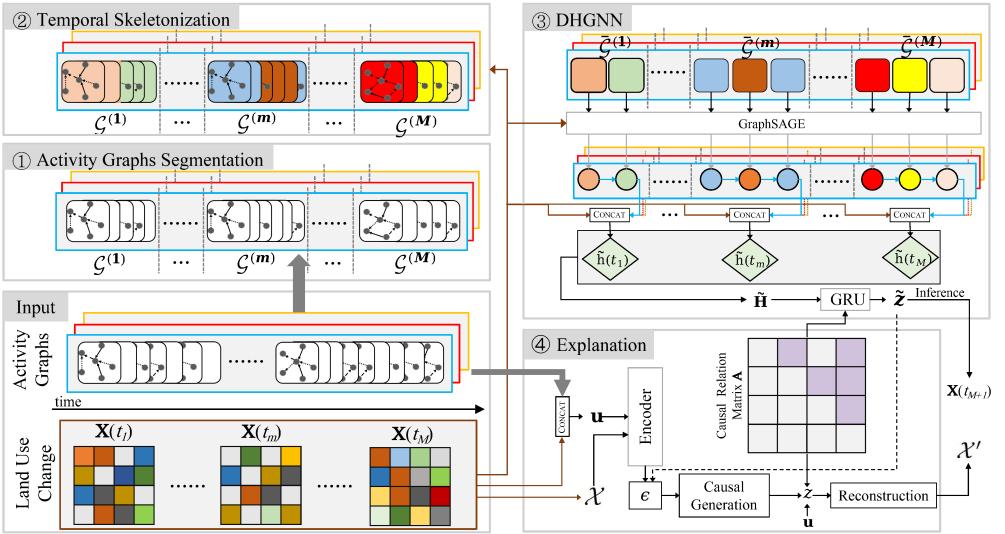


Fig. 2. The framework overview of ExHAGN.

- **Temporal Skeletonization.** While segmentation aligns activity graphs monthly, they retain daily granularity. Temporal Skeletonization clusters these segments to capture essential patterns, reduces noise and preserves key dependencies, enabling better integration with land use data. This module simplifies the complexity of the segmented activity graphs by condensing dynamic graphs. It captures essential temporal patterns while preserving the critical dependencies of citywide activities.
- **DHGNN.** Urban activities and land use changes are inherently diverse and complex, requiring a model that can handle this variety effectively. This module integrates the diverse dynamics of activity graphs and land use changes. It also uses a causal relation matrix from the explanation module to improve predictions by incorporating causal relationships.
- **Explanation.** Given that land use changes are influenced by a range of environmental and social factors, understanding the causal links between these elements is crucial for meaningful predictions. This module goes beyond simple feature correlations by constructing a causal relation matrix that identifies key relationships among observed factors. This matrix offers the drivers of land use dynamics and interpretable "rules", which enhances the model's predictive capacity by grounding it in the actual factors shaping urban evolution over time.

Overall, we first ensure that the timing of human behaviors is accurately represented in alignment with land use changes through segmentation and skeletonization. In the activity graph segmentation module, the dynamic graphs \mathcal{G} are segmented in a heuristic way, and these segmented graphs are subsequently clustered in the temporal skeletonization module using land use changes X as input as well. Next, a dynamic heterogeneous graph neural network (DHGNN) is employed to integrate the complex dynamics and heterogeneity present across the graphs and land use changes. Additionally, the explanation module generates causal explanations, constructing a causal relation matrix A . This matrix informs both the final predictions and the generation of explanations, creating a feedback loop that enhances model robustness. Each module is described in detail in the following subsections.

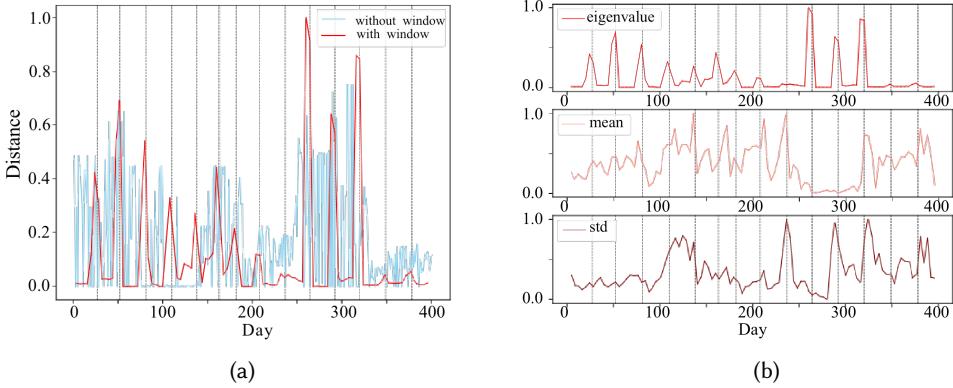


Fig. 3. Different strategies for segmenting time series of activity graphs. (a) Distance curves of STSAG w/ and w/o sliding window. (b) Encoding the activity graph time series with the adjacency matrix's eigen-, mean-, and std.-value.

4.1 Segmenting Activity Graphs

Before clustering activity graphs, we must identify the start and end points of corresponding activity graphs based on a land use change snapshot $X(t)$, such that the timing of human behaviors can be accurately represented in alignment with land use changes. This involves segmenting the series of activity graphs to align with the dynamics of land use change.

We develop a change point detection method to segment time-series of activity graph, namely STSAG, utilizing time-series change point detection algorithm [18, 37]. Following this, for a given land use change snapshot $X(t_m)$, the resulting segmented sequence is denoted as $\mathcal{G}^{(m)} = \{G(s_{l_m}), G(s_{l_m+1}), \dots, G(s_{l_{m+1}-1})\}$, where $\mathcal{G}^{(m)} \subseteq \mathcal{G}$ and $l_{m+1} - l_m > 0$ denotes the length of segmented sub-series.

To develop an efficient change point detection algorithm for \mathcal{G} , we begin by embedding each snapshot, $G(s_l)$, into a low-dimensional vector space. We analyze the eigenvector of its adjacency matrix, selecting the top- η eigenvalues as the representative vector for $G(s_l)$. A sliding window of size w , moving with a stride of ι time steps, is used to partition the original series. For each i -th window, we aggregate all snapshots within to create a column vector, $\mathbf{v}_i \in \mathbb{R}^{(\eta \cdot w) \times 1}$. This process results in the matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{\lfloor \frac{L-w+1}{\iota} \rfloor}]$, which encapsulates the input activity graphs.

Subsequently, we employ an auto-encoder to extract latent features from the input series \mathbf{V} . Specifically, the auto-encoder maps \mathbf{V} to a latent feature set f and reconstructs the original input, denoted by $\tilde{\mathbf{V}} = \text{AutoEncoder}(\mathbf{V})$, using a deep learning model. The reconstruction loss is

$$\mathcal{L}_s = \|\mathbf{V} - \tilde{\mathbf{V}}\|^2. \quad (1)$$

We now calculate the distance between features of consecutive time windows,

$$Dist_j = \|f_j - f_{j-1}\|_2, \quad j = 2, \dots, \lfloor \frac{L-w+1}{\iota} \rfloor. \quad (2)$$

By plotting a distance curve and identifying local maxima as change points, we can effectively segment the series of activity graphs. In Fig. 3, we compare the efficacy of various change point detection methods. The gray dashed lines indicate natural months, where land use changes monthly, showing that we achieve temporal alignment monthly with land use changes. The results indicate that the sliding window mechanism (Fig. 3a) provides more satisfactory change point detection

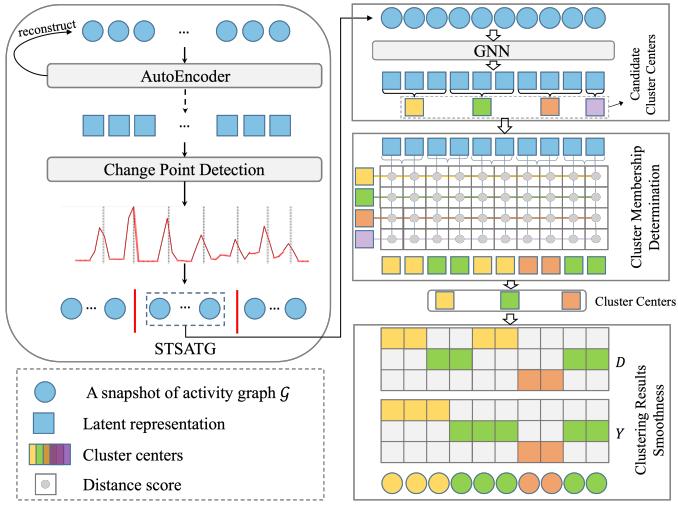


Fig. 4. The architecture of temporal skeletonization.

outcomes. Additionally, the series of activity graphs are more effectively partitioned when paired with land use change data using eigenvalue-based encoding, as demonstrated in Fig. 3b.

4.2 Temporal Skeletonization

After synchronizing the time resolution of evolving activity graphs with land use change dynamics, we further reduce the temporal cardinality of segmented activity graphs to emphasize temporal patterns. To capture the complex dimensional structure of activity graphs, we employ a temporal skeletonization algorithm facilitated by graph neural networks (GNNs). This integration into the deep learning framework enhances the utility of downstream deep models. Additionally, we account for the continuity in dynamic graphs. As depicted in Fig. 4, we initially establish cluster centers using a “top-down” partition strategy and then create clusters through a “divide-and-merge” strategy. Following the approach in [21], we apply sparsity-constrained optimization to finalize the clustering outcomes.

4.2.1 Initializing Candidate Cluster Centers. Utilizing GNN models, for each m -th segmented activity graph $\mathcal{G}^{(m)}$, we generate representations for each graph $G(s_l) \in \mathcal{G}^{(m)}$ as follows:

$$\mathbf{e}(l) = \text{MEAN}(\text{GNN}(G(s_l), \mathbf{X}(t_m))), \quad (3)$$

where $\mathbf{X}(t_m)$ represents the snapshot of land use change associated with $\mathcal{G}^{(m)}$, and $\text{MEAN}(\cdot)$ performs a row-wise 1-D average operation. “GNN” refers to any graph neural network model, e.g., GCN [41] or GraphSAGE [15]. The graphs of $\mathcal{G}^{(m)}$ are partitioned into clusters of fixed size u , and the averaged embeddings are used to represent each cluster center. Specifically, the k -th center, $\mathbf{c}(k)$, where $k = 1, \dots, K$ and $K = \lceil \frac{l_{m+1}-l_m}{u} \rceil$, is calculated as follows:

$$\mathbf{c}(k) = \frac{1}{u} \sum_{l=(k-1)u+1}^{k \cdot u} \mathbf{e}(l). \quad (4)$$

4.2.2 Cluster Membership Determination. Using the candidate cluster centers $C = \{c(k) | 1 \leq k \leq K\}$, we implement a “divide-and-merge” strategy to finalize the temporal clusters. Specifically, to

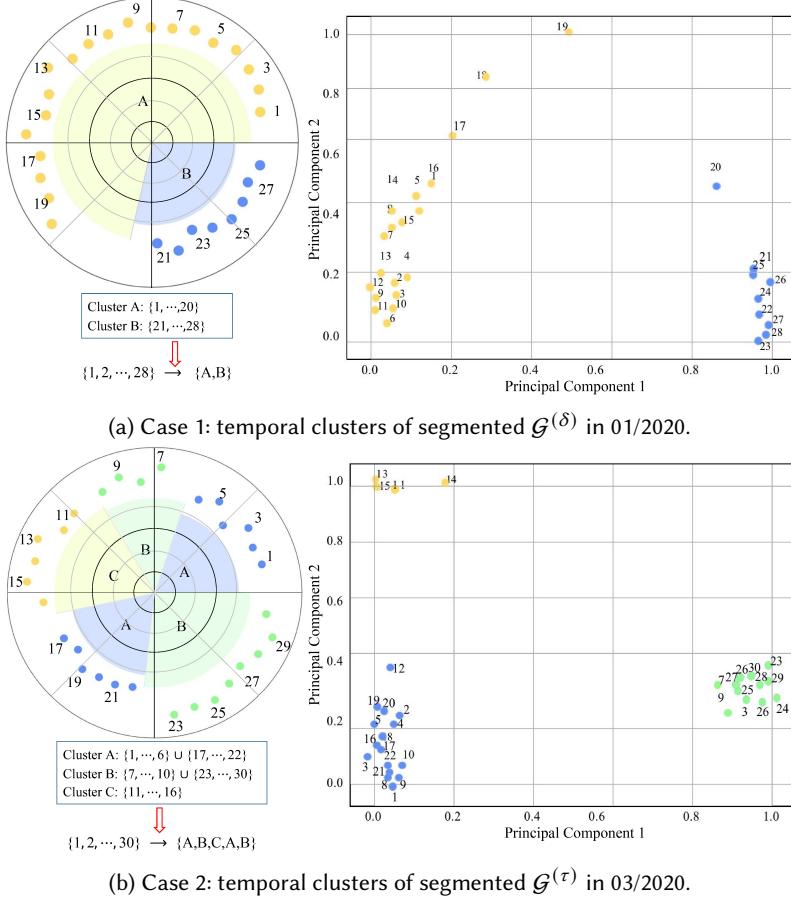


Fig. 5. Showcase the temporal clustering results. The polar plots of clustering results are provided (in the left column). Besides, the corresponding graph embeddings are also plotted with Principal Component Analysis (PCA).

ensure the smoothness of the resulting clusters, we partition the activity graphs in $\mathcal{G}^{(m)}$ into small groups, each containing v ($v > 1$ and $v < u$) consecutive graphs: $g(j) = \{G(s_l) | (j-1) \cdot v + 1 \leq l \leq j \cdot v\}$ ($j = 1, 2, \dots, J-1$) or $g(j) = \{G(s_l) | (J-1) \cdot v + 1 \leq l \leq l_m\}$ ($j = J$) where $J = \lceil \frac{l_{m+1}-l_m}{v} \rceil$. We then calculate the distance between $g(j)$ and $c(k)$. A group $g(j)$ is assigned to cluster center $c(k)$ based on the following criterion:

$$\min_k \sum_{G(s_l) \in g(j)} d_{lk}, \quad (5)$$

where $d_{lk} = \|\mathbf{e}(l) - \mathbf{c}(k)\|_2$. Cluster centers not assigned to any graphs are automatically disqualified and removed. For simplicity, we denote the resulting graph clusters in $\mathcal{G}^{(m)}$ as $C^{(m)} = \{c(k) | 1 \leq k \leq K\}$.

4.2.3 Smoothing Clustering Results. To mitigate the negative impacts of irregularities in activity graphs, we apply sparsity-constrained optimization to the temporal clusters $C^{(m)}$. We construct a distance matrix $D^{(m)} \in \mathbb{R}^{|G^{(m)}| \times K}$ to assess the pairwise distance between cluster centers and

activity graphs. This matrix helps partition $\mathcal{G}^{(m)}$ into $\geq K$ sub-sequences. For a smoother version of $D^{(m)}$, denoted by $Y^{(m)}$, we optimize the following sparsity-constrained alignment:

$$\begin{aligned} \mathcal{L}_{cl} = & \sum_{s=1}^{|G^{(m)}|} \sum_{k=1}^K Y_{sk}^{(m)} D_{sk}^{(m)}, \\ \text{s.t. } & \frac{1}{|G^{(m)}|} \sum_{s=1}^{|G^{(m)}|-1} \|Y_s^{(m)} - Y_{s+1}^{(m)}\|_1 \leq \rho, \\ & \sum_{k=1}^K Y_{sk}^{(m)} = 1, Y_{sk}^{(m)} \geq 0, \end{aligned} \quad (6)$$

where ρ is a tuning parameter for the smoothness of Y . We illustrate the temporal clustering results in Fig. 5. The clusters shown in the left column of Figs. 5a and 5b align well with the scatter plots of graph embeddings (right column of Figs. 5a and 5b), demonstrating the effectiveness of our temporal clustering approach.

After obtaining K temporal clusters for $\mathcal{G}^{(m)}$, the graph series is transformed into a smoother sequence. For example, as illustrated in Fig. 5a, a segmented sequence from the demand activity graph $\mathcal{G}^{(\delta)}$ can be encoded as A, \dots, A, B, \dots, B , while for the traffic activity graph $\mathcal{G}^{(\tau)}$ (Fig. 5b), the sequence is represented by three symbols $A, \dots, A, B, \dots, B, C, \dots, C, A, \dots, A, B, \dots, B$. These sequences can then be further compacted to A, B and A, B, C, A, B , respectively. Subsequently, for the graphs in $\mathcal{G}^{(m)}$ associated with $X(t_m)$, we merge consecutive graphs sharing the same label to form a skeletonized evolving activity graph, denoted as $\bar{\mathcal{G}}^{(m)} = \bar{G}(1), \bar{G}(2), \dots, \bar{G}(\Lambda_m)$, where Λ_m represents the length of the re-encoded sequence.

4.3 Dynamic Heterogeneous Graph Neural Network

In the m -th skeletonized dynamic graph ($\bar{\mathcal{G}}^{(m)}$), each “fused” graph, $\bar{G}(\lambda) \in \bar{\mathcal{G}}^{(m)}$, is formed by averaging the adjacency matrices of consecutive graphs that share the same cluster label, using the operation $\bar{G}(\lambda) = \text{MEAN}(G | G \in \mathcal{G}_\lambda^{(m)})$. Here, $\mathcal{G}_\lambda^{(m)}$ represents the λ -th group of graphs and $\text{MEAN}(\cdot)$ performs an element-wise average. These “summarized” graphs are then input into a downstream GNN model. To effectively capture the temporal dynamics and heterogeneity among different activities, we devise a dynamic heterogeneous graph neural network (DHGNN). Before modeling the intra-dynamics of $\bar{\mathcal{G}}^{(m)}$, we first need to establish the representation of each $\bar{G}(\lambda) \in \bar{\mathcal{G}}^{(m)}$.

4.3.1 Representing the “Summarized” Graph within $\bar{\mathcal{G}}^{(m)}$. To effectively represent $\bar{G}(\lambda)$ within $\bar{\mathcal{G}}^{(m)}$, we utilize GraphSAGE [15] to capture node and structural information. Specifically, each node v in $\mathcal{V}(\bar{G}(\lambda))$ is processed using the following aggregation mechanism:

$$\begin{aligned} \mathbf{z}_v^0 &= \mathbf{X}(t_m, v) = \mathbf{x}_v(t_m), \\ \mathbf{z}_{N(v)}^h &= \text{MEAN}(\mathbf{z}_v^{h-1}, v \in N(v)), \quad h \in \{1, \dots, H\}, \\ \mathbf{z}_v^h &= \sigma(\mathbf{w}^h \cdot (\mathbf{z}_v^{h-1} \| \mathbf{z}_{N(v)}^h \| \Phi_\lambda)), \end{aligned} \quad (7)$$

where $\mathbf{x}_v(t_m)$ is the land use attribute of node v at time t_m , σ represents a nonlinearity, \mathbf{w}^h denotes the trainable weights of the h -th layer, and $N(v)$ is the neighborhood of v . Additionally, to encode temporal patterns, we employ the positional embedding Φ_λ , defined for each cluster within the sequence as follows:

$$\Phi_\lambda = \mathbf{c}(k) + \text{PE}(\lambda), \quad (8)$$

where $\text{PE}(\lambda)$ is the positional encoding technique inspired by Transformer [38], and $\mathbf{c}(k)$ is the embedding of the cluster center to which $\bar{G}(\lambda)$ belongs. The resulting node embeddings generated by GraphSAGE are denoted as $\mathbf{z}_v(\lambda)$ for each node v in $\mathcal{V}(\bar{G}(\lambda))$.

4.3.2 Modeling Intra-Dynamics of Activity Graph. Given the short length of $\bar{G}^{(m)}$, we employ a short memory network that combines a recurrent structure with a multilayer perceptron (MLP). The structure of the MLP is $f(x) = \sigma(w_2 \cdot \sigma(w_1 \cdot x + b_1) + b_2)$, where w_1, w_2, b_1, b_2 are parameters, and σ represents the ReLU activation function. Specifically, for each node v in $\mathcal{V}(\bar{G}(\lambda))$, the recurrent structure is defined as follows:

$$\mathbf{h}_v(\lambda) = \text{MLP}(\mathbf{z}_v(\lambda), \mathbf{h}_v(\lambda - 1)), \quad \lambda = 2, \dots, \Lambda_m. \quad (9)$$

Here, we apply a minibatch setting [15] to learn the representation $\mathbf{h}_v(\lambda)$ ($\forall \lambda \in \{1, \dots, \Lambda_m\}$). Then we take the final embedding $\mathbf{h}_v(\Lambda_m)$ as the outcome embedding $\mathbf{h}_v(t_m)$, which is also treated as the activity representation corresponding to land use $\mathbf{x}_v(t_m)$. Moving forward, we will omit the subscript “ v ” when discussing nodes (or regions) in a citywide activity graph.

In this work, we examine citywide activities from three perspectives, i.e., human movement (\mathcal{G}^μ), travel demand (\mathcal{G}^δ), and traffic (\mathcal{G}^τ). Therefore, $\forall m \in \{1, \dots, M\}$, we derive $\mathbf{h}^\mu(t_m)$, $\mathbf{h}^\delta(t_m)$, and $\mathbf{h}^\tau(t_m)$ to encapsulate the complex activities influencing citywide land use changes.

4.3.3 Heterogeneous Activities and Long-Term Dynamics. For each region $r \in \mathcal{R}$, we integrate $\mathbf{h}^\mu(t_m)$, $\mathbf{h}^\delta(t_m)$, and $\mathbf{h}^\tau(t_m)$ with the land use change snapshot $\mathbf{x}(t_m)$ to create a new embedding, which can describe the portrait of regional land use and associated heterogeneous activities:

$$\tilde{\mathbf{h}}(t_m) = \mathbf{h}^{(\mu)}(t_m) \|\mathbf{h}^{(\delta)}(t_m) \|\mathbf{h}^{(\tau)}(t_m) \|\mathbf{x}(t_m).$$

Then, we employ a GRU [6] to capture long-term dynamics across various land use change snapshots:

$$\tilde{\mathbf{H}} = [\tilde{\mathbf{h}}(t_1), \tilde{\mathbf{h}}(t_2), \dots, \tilde{\mathbf{h}}(t_M)], \quad \tilde{\mathbf{z}} = \text{GRU}(\tilde{\mathbf{H}} \times \mathbf{A}), \quad (10)$$

where $\tilde{\mathbf{H}} \in \mathbb{R}^{M \times D}$ represents the sequence of embeddings, and $\mathbf{A} \in \mathbb{R}^{D \times D}$ is the correlation matrix, derived from causal inference (refer to Section 4.4). Finally, we predict the land use at time step t_{M+1} using the following equation:

$$\hat{\mathbf{x}}_{t_{M+1}} = \tilde{\mathbf{z}} \mathbf{w}^x, \quad (11)$$

where \mathbf{w}^x are the trainable parameters. We train the parameters of DHGNN in a minibatch manner, optimizing the loss defined as follows:

$$\mathcal{L}_p = \|\mathbf{x}(t_{M+1}) - \hat{\mathbf{x}}(t_{M+1})\|_2^2. \quad (12)$$

Overall, DHGNN captures both the heterogeneity of different citywide activities and their temporal dynamics. By representing activities as dynamic graphs and integrating temporal patterns, the model adapts to varying activity types and efficiently handles large-scale data, ensuring a nuanced understanding of urban activity patterns.

4.4 Causal-Inference-Guided Explanation

Land use change is correlated with environmental contexts, which has been validated by the community [3, 32]. To better understand the “rules” governing land-use change prediction, we move beyond traditional feature- or sample-based interpreters [47]. Instead, we introduce a disentangled representation learning module, inspired by CausalVAE [45], as depicted in the bottom right block of Fig. 2. Specifically, with the adjacency matrix \mathbf{A} representing causal relationships among concepts of interest within observations X , the causal generation layer is defined as follows:

$$\mathbf{z} = \mathbf{A}^T \mathbf{z} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \mathbf{I}), \quad (13)$$

Algorithm 1: Training procedure of ExHAGN.

Input: Land use change $\mathcal{X}(t_{1:M})$, temporal dynamic graphs \mathcal{G}

- 1 Initialize model parameter Θ , correlation matrix \mathbf{A} ;
- 2 Segments temporal dynamic graphs \mathcal{G} by STSAG;
- 3 **for** number of training iterations **do**
- 4 // Cluster the segmented graph;
- 5 $\bar{\mathcal{G}}^{(1:M)} = \text{TemporalCluster}(\mathcal{X}(t_{1:M}), \mathcal{G}^{(1:M)})$;
- 6 Calculate the constrain \mathcal{L}_{cl} in Eq. (6);
- 7 // Predict the land use change $\widehat{\mathbf{X}}(t_{M+1})$, $\widetilde{\mathbf{z}} = \text{DHGNN}(\mathcal{X}(t_{1:M}), \bar{\mathcal{G}}^{(1:M)}, \mathbf{A})$;
- 8 Calculate the prediction loss \mathcal{L}_p in Eq. (12);
- 9 // Generate causal-inference-guided explanation $\mathbf{A} = \text{Causal}(\widetilde{\mathbf{z}}, \mathcal{X}(t_{1:M}), \mathcal{G})$;
- 10 Calculate the causal loss \mathcal{L}_{ca} in Eq. (15);
- 11 Update the whole framework by Eq. (16);
- 12 **end**

where the Gaussian variable ε denotes independent exogenous factors and $\mathbf{z} \in \mathbb{R}^D$ is the structured causal representation of D concepts. Since the concepts are causally structured by a directed acyclic graph (DAG), \mathbf{A} can be permuted into an upper triangular matrix. Besides, exogenous factors ε can be drawn from the distribution of outcome latent variable $\widetilde{\mathbf{z}}$ (generated with Eq. (10)) as follows:

$$\varepsilon \sim p(\widetilde{\mathbf{z}}; \mathbf{w}^z), \quad (14)$$

where \mathbf{w}^z are the learnable parameters.

4.4.1 Supervision Signals. As suggested by [45], supervisory signals help learn the structural causal representation, such that the identifiability issues caused by unsupervised learning [22] can be avoided. We take the labels of land use categories (C) plus activity graph types (μ , δ , and τ) as additional signals (denoted by \mathbf{u}), which also correspond to the concepts of interest to be disentangled. We build the supervision signals as follows:

$$\mathbf{u} = \text{AVG}_{1D}(\mathcal{X}) \parallel \text{AVG}_{2D}(\mathcal{G}^{(\mu)}) \parallel \text{AVG}_{2D}(\mathcal{G}^{(\delta)}) \parallel \text{AVG}_{2D}(\mathcal{G}^{(\tau)}),$$

where $\text{AVG}_{1D}(\cdot)$ and $\text{AVG}_{2D}(\cdot)$ denote 1-D and 2-D average pooling operations, respectively.

4.4.2 Learning Strategy for Causal Representation. We utilize the optimization strategy from CausalVAE [45] to concurrently learn the causal representation \mathbf{z} and the causal structure \mathbf{A} . Specifically, we use the following loss function as the objective for causal representation learning¹:

$$\mathcal{L}_{ca} = -\text{ELBO} + l_u + l_m + H(\mathbf{A}), \quad (15)$$

where ELBO denotes the reformulated evidence lower bound introduced by [45], l_u and l_m are two constraints that ensure \mathbf{A} can describe the causal relations among labels. Besides, $H(\mathbf{A})$ denotes the constraint that controls the “DAGness” of \mathbf{A} .

4.5 Optimization

We outline the training process in Algorithm 1. Initially, we train the change point detection method STSAG by optimizing Eq. (1). Then, with Eqs. (6), (12) and (15), we jointly optimize the following objective function:

$$\mathcal{L}_{\text{ExHAGN}} = \mathcal{L}_{cl} + \mathcal{L}_p + \mathcal{L}_{ca} + \|\Theta\|_2. \quad (16)$$

¹For more technical details on causal representation learning, please see [45].

Table 1. Statistical descriptions of the datasets.

Data		Statistics	Beijing	Shanghai	Chengdu
Land Use	#Regions	39,942	42,856	29,272	
	#POIs per Month	1,273K $\pm 137K$	1,578K $\pm 170K$	1,063K $\pm 115K$	
Activity	Movement	#Records per Day #Daily Records per User	18,318K $\pm 5,713K$ 2.44 ± 0.76	17,586K $\pm 4,106K$ 2.43 ± 0.57	15,011K $\pm 4,008K$ 3.54 ± 0.95
	Travel Demand	#Records per Day #Daily Records per User	3,213K $\pm 1,052K$ 0.43 ± 0.14	2,962K $\pm 834K$ 0.41 ± 0.12	566K $\pm 306K$ 0.13 ± 0.07
	Traffic	#Records per Day #Daily Records per User	979K $\pm 590K$ 0.13 ± 0.07	1,277K $\pm 355K$ 0.17 ± 0.05	1,746K $\pm 579K$ 0.42 ± 0.14

Here, we adopt L_2 regularization to prevent overfitting.

5 EXPERIMENT

5.1 Experimental Setup

Datasets. We conducted empirical studies using datasets collected from Beijing, Shanghai, and Chengdu in China, detailed in Table 1. Initially, we partitioned these cities into non-overlapping regions based on road networks [50]. Subsequently, POIs were collected through a third-party location-based service (LBS) app from April 2019 to June 2020. We then constructed land-use portfolios for the regions over the same period using the TF-IDF algorithm. Simultaneously, from April 2019 to June 2020, we collected user behavioral data via the same LBS App to develop three types of activity graphs: \mathcal{G}^μ , \mathcal{G}^δ , and \mathcal{G}^τ . Specifically, we recorded users' check-ins to analyze human movements, forming \mathcal{G}^μ . Searches within the app indicating users' travel intent were used to create the travel demand activity graph \mathcal{G}^δ . Additionally, data on users' origins, destinations, and transportation modes were gathered to construct the traffic activity graph, \mathcal{G}^τ . We divided all datasets into training, validation, and testing sets, allocating 80%, 10%, and 10%, respectively, by region. The training sets were initially z-normalized, and this normalization setting was subsequently applied to the validation and testing sets.

Dataset Preprocessing. In this section, we will describe the land use change data generation process. We have the following data:

- **Block:** A region is divided into blocks with road network information.
- **Land Use Feature:** We define land use features based on the “Code for Classification of Urban and Rural Land Use and Planning Standards of Development Land”². We process and select the ten most active types for our dataset: business, commercial, administrative, residential, recreational, health care, public facility, municipal utility, industrial, and educational.
- **POI:** POI data is aligned with land use features as per national standards¹. The number of POIs and the area of POIs (AOI) corresponding to each feature are key inputs for feature value generation.

We then define the numerical expression of land use features $\mathbf{x} = [x_1, x_2, \dots, x_D]$ where each x_i is calculated as:

$$x_i = a_i b_i \quad (17)$$

Here, a_i and b_i represent the area and frequency weights of each feature, respectively.

For area weight, we define the area of property a_i as $\frac{p_i}{p}$, where p is the region's total area. If all area proportions are less than the hyperparameter α , the weight is set to 1. If the area proportions

²https://www.mohurd.gov.cn/gongkai/fdzdgknr/zqyj/201805/20180522_236162.html

greater than and less than α coexist within the region, the weights are assigned as follows:

$$a^i = \begin{cases} \frac{1}{n} \left(1 - \frac{p_a}{p} \right) & \frac{p_i}{p} < \alpha \\ \frac{p_i}{p} & \frac{p_i}{p} > \alpha \end{cases} \quad (18)$$

where $p_a = \sum_{\frac{p_i}{p} > \alpha} p_i$ represents the total area of properties exceeding α , and n is the count of properties with area proportions smaller than α . We empirically set $\alpha = 0.05$.

For frequency weight, we use the term frequency-inverse document frequency (TF-IDF) model:

$$\text{TF-IDF}(m_i, m, \mathcal{M}) = \text{TF}(m_i, m) \times \text{IDF}(m_i, \mathcal{M}) \quad (19)$$

where m_i is the number of POIs corresponding to feature i in the region, m is the total number of POIs across all features in the region, and \mathcal{M} is the total number of all POIs across all regions. The term frequency $\text{TF}(m_i, m)$ is calculated as $\frac{m_i}{m}$. The inverse document frequency (IDF) is given by:

$$\text{IDF}(m, \mathcal{M}) = \frac{|\mathcal{M}|}{|m \in \mathcal{M} : m_i \in m|} \quad (20)$$

where $|m \in \mathcal{M} : m_i \in m|$ is the count of regions where the feature m_i appears.

Baselines. We evaluated ExHAGN against six baseline methods: two statistical approaches (ARIMA and LightGBM), two state-of-the-art spatiotemporal deep learning methods (STGCN and DCRNN), and two dynamic GNN methods (VGRNN and TGN). Each method is briefly described as follows:

- **ARIMA** [4] is a popular time series forecasting model built upon moving average and auto-regression.
- **LightGBM** [11] is a well-known approach for regression, and it is built on top of ensemble methods.
- **STGCN** [48] integrates ChebNet [8] into convolution networks with a predefined graph. We implemented it by aggregating the dynamic graphs into a static graph.
- **DCRNN** [20] merges diffusion graph convolutions with RNNs for spatiotemporal prediction. We apply the same predefined graph as used in STGCN.
- **VGRNN** [14] integrates graph convolutional networks with RNNs within a VAE framework to model dynamic graphs. In our implementation, we use land use changes as node features in dynamic graphs, predicting these changes from the output embeddings.
- **TGN** [30] is a dynamic graph model that encodes interactions and introduces a memory mechanism to capture temporal dependencies. We use this model to predict land use changes based on the output embeddings.

Evaluation Metrics. In this work, we represent a region's land use as a portfolio, quantified as a vector of real numbers. We use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as our evaluation metrics, distinguishing between micro- and macro-metrics as follows:

$$\begin{aligned} \text{MAE}_{micro} &= \frac{\sum_1^\Gamma \sum_{n=1}^N |x_{\gamma,n} - \hat{x}_{\gamma,n}|}{N \times \Gamma}, \\ \text{RMSE}_{micro} &= \sqrt{\frac{\sum_1^\Gamma \sum_{n=1}^N (x_{\gamma,n} - \hat{x}_{\gamma,n})^2}{N \times \Gamma}}, \\ \text{MAE}_{macro} &= \frac{1}{\Gamma \sum_1^\Gamma} \frac{\sum_{n=1}^N |x_{\gamma,n} - \hat{x}_{\gamma,n}|}{N}, \\ \text{RMSE}_{macro} &= \frac{1}{\Gamma \sum_1^\Gamma} \sqrt{\frac{\sum_{n=1}^N (x_{\gamma,n} - \hat{x}_{\gamma,n})^2}{N}}. \end{aligned}$$

Table 2. Performance comparison of our model against baselines, highlighting the best results in bold in each row. The last column details the improvement of our model over the best-performing baseline.

Dataset	Metric	ARIMA	LightGBM	STGCN	DCRNN	VGRNN	TGN	ExHAGN	Improvement
Beijing	MAE _{micro}	8.03E-3	7.47E-3	7.20E-3	7.14E-3	7.45E-3	6.96E-3	6.31E-3	9.34%
	RMSE _{micro}	6.89E-2	6.15E-2	4.94E-2	4.65E-2	5.34E-2	4.19E-2	3.84E-2	8.30%
	MAE _{macro}	1.03E-2	8.16E-3	8.00E-3	7.52E-3	7.74E-3	7.01E-3	6.28E-3	10.34%
	RMSE _{macro}	6.51E-2	5.23E-2	4.27E-2	4.01E-2	3.79E-2	3.20E-2	2.86E-2	10.56%
Shanghai	MAE _{micro}	7.52E-3	7.27E-3	6.83E-3	6.98E-3	7.24E-3	6.44E-3	6.12E-3	5.04%
	RMSE _{micro}	8.39E-2	7.82E-2	6.53E-2	6.49E-2	7.27E-2	5.60E-2	5.23E-2	6.66%
	MAE _{macro}	7.11E-3	6.74E-3	6.05E-3	5.95E-3	5.25E-3	4.58E-3	4.13E-3	9.96%
	RMSE _{macro}	4.68E-2	4.07E-2	3.54E-2	3.75E-2	3.92E-2	3.20E-2	2.89E-2	9.72%
Chengdu	MAE _{micro}	7.61E-3	6.57E-3	5.37E-3	5.32E-3	5.55E-3	5.11E-3	4.70E-3	7.92%
	RMSE _{micro}	6.41E-2	5.72E-2	4.38E-2	4.12E-2	4.72E-2	3.73E-2	3.40E-2	8.79%
	MAE _{macro}	6.06E-3	5.31E-3	4.51E-3	4.79E-3	4.99E-3	4.40E-3	3.98E-3	9.37%
	RMSE _{macro}	4.61E-2	3.23E-2	2.77E-2	2.99E-2	2.89E-2	2.57E-2	2.30E-2	10.82%

Implementation Details. We adopt an Adam optimizer with an initial learning rate of 0.001 and a batch size of 512, incorporating early stopping within 300 epochs to optimize training. For the STSAG module, we represent each graph using the top 20 eigenvalues ($\eta = 20$). In Beijing, the sliding window size w and stride t are set to (10, 5), (8, 6), (10, 4) for movement, traffic, and demand graphs, respectively. In Shanghai, these parameters are (10, 6), (6, 6), (14, 8), and in Chengdu, they are (8, 6), (16, 6), (10, 5). The temporal skeletonization module uses partition sizes of $u = 5, 5$, and 4 for Beijing, Shanghai, and Chengdu, respectively, with a uniform group size $v = 3$ across all three datasets. For the DHGNN, we use 2 aggregator layers with a sample size of 10 in the GraphSAGE aggregator. In the causal-inference-guided explanation module, we define the number of concepts $D = 13$, encompassing 10 land use categories and 3 types of activities. Baseline implementations utilize source code provided by the original authors.

5.2 Land Use Change Prediction Results

Table 2 displays the superior performance of ExHAGN compared to baseline methods across three datasets. ExHAGN consistently outperforms all alternatives, reducing errors by 5.04%-10.56% across all evaluation metrics compared to the second-best results.

Generally, deep learning methods outperform statistical ones due to their ability to model nonlinearities. Among the deep methods, VGRNN performs the poorest, likely because it emphasizes graph variability, which is more suited for link prediction rather than land use change. While STGCN and DCRNN effectively model spatial and temporal dependencies for forecasting, they are still outperformed by TGN, which enhances performance by integrating dynamic graphs with memory mechanisms. However, TGN can distort the alignment of temporal patterns in dynamic graphs and land use changes. In contrast, ExHAGN aligns dynamic graphs with land use changes, preserving inherent temporal patterns, and utilizes a causal graph to enhance prediction accuracy. Overall, by synthesizing dynamic graph modeling, temporal pattern preservation, and causal inference, ExHAGN offers a more robust and effective solution compared to competing methods.

5.2.1 Category-Specific Prediction Results. This section additionally explores the prediction of category-specific land use changes. We start by comparing ExHAGN to baseline methods and then conduct a robustness analysis. We focus on three key land use categories: commercial, industrial, and residential, based on the classification by Rodrigue et al. [28]. They categorized land use into core, central, and peripheral activities. Core activities include tertiary and quaternary sectors such as business, commercial, and recreational. Central activities encompass production and distribution

Table 3. Category-specific performance of ExHAGN on the Beijing dataset.

Land Use	Metric	ARIMA	lightGBM	STGCN	DCRNN	VGRNN	TGN	ExHAGN
Commercial	MAE _{micro}	4.21E-3	2.67E-3	4.76E-3	4.72E-3	4.43E-3	4.14E-3	2.52E-3
	RMSE _{micro}	2.18E-2	1.81E-2	2.16E-2	1.99E-2	2.39E-2	1.88E-2	1.79E-2
Residential	MAE _{micro}	9.56E-3	7.13E-3	9.96E-3	9.75E-3	9.74E-3	8.66E-3	4.86E-3
	RMSE _{micro}	3.04E-2	2.53E-2	2.72E-2	2.50E-2	3.07E-2	2.37E-2	2.17E-2
Industrial	MAE _{micro}	3.56E-3	2.97E-3	3.91E-3	3.88E-3	3.69E-3	3.40E-3	2.16E-3
	RMSE _{micro}	2.31E-2	1.92E-2	1.27E-2	1.16E-2	1.40E-2	1.10E-2	9.65E-3

Table 4. ExHAGN’s performance in three proportions on the Beijing dataset.

Land Use	Metric	Low	Medium	High
Commercial	MAE _{micro}	3.04E-3	3.23E-3	3.33E-3
	RMSE _{micro}	1.90E-2	2.04E-3	1.76E-2
Residential	MAE _{micro}	3.82E-3	3.93E-3	4.08E-3
	RMSE _{micro}	2.08E-2	1.85E-2	2.15E-2
Industrial	MAE _{micro}	2.20E-3	2.11E-3	2.29E-3
	RMSE _{micro}	6.79E-3	6.64E-3	6.12E-3

Table 5. Performance comparison of ExHAGN and its three variants w.r.t. different activities on three datasets, highlighting the best performance in bold.

Dataset	Beijing		Shanghai		Chengdu	
	Metric	MAE _{micro}	RMSE _{micro}	MAE _{micro}	RMSE _{micro}	MAE _{micro}
w/o movement	6.87E-3	4.07E-2	6.66E-3	5.54E-2	5.12E-3	3.60E-2
w/o demand	6.84E-3	3.97E-2	6.63E-3	5.41E-2	5.10E-3	3.52E-2
w/o traffic	6.58E-3	3.89E-2	6.38E-3	5.30E-2	4.93E-3	3.44E-2
ExHAGN	6.31E-3	3.84E-2	6.12E-3	5.23E-2	4.70E-3	3.40E-2

functions, like industrial and administrative roles. Peripheral activities typically involve residential areas or cater to local needs, including healthcare, public facilities, municipal utilities, and education.

Table 3 displays the performance of ExHAGN and baselines in category-specific land use prediction. ExHAGN consistently outperforms the baselines across all metrics on the Beijing dataset. To assess the robustness of our model, we further evaluate its performance across land use categories at different frequency levels: high, middle, and low. We categorize each land use based on its occurrence frequency, and the comparison results are detailed in Table 4. The results demonstrate minimal variation across frequency levels in ExHAGN, indicating its ability to reliably predict land use changes regardless of the category’s frequency. This robustness stems from both the temporal skeletonization module, which captures essential patterns across different frequencies, and the causal explanation module, which extracts stable rules to enhance prediction consistency. Together, they ensure ExHAGN’s adaptability to diverse scenarios.

5.3 Ablation Study

5.3.1 Impacts of Different Activities. We evaluate the impact of different temporal dynamic graphs on land use change prediction by comparing several variants of ExHAGN. Each variant is created by removing one type of temporal dynamic graph from ExHAGN.

Table 6. Performance comparison of ExHAGN and its variants w.r.t. different modules on three datasets. The best-performing one is highlighted in bold.

Dataset	Beijing		Shanghai		Chengdu	
Metric	MAE _{micro}	RMSE _{micro}	MAE _{micro}	RMSE _{micro}	MAE _{micro}	RMSE _{micro}
w/o segment	6.62E-3	3.91E-2	6.42E-3	5.41E-2	4.91E-3	3.43E-2
w/o cluster	6.93E-3	4.10E-2	6.67E-3	5.54E-2	5.13E-3	3.71E-2
w/o causal	6.75E-3	4.02E-2	6.51E-3	5.46E-2	4.95E-3	3.61E-2
ExHAGN	6.31E-3	3.84E-2	6.12E-3	5.23E-2	4.70E-3	3.40E-2

The performance of ExHAGN and its three variants across all datasets is presented in Table 5. The variant without movement data consistently underperforms across all metrics and datasets, highlighting human movement as a critical factor in predicting land use changes. This may be due to human movement directly reflecting behavioral patterns and preferences more effectively than the other activities. Human movement routes are essential for accurate predictions, whereas travel demand and traffic data offer more indirect insights, such as popular destinations. Overall, the complete model, which integrates all three dynamic graphs, significantly outperforms its variants on all evaluation measures.

5.3.2 Impacts of Different Modules. To assess the impact of activity graph segmentation (STSAG), temporal skeletonization, and causal-inference-guided explanation in our model, we compare ExHAGN against three variants:

- **w/o segment** omits STSAG, aligning activity graphs with land use changes using natural monthly segmentation.
- **w/o cluster** eliminates the temporal skeletonization mechanism, treating segmented graphs as a whole rather than as clustered evolving activity graphs.
- **w/o causal** omits the correlation matrix \mathbf{A} from Eq. (10).

The comparison results in Table 6 show that *w/o cluster* has the poorest performance among the variants, highlighting the significance of the temporal skeletonization module in integrating graphs with temporal pattern information to enhance downstream deep models. The temporal skeletonization module plays a key role by condensing the temporal complexity of dynamic graphs and extracting critical temporal dependencies. Removing this module limits the model's ability to capture essential timing patterns, resulting in poorer performance. Additionally, *w/o causal* underperforms *w/o segment*, likely due to the correlation matrix in Eq. (10) enriching the model by integrating both land use change and activity graph data, which proves more advantageous than mere segmentation. Overall, ExHAGN consistently surpasses all three variants across all evaluation metrics by effectively combining these three modules.

5.4 Model Analysis

5.4.1 Segmentation Convergence in STSAG.. In ExHAGN, segmentation results are dynamically adjusted throughout the STSAG training process to ideally converge before entering the subsequent temporal skeletonization module. STSAG utilizes a distance curve to identify change points, and segmentation is deemed to have converged when this curve stabilizes.

To quantify convergence, we inspect the distance curve through Gaussian distribution fitting. Specifically, each peak on the distance curve is fitted using the density function of a Gaussian distribution:

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right).$$

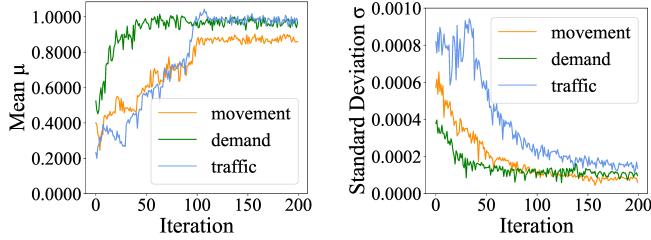


Fig. 6. Convergence of the change point detection w.r.t. different activity graphs.

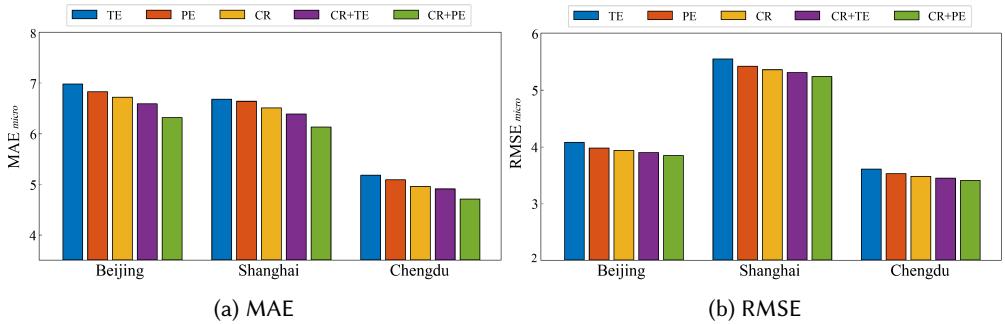


Fig. 7. Comparison of different temporal encodings.

Fig. 6 displays the convergence of μ and σ over iterations in the Beijing dataset. The stabilization of these values after a few iterations indicates the effectiveness of the temporal segmentation approach in ExHAGN, demonstrating its capability to achieve convergence.

5.4.2 Temporal Encoding. We introduce Eq. (8) to encode temporal patterns. We assess various encoding methods to demonstrate their efficacy:

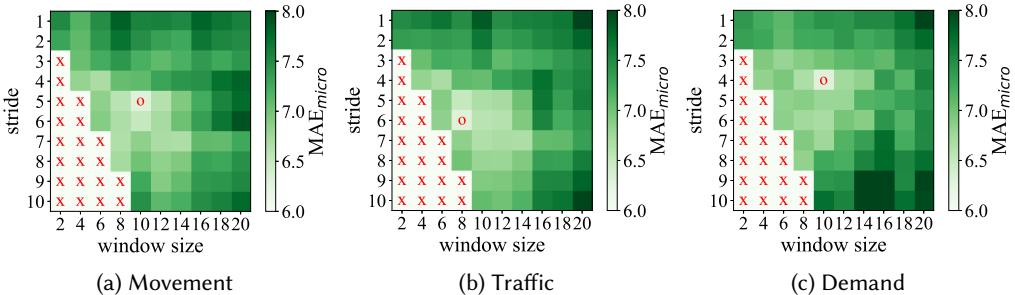
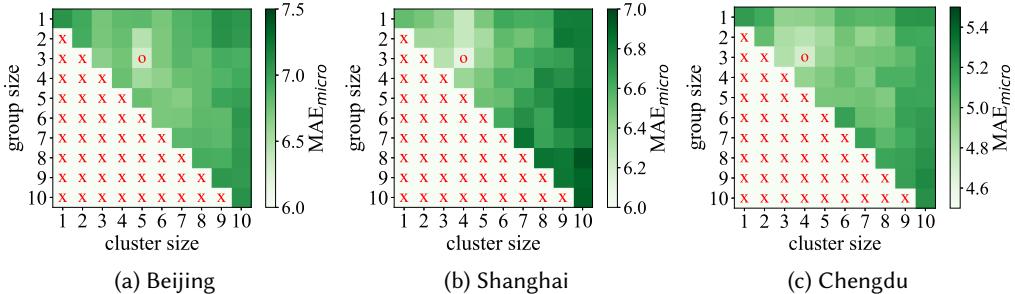
- **CR+PE** integrates cluster center representation with positional encoding as described in Eq. (8).
- **CR+TE** replaces the positional encoding with time encoding proposed in [30].
- **CR** solely relies on cluster center representation.
- **PE** exclusively utilizes positional encoding.
- **TE** exclusively employs time encoding.

As shown in Fig. 7, PE outperforms TE, likely due to the role of the memory mechanism in our model. TE encodes time-based information, which overlaps with the temporal dependencies already captured by the memory mechanism, leading to redundancy. In contrast, PE encodes positional information, complementing the model by adding spatial or sequence context without conflicting with the memory-based temporal learning. This synergy explains why PE performs better than TE in our model, which already handles temporal dynamics efficiently. Additionally, the combination of CR and PE proves optimal, as CR captures essential temporal patterns, further enhancing the positional encoding.

5.4.3 Dynamics Modeling. In the DHGNN module, we adopt the short memory network for intra-dynamics modeling in Eq. (9) and a GRU for long-term dynamics modeling in Eq. (10). To validate their effectiveness, we devised two respective variants for comparison.

Table 7. Comparing ExHAGN against its two variants. The best scores are in bold.

Dataset	Metric	ExHAGN _g	ExHAGN _m	ExHAGN
Beijing	MAE _{micro}	6.62E-3	6.58E-3	6.31E-3
	RMSE _{micro}	3.94E-2	3.89E-2	3.84E-2
Shanghai	MAE _{micro}	6.42E-3	6.38E-3	6.12E-3
	RMSE _{micro}	5.37E-2	5.33E-2	5.23E-2
Chengdu	MAE _{micro}	5.05E-3	4.87E-3	4.70E-3
	RMSE _{micro}	3.48E-2	3.46E-2	3.40E-2

Fig. 8. The grid search of parameters (window size w and stride i) in the Beijing dataset.Fig. 9. The grid search of parameters includes cluster size u and group size v .

- ExHAGN_m substitutes the GRU with a short memory network in Eq. (10).
- ExHAGN_g replaces the short memory network with a GRU in Eq. (9).

The performance of ExHAGN and its two variants across three datasets is presented in Table 7. Results reveal that ExHAGN_m outperforms ExHAGN_g. Furthermore, ExHAGN surpasses both variants on all evaluation metrics, underscoring the effectiveness of using a short memory network for intra-dynamics modeling, which better handles short temporal dependencies, and a GRU for superior long-term dynamic modeling in ExHAGN.

5.4.4 Hyperparameters. We inspect ExHAGN under various hyperparameter settings to optimize selection. For the segmentation module, we explore two hyperparameters: window size w and stride i . Conducting a grid search within specified ranges, we ensure that $w \geq i$ enables valid change point detection. If $w < i$, gaps between segments may arise as the analysis window moves across the data. This can lead to missed potential change points and segments that do not fully overlap

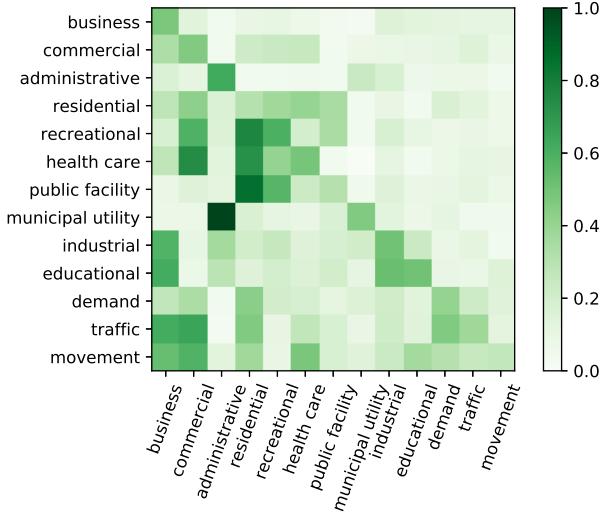


Fig. 10. The correlation matrix \mathbf{A} (Beijing dataset).

with change point areas. The outcomes, depicted in Fig. 8, illustrate that optimal hyperparameters for movement, traffic, and demand activity graphs are $w = 10, 8, 10$ and $t = 5, 6, 4$, respectively.

In the temporal skeletonization module, we examine the cluster size u for generating initial centers and group size v for the “divide-and-merge” strategy, ensuring $u \leq v$. Fig. 9 presents the grid search results across three datasets, with MAE_{micro} plotted against u and v . Optimal settings for Beijing, Shanghai, and Chengdu are found to be $u = 5, 4, 4$ and $v = 3, 3, 3$, respectively. This grid search helps pinpoint the best hyperparameters to enhance model performance across metrics.

5.4.5 Correlation Matrix A. The correlation matrix \mathbf{A} for Beijing is depicted in Fig. 10, encompassing ten land usages and three activity graphs (demand, traffic, movement). We configure \mathbf{A} as a lower triangular matrix, reflecting the directional influence among usages and activities.

Within the matrix, the diagonal indicates the auto-correlation of land use features. Notably, commercial land exhibits strong correlations with recreational and healthcare land, aligning with urban agglomeration laws [32]. Regarding the interaction between land use features and activity graphs, human activities significantly influence commercial land use but have a minimal effect on administrative land, which generally receives less consideration in its spatial planning. Additionally, the demand and traffic graphs show a high correlation due to their related conceptual definitions.

Overall, our model effectively uncovers the underlying rules influenced by environmental contexts in land use, as evidenced by the patterns observed in \mathbf{A} . This helps in understanding the dynamics of land use change.

5.5 Case Study

We further analyze two cases from the Beijing dataset, focusing on the predictive accuracy for specific regions and the influence of activity graphs, which capture human activities over time. For each case, we identify key regions associated with the target based on interactions with neighboring areas.

Case 1 involves a region in Chaoyang, Beijing. Fig. 11a depicts a successful prediction of a significant increase in business land use. Fig. 11b identifies crucial neighboring regions based on last month’s human activities; these areas primarily comprise commercial districts, residences, and

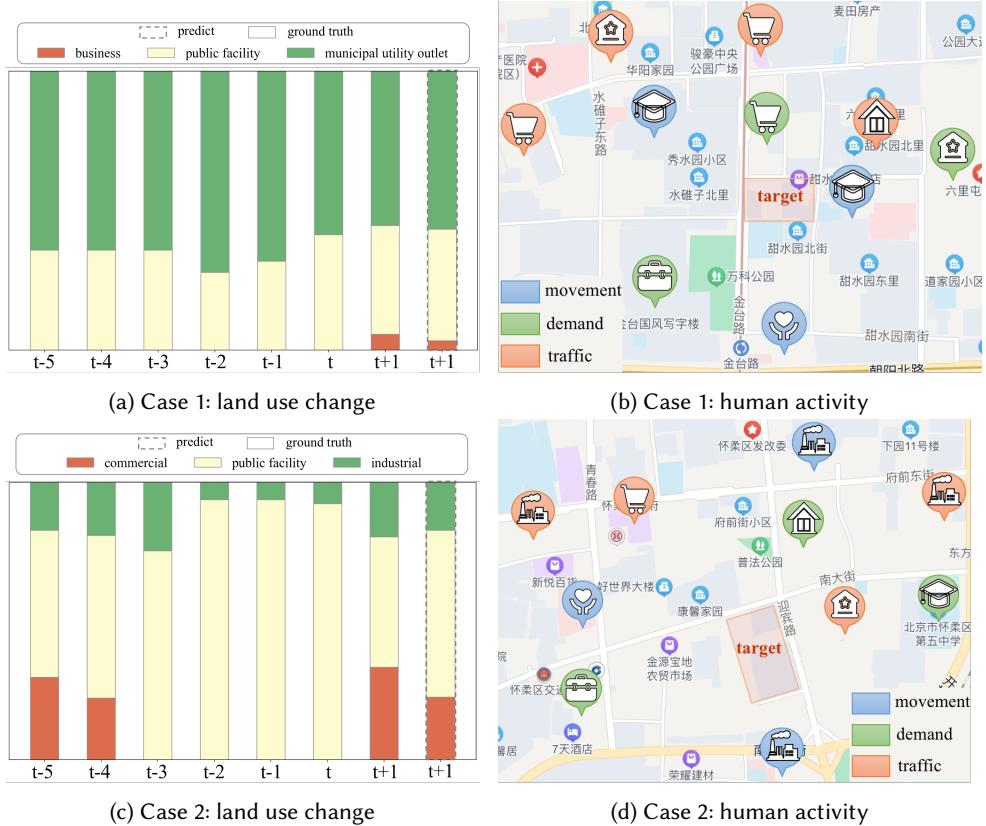


Fig. 11. Case studies in the Beijing dataset.

schools. This suggests that the rise in business activity in the target region may be driven by the surrounding commercial hubs and the local demand for business services.

Case 2 examines a suburban area of Beijing characterized by public facilities and factories. Fig. 11c illustrates the accurate prediction of a temporary disappearance of commercial land use, which returns in the target month. Fig. 11d indicates that the target is surrounded by factories, underscoring the commercial dynamics influenced by industrial activities.

These cases validate the effectiveness of the correlation matrix \mathbf{A} ; in Case 1, commercial and educational sectors are key contributors to business activities, aligning with \mathbf{A} . In Case 2, the connections between industrial, residential, and commercial categories corroborate the insights from \mathbf{A} . The activity graphs further confirm that ExHAGN capably utilizes varied temporal segmentation and clustering to analyze activities from multiple perspectives. Overall, ExHAGN not only predicts existing land use accurately but also adeptly forecasts significant land use changes, effectively leveraging human activity and inherent correlations for precise predictions.

6 CONCLUSION

In this paper, we redefined urban land use change prediction as a regression problem and introduced a data-driven framework, ExHAGN, for explainable prediction in an end-to-end manner. Specifically, we developed a temporal skeletonization mechanism to cluster dynamic graphs across various

snapshots, preserving the inherent “timing clues” from daily human activities. We also implemented a heterogeneous dynamic graph neural network to simultaneously capture neighborhood relationships and long-term temporal dependencies. Furthermore, we introduced an explainable algorithm using causal inference to highlight key factors influencing community-level land use changes, which also guides our model’s learning process. Extensive testing on real-world datasets confirms that ExHAGN surpasses state-of-the-art baselines in predicting land use changes. Our qualitative and quantitative analyses underscore ExHAGN’s effectiveness. To our knowledge, this is the first end-to-end approach to predict land use changes using fine-grained human activity data. Future work will explore the applicability of our predictive model to other domains to assess ExHAGN’s generalizability.

ACKNOWLEDGEMENTS

This research work was supported by Beijing University of Posts and Telecommunications Excellent Ph.D Students Foundation under Grant No. CX2022130.

REFERENCES

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *ArXiv Preprint arXiv:1409.0473* (2014).
- [2] José I Barredo, Marjo Kasanko, Niall McCormick, and Carlo Lavalle. 2003. Modelling dynamic spatial processes: simulation of urban future scenarios through cellular automata. *Landscape Urban Plan.* (2003).
- [3] Michael Batty, Yichun Xie, and Zhanli Sun. 1999. Modeling urban dynamics through GIS-based cellular automata. *Comput. Environ. Urban Syst.* (1999).
- [4] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.
- [5] Hengrui Cai, Rui Song, and Wenbin Lu. 2020. ANOCE: Analysis of Causal Effects with Multiple Mediators via Constrained Structural Learning. In *Proc. Inter. Conf. Learn. Repres. (ICLR)*.
- [6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).
- [7] William HE Day and Herbert Edelsbrunner. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *J. Classif.* (1984).
- [8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Proc. 30st Int. Conf. Neural Inf. Process. Syst. (NIPS) 29* (2016).
- [9] J Ronald Eastman. 2003. IDRISI Kilimanjaro: guide to GIS and image processing. (2003).
- [10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining (KDD-96)*, Vol. 96. 226–231.
- [11] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Ann. Statist.* (2001), 1189–1232.
- [12] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. 2015. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.* 1 (2015), 44–65.
- [13] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *Nature* 453, 7196 (2008), 779–782.
- [14] Ehsan Hajiramezanal, Arman Hasanzadeh, Nick Duffield, Krishna R Narayanan, Mingyuan Zhou, and Xiaoning Qian. 2019. Variational graph recurrent neural networks. *ArXiv Preprint arXiv:1908.09710* (2019).
- [15] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*. 1025–1035.
- [16] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *J. R. Stat. Soc. C: Applied Statistics* 28, 1 (1979), 100–108.
- [17] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. Nais: Neural attentive item similarity model for recommendation. *IEEE Trans. Knowl. Data Eng.* 30, 12 (2018), 2354–2366.
- [18] Wei-Han Lee, Jorge Ortiz, Bongjun Ko, and Ruby Lee. 2018. Time series segmentation through automatic feature learning. *ArXiv Preprint arXiv:1801.05394* (2018).
- [19] Xia Li and Anthony Gar-On Yeh. 2000. Modelling sustainable urban development by the integration of constrained cellular automata and GIS. *Int. J. Geogr. Inf. Sci.* 14, 2 (2000), 131–152.

- [20] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *ArXiv Preprint arXiv:1707.01926* (2017).
- [21] Chuanren Liu, Kai Zhang, Hui Xiong, Guofei Jiang, and Qiang Yang. 2015. Temporal skeletonization on sequential data: patterns, categorization, and visualization. *IEEE Trans. Knowl. Data Eng.* 28, 1 (2015), 211–223.
- [22] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proc. Inter. Conf. Mach. Learn. (ICML)*. PMLR, 4114–4124.
- [23] Ting Ma, Longtao Huang, Qianqian Lu, and Songlin Hu. 2023. Kr-gcn: Knowledge-aware reasoning with graph convolution network for explainable recommendation. *ACM Trans. Inf. Syst.* 41, 1 (2023), 1–27.
- [24] Lin Mu, Lizhe Wang, Yuewei Wang, Xiaodao Chen, and Wei Han. 2019. Urban land use and land cover change prediction via self-adaptive cellular based deep learning with multisourced data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* 12, 12 (2019), 5233–5247.
- [25] Hichem Omrani, Fahed Abdallah, Omar Charif, and Nicholas T Longford. 2015. Multi-label class assignment in land-use modelling. *Int. J. Geogr. Inf. Sci.* 29, 6 (2015), 1023–1041.
- [26] Hae-Sang Park and Chi-Hyuck Jun. 2009. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications* 36, 2 (2009), 3336–3341.
- [27] Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi. 2016. Deep subspace clustering with sparsity prior.. In *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*. 1925–1931.
- [28] Jean-Paul Rodrigue. 2020. *The geography of transport systems*. Routledge. 174–175 pages.
- [29] Alex Rodriguez and Alessandro Laio. 2014. Clustering by fast search and find of density peaks. *Science* 344, 6191 (2014), 1492–1496.
- [30] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal graph networks for deep learning on dynamic graphs. *ArXiv Preprint arXiv:2006.10637* (2020).
- [31] David Schrank, Bill Eisele, and Tim Lomax. 2012. TTI's 2012 urban mobility report. *Texas A&M Transportation Institute. The Texas A&M University System* 4 (2012).
- [32] Kiril Stanilov and Michael Batty. 2011. Exploring the historical determinants of urban growth patterns through cellular automata. *Trans. in GIS* 15, 3 (2011), 253–271.
- [33] Gabor J Szekely, Maria L Rizzo, et al. 2005. Hierarchical clustering via joint between-within distances: Extending Ward's minimum variance method. *J. Classif.* 22, 2 (2005), 151–184.
- [34] Amin Tayyеби and Bryan C Pijanowski. 2014. Modeling multiple land use changes using ANN, CART and MARS: Comparing tradeoffs in goodness of fit and explanatory power of data mining tools. *Int. J. Appl. Earth Obs. Geoinformation* 28 (2014), 102–116.
- [35] Amin Tayyеби, Bryan C Pijanowski, Marc Linderman, and Claudio Gratton. 2014. Comparing three global parametric and local non-parametric models to simulate land use change in diverse areas of the world. *Environ. Model. Softw.* 59 (2014), 202–221.
- [36] Paul M Torrens and David O'Sullivan. 2001. Cellular automata and urban simulation: where do we go from here? , 163–168 pages.
- [37] Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Process.* 167 (2020), 107299.
- [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)* 30 (2017).
- [39] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Attributed graph clustering: A deep attentional embedding approach. (2019), 3719–3725.
- [40] Chao Wang, Hengshu Zhu, Peng Wang, Chen Zhu, Xi Zhang, Enhong Chen, and Hui Xiong. 2021. Personalized and explainable employee training course recommendations: A bayesian variational approach. *ACM Trans. Inf. Syst.* 40, 4 (2021), 1–32.
- [41] Max Welling and Thomas N Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*.
- [42] Roger White and Guy Engelen. 1993. Cellular automata and fractal urban form: a cellular modelling approach to the evolution of urban land-use patterns. *Environ. and Plan. A* 25, 8 (1993), 1175–1199.
- [43] Fengli Xu, Yong Li, Depeng Jin, Jianhua Lu, and Chaoming Song. 2021. Emergence of urban growth patterns from human mobility behavior. *Nat. Comput. Sci.* 1, 12 (2021), 791–800.
- [44] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. Inter. Conf. Mach. Learn. (ICML)*. PMLR, 2048–2057.
- [45] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. 2021. CausalVAE: disentangled representation learning via neural structural causal models. In *Proc. IEEE/CVF Conf. Comp. Vision Pattern Recog. (CVPR)*.

- 9593–9602.
- [46] Qingsheng Yang, Xia Li, and Xun Shi. 2008. Cellular automata for simulating land use changes based on support vector machines. *Comput. Geosci.* 34, 6 (2008), 592–602.
 - [47] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)* 32 (2019).
 - [48] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *ArXiv Preprint arXiv:1709.04875* (2017).
 - [49] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. In *Proc. Inter. Conf. Mach. Learn. (ICML)*. PMLR, 12241–12252.
 - [50] Nicholas Jing Yuan, Yu Zheng, and Xing Xie. 2012. Segmentation of urban areas using road networks. *Microsoft, Albuquerque, NM, USA, Tech. Rep. MSR-TR-2012-65* (2012).
 - [51] Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, Martin Ester, et al. 2022. A Comprehensive Survey on Deep Clustering: Taxonomy, Challenges, and Future Directions. *arXiv preprint arXiv:2206.07579* (2022).