Learning to Simulate Daily Activities via Modeling Dynamic Human Needs

Yuan Yuan
Department of Electronic Engineering
BNRist Tsinghua University
Beijing, China

Huandong Wang
Department of Electronic Engineering
BNRist Tsinghua University
Beijing, China

Jingtao Ding*
Department of Electronic Engineering
BNRist Tsinghua University
Beijing, China

Depeng Jin
Department of Electronic Engineering
BNRist Tsinghua University
Beijing, China

Yong Li
Department of Electronic Engineering
BNRist Tsinghua University
Beijing, China

ABSTRACT

Daily activity data that records individuals' various types of activities in daily life are widely used in many applications such as activity scheduling, activity recommendation, and policymaking. Though with high value, its accessibility is limited due to high collection costs and potential privacy issues. Therefore, simulating human activities to produce massive high-quality data is of great importance to benefit practical applications. However, existing solutions, including *rule-based methods* with simplified assumptions of human behavior and *data-driven methods* directly fitting real-world data, both cannot fully qualify for matching reality. In this paper, motivated by the classic psychological theory, Maslow's need theory describing human motivation, we propose a knowledge-driven simulation framework based on generative adversarial imitation learning. To enhance the fidelity and utility of the generated activity data, our core idea is to model the evolution of human needs as the underlying mechanism that drives activity generation in the simulation model. Specifically, this is achieved by a hierarchical model structure that disentangles different need levels, and the use of neural stochastic differential equations that successfully captures piecewise-continuous characteristics of need dynamics. Extensive experiments demonstrate that our framework outperforms the state-of-the-art baselines in terms of data fidelity and utility. Besides, we present the insightful interpretability of the need modeling. The code is available at https://github.com/tsinghua-fib-lab/Activity-Simulation-SAND.

CCS CONCEPTS

• **Computing methodologies** → **Modeling methodologies**; **Modeling and simulation**; • **Computing methodologiFes** → **Model development and analysis**;

KEYWORDS

Daily activities, Simulation, Human needs, GAIL

ACM Reference Format:
Yuan Yuan, Huandong Wang, Jingtao Ding, Depeng Jin, and Yong Li. 2023. Learning to Simulate Daily Activities via Modeling Dynamic Human Needs. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583276

*Jingtao Ding is the corresponding author (dingjt15@tsinghua.org.cn).

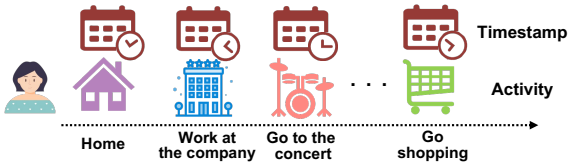


Figure 1: An example of activity sequences, where each entry contains information of the timestamp and activity type.

1 INTRODUCTION

Web applications such as Yelp[1] and Meituan[2] have greatly improved the quality of people's daily life, and at the same time make it possible to record fine-grained activity data. For example, as illustrated in Figure 1, daily life of an individual is usually logged as an activity sequence, *i.e.*, $S = [a_1, a_2, ..., a_n]$, where each entry $a_i = (t_i, k_i)$ contains a timestamp $t_i \in \mathbb{R}^+$ and a discrete activity type $k_i \in C$. Mining activity sequences is valuable for both research and industry in modeling user behaviors and supporting a wide range of applications, like activity planning and recommendation [4, 21, 48]. Despite its high value, only a limited scale of such data is open-sourced for third-party researchers due to privacy-related restrictions on data sharing, which largely hinders the development of downstream applications [23, 24]. Therefore, it is crucial to generate artificial data of human activities by simulation, which can reduce reliance on expensive real data and avoid privacy concerns. In this paper, we study the problem of personalized user activity simulation that models individuals' decision process of what activity to perform at what time, and then generates artificial personalized activity data correspondingly. In order to be publicly shared and used as real-world data, the generated data is expected to be dissociated from real data, *i.e.*, without privacy concerns, and meanwhile capable of retaining data fidelity and utility.

Existing solutions to this problem can be classified into two categories, *i.e.*, *rule-based methods* and *data-driven methods*. *Rule-based methods* that simulate for activity scheduling [2, 6, 10, 26] have a basic assumption that activities can be described by predefined rules derived from activity theories such as utility maximization [43]. However, real-world sequences exhibit complex transition patterns

[1]https://www.yelp.com/
[2]https://about.meituan.com/

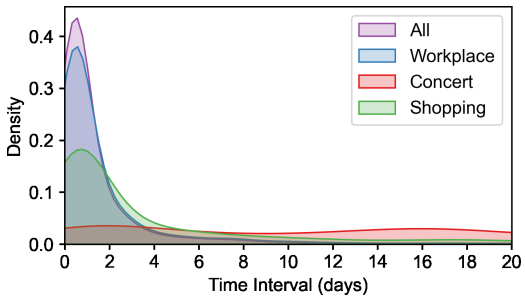


Figure 2: Interval distributions of different activities. Different activities inherently have distinct temporal dynamics.

between activities with time dependence and high-order correlations, which are difficult to describe with prior simple rules [11]. Therefore, only relying on simplified assumptions makes *rule-based methods* less qualified for modeling real-world activity behaviors. Instead, *data-driven methods* tackle this problem by directly fitting real-world data. A series of sequential generative methods have been developed, from classical probability models, such as Markov models [40], to deep learning models, such as Recurrent Neural Networks (RNNs) [13] and Generative Adversarial Imitation Learning (GAIL) [15]. Nevertheless, the above models cannot fully capture the temporal dynamics underlying human daily activities due to the unrealistic inductive bias of being time-invariant [41] or discrete updates only at observed time points [28]. Comparatively, daily activities are always irregularly sampled and longer time intervals introduce larger uncertainty between observations, which requires a deeper understanding and fine-grained characterization.

More importantly, there exist complex and various patterns in terms of temporal dynamics of different activities, which are hard to discriminate from each other when mixed together. For example, as Figure 2 illustrates, time intervals of going to the "Concert" exhibit totally distinct patterns compared with going to the "Workplace" that is highly similar to "All". Although individuals lead generally regular daily routines, some activities still occur occasionally but cannot be ignored. However, with the overall distribution exhibiting long-tailed characteristics, the coarse-grained learning paradigm of state-of-the-art *data-driven methods* can be easily biased by the uneven distribution and fail to adequately capture unique patterns of each activity. Therefore, to generate faithful data that matches reality, it is better not to solely rely on the observed data that may possibly reveal an overall but misleading activity pattern.

To address the above issues and achieve a realistic simulation, we propose a novel framework informed by psychological theories and integrate activity-related knowledge into the state-of-the-art GAIL method. Our key idea is to highlight the intrinsic drives of activity decisions, namely, **human needs**, which are well supported by Maslow's need theories. Accordingly, human needs can be categorized into three levels: *physiological needs*, *safety needs*, and *social needs*. Guided by this knowledge, we explicitly model human needs in a data-driven manner. We disentangle the needs behind daily activities to fully capture the aforementioned complex patterns in empirical data. Specifically, we simultaneously model each need dynamics with an alternating process between *spontaneous flow* and *instantaneous jump*. For example, the accumulation of needs in evolution (flow) triggers the occurrence of related activities while the decaying needs after satisfaction (jump) can restrain tendencies towards specific activities.

In terms of the specific model design, the proposed GAIL-based framework consists of a discriminator that provides reward signals and a generator that learns to generate high-quality activities with a policy network. Particularly, we utilize Maslow's Theory in our framework to enhance the activity simulation with need modeling from the following two perspectives. First, to overcome the challenge of complex activity patterns, we design a hierarchical structure in the modeling to disentangle different need levels and explicitly incorporate the underlying influence of human needs on activity decisions. Second, to address the limitations of RNN-based methods in modeling continuous-time dynamics, we leverage Neural Stochastic Differential Equations [20] to capture piecewise-continuous characteristics of need dynamics alternating between *spontaneous flow* and *instantaneous jump*. The above need dynamics further serve as the states that define the policy function, which calculates activity intensities based on the current need state and decides the next action accordingly. In conclusion, our contributions can be summarized as follows:

- We are the first to explicitly model the intrinsic drives of activities, *i.e.*, human needs, which brings the synergy of psychological theories and data-driven learning.
- We propose a novel knowledge-driven activity simulation framework based on GAIL, leveraging Maslow's theory to enhance the simulation reality by capturing need dynamics.
- Extensive experiments on two real-world datasets show the effectiveness of the framework in generating synthetic data regarding fidelity, utility, and interpretability.

2 PRELIMINARIES

Problem Statement. Daily activity data can be defined as a temporal sequence of events $S = [a_1, a_2, ..., a_n]$, where a_i is a tuple (t_i, k_i), t_i denotes the timestamp and k_i is the activity type, *e.g.*, eating at restaurants, working at companies, playing at sports centers. The problem of activity simulation can be defined as follows:

Definition 1 (Human Activity Simulation). *Given a real-world activity dataset, generate a realistic activity sequence* $\hat{S} = [\hat{a}_1, \hat{a}_2, ..., \hat{a}_n]$ *with a parameterized generative model.*

Temporal Point Process. A temporal point process (TPP) [32] can be realized by an event sequence $\mathcal{H}_T = \{(t_1, k_1), ..., (t_n, k_n) | t_n < T\}$. Here t_i represents the arrival time of the event and k_i is the event mark. Let $\mathcal{H}_t$ denote the history of past events up to time t, the conditional intensity function $\lambda_k^*(t)$ (the k_{th} event category) is defined as: $\lambda_k^*(t) = \lim_{\Delta t \to 0^+} \frac{\mathbb{P}(\text{event of type} k \text{ in} [t, t+\Delta t] | \mathcal{H}_t)}{\Delta t}$. Note that $\lambda^*(t) = \sum \lambda_k^*(t)$ denotes the total conditional intensity, deciding the arrival time without considering event types. Then the event type is sampled at the probability proportional to $\lambda_k^*(t)$.

Neural Ordinary Differential Equations. NODE [8] describes the evolution of the system state over continuous time $t \in \mathbb{R}^+$ by modeling the first-order ordinary differential equations with neural networks. Specifically, the derivative of the latent state is modeled

as: $d\mathbf{h}(t) = f(\mathbf{h}(t), t; \theta) \cdot dt$, where $\mathbf{h}(t)$ is the latent state and f parameterized by a neural network describes the derivative at time t. The system output at time t_1 can be solved with an initial value at time t_0 by an ODE solver: $\mathbf{h}(t_1) = \mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t; \theta) \cdot dt$.

In this work, we take the first attempt to characterize human needs with neural differential equations.

3 METHOD

We first introduce how we model human needs to motivate the framework design in Section 3.1, then explain the MDP modeling of the decision process in Section 3.2, and finally elaborate on the framework details in Section 3.3.

3.1 Human Needs Modeling

3.1.1 ***Hierarchy of Needs.*** According to a classic theory in psychology, i.e., Maslow's Theory [31], people are motivated to achieve a hierarchy of needs, including *physiological needs*, *safety needs*, *social needs*, *esteem needs*, and *self-actualization needs*, in a priority order, where higher levels of need are modeled as long-term changes such as life stages. With the development of Maslow's Theory, the follow-up theories [7, 9, 42] have introduced flexibility in the hierarchy. For example, different needs can be pursued simultaneously, and there exist transition probabilities between any pair of needs. We do not take the top two need levels for *esteem* and *self-actualization* into consideration because they are too abstract and their effects can only be observed in a long term.

Here we classify individuals' activities into three need levels, including *physiological needs* (level-1), *safety needs* (level-2), and *social needs* (level-3), which are sufficient to depict patterns of daily life [22, 23]. These three need levels are often triggered or satisfied in a short period, which are consistent with daily activities that happen within a short term (a few hours). We provide descriptions of each need level as follows:

- ***Physiological needs*** refer to biological requirements for survival, *e.g.*, food, drink, and shelter. The human body cannot function optimally without satisfying these needs.
- ***Safety needs*** refer to requirements for security and safety, *e.g.*, education and employment. Besides physiological needs, people expect their lives to be orderly, regular, and controllable.
- ***Social needs*** refer to requirements for spirits, *e.g.*, entertainment and social relationships. After meeting physiological and safety needs, people are also striving for spiritual satisfaction.

In our modeling, we follow Maslow's Theory in a more flexible way, rather than the original needs pursued in a rigid order. The fulfillment order can be flexible according to individual preferences and external circumstances. Based on well-respected need theories, each activity is explicitly labeled with one of the need levels[3]. The association between human needs and activities based on expert knowledge bridges the gap between classic psychological theories and human behavior modeling, which provides opportunities to model human needs computationally in a data-driven manner.

3.1.2 ***Evolution of Needs.*** In real-world scenarios, human needs are not static but generally evolve with time dynamically, which not only derive from spontaneous changes, but also can be interrupted by happened activities. To better learn sequential activity patterns, it is essential to capture the underlying mechanism of need dynamics. However, it is non-trivial because human needs cannot be observed explicitly and are affected by various factors, such as activity relations and periodicity. Besides, different from activities that happen one by one, need dynamics are more complicated with synchronicity and competitiveness among different levels.

[3]We refer the readers to Section 4.1.2 for more details of the need annotation.

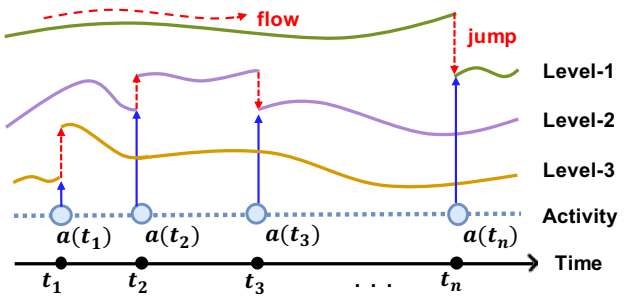


Figure 3: Illustration of the need evolution. Representations of three-level need states evolve continuously over time until interrupted by a corresponding activity (*e.g.*, $a(t_1)$ corresponds to level-1). Note that the need state is modeled by an embedding rather than a scalar, thus the jump up and down do not indicate an increase or decrease.

To effectively capture the underlying need dynamics, we innovatively capture piecewise-continuous dynamics in human needs including *spontaneous flow* and *instantaneous jump* as follows:

- ***Spontaneous flow*** denotes the continuous-time flow of need states. For example, needs for some activities can accumulate without taking them for a long time. Meanwhile, needs can also decay gradually as time goes by.
- ***Instantaneous jump*** models the influence of activities on the need states. For instance, the happened activities can immediately change the evolution trajectory of the corresponding need state.

Naturally, the two kinds of dynamics describe an active process of need evolution and need satisfaction.

Particularly, the three levels are disentangled in dynamic modeling, so they follow distinct evolution laws. Figure 3 illustrates the two evolution mechanisms of different need levels. Nevertheless, it is challenging to learn such dynamics since needs are intrinsically unobserved and stochastic with the coexistence of continuity and jump. To tackle this problem, we represent human needs with a stochastic embedding process $\mathbf{z}(t)$ defined as follows:

Definition 2 (Need Embedding Process). *The need embedding processes are* $\{\mathbf{z}_i(t), i \in \{1, 2, 3\}, t \geq 0\}$, *where* $\mathbf{z}_i(t)$ *is the representation of the* i_{th} *need level at time* t.

In the above definition, we depict human needs with an embedding process $\mathbf{z}(t)$ instead of a direct scalar value for stronger representation capabilities. Particularly, $\mathbf{z}(t)$ is composed of three components $\mathbf{z}_1(t)$, $\mathbf{z}_2(t)$, $\mathbf{z}_3(t)$ that correspond to different need levels. Then the need embedding process $\mathbf{z}(t)$ with both *spontaneous flow* and *instantaneous jump* can be formulated as follows:

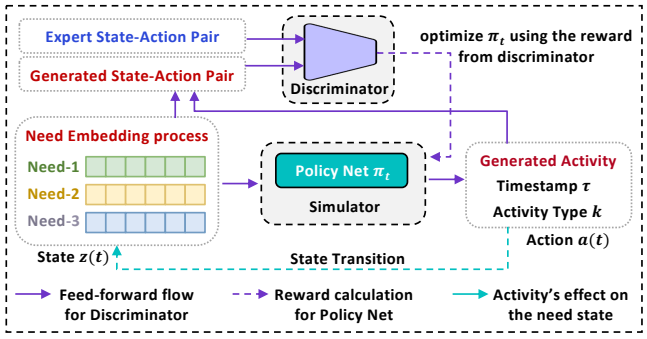


Figure 4: Illustration of the SAND framework. The policy and discriminator networks are optimized adversarially, and the state transition consists of two evolution mechanisms.

$$\begin{cases} \mathbf{z}(t+dt) = \mathbf{z}(t) + \mathcal{F}(t, \mathbf{z}(t))dt, & \text{no activity in } [t, t+dt), \\ \lim_{\Delta t \to 0^+} \mathbf{z}(t_i + \Delta t) = \mathcal{G}(t_i, \mathbf{z}(t_i), k(t_i)), & \text{with activity } k \text{ at the time } t_i, \end{cases} \tag{1}$$

where $\mathcal{F}$ and $\mathcal{G}$[4] control the *spontaneous flow* and *instantaneous jump*, respectively, and $k(t_i)$ denotes the the occurred activity.

3.2 Sequential Decision Processes

The generation of activity sequences depends on individuals' decisions on what activity to take based on his/her own need state step by step. The whole process consists of a sequence of activity decisions that aim to maximize the total received "reward" along the process. Here we model the decision process as a Markov decision process (MDP) [47], and it is described by a 4-tuple $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} >$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the state transition, and $\mathcal{R}$ is reward function. The basic elements of MDPs are : (i) **State** represents the current need state. (ii) **Action** is generated based on the state by sampling a time interval τ and an activity type k. (iii) **Policy function** decides the next activity time and type. (iv) **State transition** controls how the state updates with two transit laws, *i.e.*, spontaneous flow and instantaneous jump. (v) **Reward function** evaluates the utility of taking the action under the state, which is unknown and has to be learned from the data.

Given the activity history $\mathbf{s}_t = \{(t_i, k_i)\}_{t_i < t}$, the stochastic policy function $\pi_\theta(a|\mathbf{s}_t)$ samples an interval time τ and an activity type k to generate the next activity $a = (t_{i+1}, k_{i+1})$, where $t_{i+1} = t_i + \tau$. Then, a reward value is calculated and the state will be updated by an *instantaneous jump*. Besides, there are also feedbacks of need states to the individual over time known as the *spontaneous flow*.

3.3 Proposed Framework: SAND

In this section, we present a novel framework, SAND, which **S**imulates human **A**ctivities with **N**eed **D**ynamics. Overall, it provides the synergy of need theories and imitation learning in simulating the activity decision-making process. As shown in Figure 4, it learns the policy and reward functions adversarially, where the need embedding process $\mathbf{z}(t)$ plays an essential role in the loop. We elaborate on the details of key components in the following sections.

3.3.1 ***Learning Need Dynamics.*** To model the need dynamics including the *spontaneous flow* and *instantaneous jump*, we utilize neural stochastic differential equations [20] to describe such continuity and discontinuity, where the need embedding process $\{\mathbf{z}_i(t), i \in \{1, 2, 3\}, t \geq 0\}$ acts as the latent state. Between activity observations, each $\mathbf{z}_i(t)$ flows continuously over time. Once an activity happens, the corresponding need embedding process is interrupted by a state jump. Different from directly modeling the changes of the hidden state like RNNs [50], neural differential equations model the derivative of $\mathbf{z}(t)$ to better capture the continuous-time characteristics. Specifically, the derivative of the i_{th} need state is formulated as follows:

$$d\mathbf{z}_i(t) = f_i(\mathbf{z}_i(t), t; \theta_i) \cdot dt + \omega_i(\mathbf{z}_i(t), \mathbf{k}_i(t), t; \gamma_i) \cdot dN_i(t) , \tag{2}$$

where f_i and ω_i are both parameterized by neural networks and control the *spontaneous flow* and *instantaneous jump* of the i_{th} need embedding process, respectively, and $N_i(t)$ records the number of activities of the i_{th} level up to time t. f and ω in Eq. (2) are implementations of the function $\mathcal{F}$ and $\mathcal{G}$ defined in Eq. (1). In particular, each state $\mathbf{z}_i(t) \in \mathbb{R}^n$ is composed of two vectors: (1) $\mathbf{c}_i(t) \in \mathbb{R}^{n_1}$ encodes the internal need state, and (2) $\mathbf{h}_i(t) \in \mathbb{R}^{n_2}$ encodes effects of the historical activities.

Spontaneous flow. The top part in Figure 5 shows the network design to model spontaneous flow. The neural function f_i in Eq. (2) controls the spontaneous flow of the state $\mathbf{z}_i(t)$. Although $\mathbf{z}_i(t)$ contains two vectors $\mathbf{c}_i(t)$ and $\mathbf{h}_i(t)$, they follow distinct continuous dynamics due to different encoded information. Specifically, there is no constraint on the internal evolution of $\mathbf{c}_i(t)$, hence we model $\frac{d\mathbf{c}_i(t)}{dt}$ by an MLP. Differently, due to the temporal decaying effect of historical activities, we add constraints to the form of $\mathbf{h}_i(t)$ to model such an effect. Concretely, we use another MLP followed by a Softplus activation layer to model the decay rate. The modeling of derivatives can be formulated as follows:

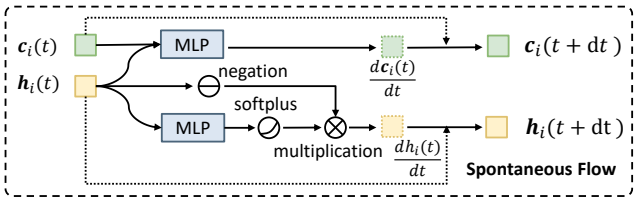
$$\frac{d\mathbf{c}_i(t)}{dt} = \text{MLP}(\mathbf{c}_i(t) \oplus \mathbf{h}_i(t)) , \tag{3}$$

$$\alpha_i = \sigma(\text{MLP}(\mathbf{c}_i(t)) , \quad \frac{d\mathbf{h}_i(t)}{dt} = -\alpha_i \mathbf{h}_i(t) , \tag{4}$$

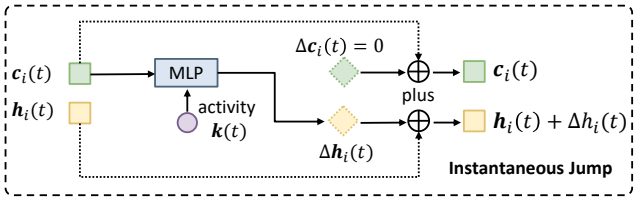
where σ is the Softplus activation function to guarantee a positive decay rate, and $\oplus$ denotes the vector concatenation.

Instantaneous jump. The bottom part in Figure 5 illustrates the network design to model the instantaneous jump introduced by happened activities. Specifically, the function ω_i in Eq. (2) outputs the effects of the instantaneous jump, and it is modeled by an MLP in practice. As discussed before, the vector $\mathbf{h}_i(t)$ encodes the activity memory, and thus it is reasonable that the instantaneous jump will only affect the vector $\mathbf{h}_i(t)$. As a result, an activity of the i_{th} need level gives rise to a change $\Delta\mathbf{h}_i(t)$ only to the corresponding activity memory embedding $\mathbf{h}_i(t)$, *i.e.*, $\Delta\mathbf{h}_j(t) = 0, \forall j \neq i$ and $\Delta\mathbf{c}_i(t) = 0, \forall i$. The MLP takes in the concatenation of the activity embedding $\mathbf{k}(t)$ and the internal state $\mathbf{c}_i(t)$, and outputs the variation $\Delta\mathbf{h}_i(t)$ in the memory embedding $\mathbf{h}_i(t)$, which is formulated as follows:

[4]The time dependent variables in our modeling are all left continuous in t, *i.e.*, $\lim_{\epsilon \to 0^+} \mathbf{z}(t - \epsilon) = \mathbf{z}(t)$.



(a) Spontaneous flow



(b) Instantaneous jump

Figure 5: Network architecture to learn need dynamics with both spontaneous flow and instantaneous jump.

$$\Delta \mathbf{h}_i(t) = \mathrm{MLP}(\mathbf{k}(t) \oplus \mathbf{c}_i(t)) \,, \tag{5}$$

$$\lim_{\epsilon \to 0^+} \mathbf{h}_i(t+\epsilon) = \mathbf{h}_i(t) + \Delta \mathbf{h}_i(t) \,, \tag{6}$$

where $\mathbf{k}(t)$ denotes the activity associated with the i_{th} need level.

3.3.2 ***Policy Function.*** Based on the activity intensity function $\lambda_k(t)$, the probability of activity type k happens within the time interval $[t, t+dt)$ is as: $P\{\text{activity } k \text{ happens in} [t, t+dt)\} = \lambda_k(t) \cdot dt$. The policy function is a mapping from the state to action that generates the arrival of the next activity with the type conditioned on the current state. With the modeling of activity intensities, the goal of the policy function is to generate intensities based on the need states $\mathbf{z}(t)$. Figure 6 shows the network design of the policy function. Although the three need levels control specific activities, they are not independent and can be pursued simultaneously, which may give rise to competing activity choices. Therefore, the states of the three levels all affect the generation of the next activity. In other words, the activity intensity $\lambda_k^*(t)$ is conditioned on embedding processes of all need levels. To model the interactions between different levels in determining the next activity, we concatenate the three embedding processes $\mathbf{z}_i(t), i \in \{1, 2, 3\}$ and leverage an MLP to obtain conditional activity intensities. Here we perform the sampling to obtain the time interval and the activity type based on the total condition intensity and type distribution as:

$$\lambda^*(t) = \sum_{k=1}^{M} \lambda_k(t), \quad p(k|t) = \frac{\lambda_k(t)}{\sum_{k=1}^{M} \lambda_k(t)} \tag{7}$$

where M is the number of activity types.

3.3.3 ***Reward Function.*** GAIL uses a reward function to evaluate the actions by comparing the generated state-action pairs with the real pairs, which is modeled by a discriminator network D_ϕ. To compare the real and policy-generated pairs more effectively, we also utilize the historical sequence information, thus, the state in

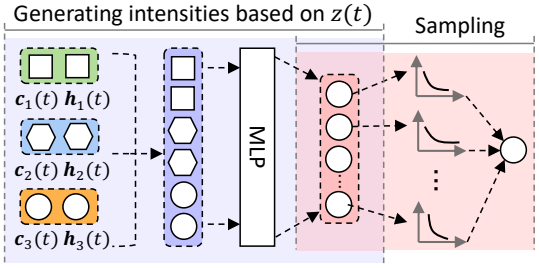


Figure 6: Network architecture of the policy function.

the discriminator is defined as $s_d = (\mathbf{z}(t), \mathrm{S})$. For the sequence $\mathrm{S} = [x_1, x_2, ..., x_n]$, x_i contains the information of the time interval τ_i, hour h_i, weekday w_i, activity type k_i, and need level n_i. In addition, the action a is set as the time interval τ since the last activity to the current one, *i.e.*, $a = (\tau, k)$. Based on the above notations, the output of the discriminator can be defined as $D_\phi(s_d, a)$.

Through an embedding layer, we first transform s_d and a into embeddings. Then we leverage an attention mechanism to aggregate the sequential features. The concatenation of the sequential embedding, state $\mathbf{z}(t)$, and action embedding is fed into an MLP with a sigmoid activation function. Thus, the reward function can be expressed as: $R(s, a) = \log D_\phi(s_d, a)$.

Appendix C introduces the training and simulation procedures.

4 EXPERIMENTS

In this section, we conduct extensive experiments to investigate the following research problems:

- **RQ1**: How does SAND perform in retaining the data fidelity compared with baseline solutions?
- **RQ2**: How do different components of SAND contribute to the final performance?
- **RQ3**: Can SAND generate high-quality synthetic data that benefit practical applications?
- **RQ4**: Can SAND provide insightful interpretations on modeling daily activities?

4.1 Experimental Settings

4.1.1 ***Datasets.*** . We conduct extensive experiments on two real-world datasets. (1) Foursquare-NYC [51] dataset contains checkin activities to various POIs collected from 2000 users with 14 activity labels during the duration from 2012-05-01 to 2012-06-01. (2) Mobile dataset contains 10000 users with 15 activity labels during the duration from 2016-09-17 to 2016-10-17, which is collected in Beijing by a major mobile operator in China. We take careful steps to consider ethical issues in using data: the research protocol has been reviewed and approved by our local institutional board and all research data is sanitized for privacy preservation, with limited access to authorized researchers bound by non-disclosure agreements. More details of the datasets are provided in Appendix A.

4.1.2 Need Annotation. According to the definition and description of each need level in Section 3.1, we ask three annotators to label each activity with one of the need levels. To ensure that correct expert knowledge is utilized, the three annotators all have expertise in related knowledge, including a senior Ph.D. candidate and two

Table 1: Overall performance of SAND and baselines in terms of the JSD-based metrics, and lower results are better. Bold denotes the best results and underline denotes the second-best results. The improvements are significant (p-value<0.05).

Dataset	Mobile Operator						Foursquare					
Metrics (JSD)	MacroInt	MicroInt	DailyAct	ActType	Weekday	Hour	MacroInt	MicroInt	DailyAct	ActType	Weekday	Hour
Semi-Markov	0.291	0.158	0.439	0.471	0.0042	0.051	0.334	0.055	0.485	0.101	0.0032	0.051
Hawkes	0.276	0.151	0.542	0.123	0.0039	0.051	0.073	0.024	0.530	0.026	0.0024	0.047
Neural Hawkes	0.026	0.143	0.125	**0.0063**	0.0036	0.052	0.072	0.041	0.119	0.012	0.0040	0.047
Neural JSDE	0.014	0.106	0.138	0.048	0.0033	0.051	0.041	0.033	**0.056**	0.0072	0.0022	0.046
THP	0.167	0.111	0.058	0.098	0.005	0.040	0.331	0.035	0.095	0.003	0.013	0.047
LSTM	0.110	0.136	0.513	0.342	0.0041	0.050	0.249	0.217	0.628	0.073	0.0033	0.051
SeqGAN	0.143	0.128	0.047	0.054	0.022	0.072	0.225	0.178	0.627	0.065	0.0034	0.051
GAIL	0.089	0.120	0.040	0.231	0.005	0.050	0.226	0.118	0.167	0.087	0.0049	0.062
SAND	**0.0096**	**0.084**	**0.025**	0.036	**0.002**	**0.009**	**0.018**	**0.014**	0.062	**0.0044**	**0.00032**	**0.0069**

postdocs with a background in psychology and behavioral sciences. If the three experts disagree on the label, we will invite another expert and start a discussion. Through this process, all activities obtain consistent labels. The annotation approach has satisfied the requirement of our problem settings due to the small scale of activity types. We also provide scalable methods in Appendix C.1

4.1.3 ***Baselines.*** To evaluate the performance of the SAND framework, we compare it against state-of-the-art baseline methods: Semi-Markov [29], a classical probability model; Hawkes Process [27], a representative point process model; Neural Hawkes Process [32], the neural extension to the Hawkes process; Transformer Hawkes Process [57] (THP) is another neural extension to the Hawkes process, which utilizes the self-attention mechanism to capture long-term dependencies; Neural JSDE [8], the state-of-the-art method to learn continuous and discrete dynamic behavior; LSTM [16], a widely used model in sequence prediction; SeqGAN [54], the state-of-the-art model for discrete sequence generation; TrajGAIL [15], a model-free imitation learning algorithm in trajectory generation.

4.1.4 ***Metrics.*** We measure whether synthetic data accurately reflects crucial characteristics of the original, real-world data. Following the mainstream practice in previous works [11, 37], we use essential metrics to describe activity patterns for comparing the statistical similarity between the generated data and real-world data, including (1) *ActInt*: time intervals between activities, including type-free intervals (MacroInt) and type-aware intervals (MicroInt); (2) *DailyAct*: daily happened activities. It is the number of activities in one day for each individual; (3) *ActType*: the overall distribution over different activity types; (4) *Weekday*: the overall time distribution over the seven days; (5) *Hour*: the overall time distribution over the twenty-four hours. To get the quantitative evaluations on the fidelity of generated data, we use Jensen–Shannon divergence (*JSD*) to measure the distribution similarity of the above patterns between the generated data and real-world data: $\mathrm{JSD}(P||Q) = H(M) - \frac{1}{2}(H(P) + H(Q))$, where H is the Shannon entropy, p and q are two distributions, and $M = \frac{p+q}{2}$. Lower JSD denotes a closer distribution between synthetic data and real data, indicating a better generative model.

4.2 Overall Performance (RQ1)

Table 1 reports the performance in retaining the data fidelity of our framework and the eight competitive baselines on two real-world datasets. From the results, we have the following findings:

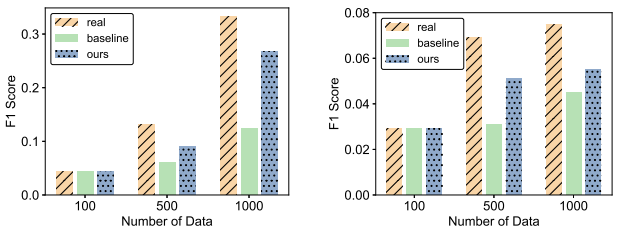
- **Our framework steadily achieves the best performance.** SAND achieves the best performance on the mobile operator dataset, by ranking first on five metrics and second on one metric. For five metrics that rank 1st, SAND reduces the JSD by over 20%. It also shows superior performance on most of the metrics on the Foursquare dataset, which ranks first on five metrics by reducing JSD by over 40%. Meanwhile, it achieves comparable performance with the best baseline on the second-rank metric.
- **Time-invariant model performs poorly in simulating human activities.** Semi-Markov performs the worst in most cases. which indicates that the time-invariant assumption fails to describe behavior transition laws due to the existence of complex temporal patterns in daily activities.
- **Learning from raw data alone is insufficient for a realistic simulation.** The LSTM model has a poor performance on the metrics of *DailyAct* and *ActType*, which means errors can be accumulated in the step-by-step generation process. By contrast, SeqGAN and GAIL improve the performance by using reinforcement learning and adversarial learning. For the Foursquare dataset that is more sparse, their superiority is lost, which further suggests the instability of purely data-driven methods.
- **It is essential to model dynamic human needs.** The neural Hawkes, THP, and neural JSDE almost achieve the sub-optimal results on the two datasets, indicating the rationality of characterizing events in continuous time by temporal point processes. However, without investigating the deeper mechanism behind observed activities, their performance is still limited.

4.3 Ablation Studies (RQ2)

The proposed SAND framework consists of two key components: modeling need dynamics and solving the MDPs with GAIL. Besides, we also use the pre-training mechanism. To further validate whether they are indeed crucial for the final performance, we conduct ablation studies on two datasets by comparing the performance of three variants of SAND, including *SAND - need, SAND - GAIL, SAND - pretrain*. Specifically, *SAND - need* calculates the latent state as [20] without modeling hierarchical human needs, *SAND - GAIL*

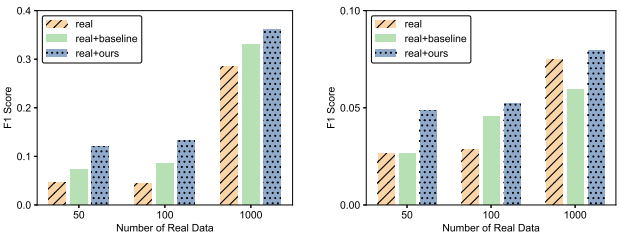
Table 2: Ablation study on SAND variants. Bold denotes the best results and underline denotes the second-best results.

Dataset	Mobile Operator						Foursquare					
Metrics (JSD)	MacroInt	MicroInt	DailyAct	ActType	Weekday	Hour	MacroInt	MicroInt	DailyAct	ActType	Weekday	Hour
SAND	**0.013**	**0.084**	**0.025**	**0.036**	**0.002**	**0.009**	**0.018**	**0.014**	0.062	**0.0044**	**0.00032**	**0.0069**
SAND - GAIL	0.013	0.116	0.085	0.040	0.0031	0.051	0.039	0.028	0.202	0.0051	0.0018	0.0092
SAND - need	0.014	0.116	0.085	0.039	0.0035	0.050	0.019	0.030	**0.0085**	0.0072	0.0021	0.048
SAND - pretrain	0.015	0.110	0.059	0.190	0.004	0.048	0.070	0.025	0.161	0.064	0.0020	0.044



(a) Mobile Operator Dataset. (b) Foursquare Dataset.

Figure 7: Activity prediction in the fully synthetic scenario.



(a) Mobile Operator Dataset. (b) Foursquare Dataset.

Figure 8: Activity prediction in the hybrid scenario. For a different number of real-world sequences, *i.e.*, 50, 100, 1000, we all add 1000 generated sequences for data augmentation.

removes the GAIL training framework, and *SAND - pretrain* starts training from raw data without the pre-training mechanism.

The evaluation results are reported in Table 2. We can observe that SAND delivers the best performance on five metrics compared with the variants that are removed with specific designs. Without modeling need dynamics, the performance is reduced significantly, indicating the necessity to consider the intrinsic motivation in human activity simulation. Besides, removing the GAIL framework also reduces the data fidelity, which suggests the strong modeling capabilities of generative adversarial mechanisms. In addition, the pre-training mechanism facilitates making full use of the activity data and enables our framework to preview the dependencies and regularities of daily activities before GAIL training, thus it also contributes to the final performance.

4.4 Practical Applications (RQ3)

In user-based applications, real-world activity records usually cannot be directly shared due to privacy issues. Under this circumstance, SAND can be used to generate synthetic data to mask sensitive information while retaining the usability of real data. To examine the utility of the generated synthetic data, we perform experiments with synthetic data of two categories:

- **Fully synthetic scenario**; Only synthetic data is used in applications, which provides a more robust privacy protection.
- **Hybrid scenario**; It combines real and synthetic data, which is widely used in data augmentation settings.

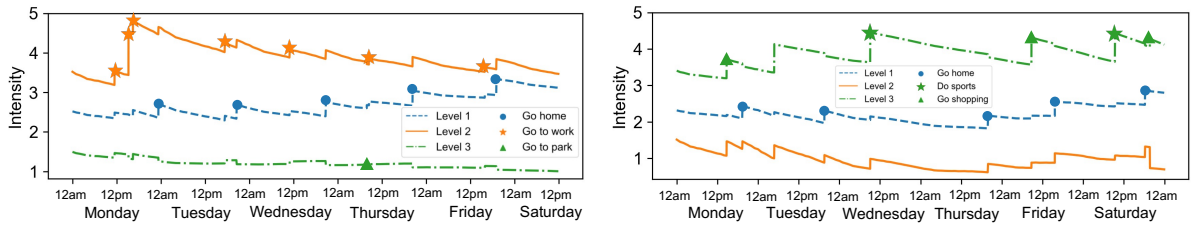
We select two representative applications [18, 33] based on the activity data: (1) activity prediction and (2) interval estimation, which are fundamental to many activity-related problems, such as activity recommendation and planning.

We utilize a widely-used model, LSTM with attention mechanism, to predict individuals' future activity types based on their historical sequence[5]. As shown in Figure 7, compared with the best baseline, the prediction performance on the dataset generated by our framework is much closer to the performance on the real data, showing the retained utility of the generated data. Figure 8 illustrates that the model trained on the augmented data exhibits significantly better performance than that only trained on the real-world data. Meanwhile, the data augmented by SAND outperforms that by the best baseline. Moreover, the augmented data becomes more useful when the real-world data is of small scale, *e.g.*, only with 50 or 100 real-world sequences. These results validate the practical value of the synthetic data.

4.5 Interpretability of Dynamic Needs (RQ4)

To validate whether SAND can provide insightful interpretability, we perform a case study on the learned intensity values of different need levels in the simulation process. Figure 9 illustrates the simulated activity sequences of two individuals for one week, together with the corresponding intensity values of three need levels. In terms of the model interpretability, we have two main observations. First, the proposed SAND can generate distinct but lifelike activity sequences that are hard to tell apart from real-world data. Specifically, comparing Figure 9(a) and (b), the two synthetic individuals lead quite personalized lifestyles. Individual 1 follows regular working routines with the intensity dynamics of the level-2 need varying periodically, while individual 2 enjoys more freedom without working, showing a constantly low intensity of the level-2 need. Second, SAND can simulate human daily activity in an interpretable way with need modeling. As observed from Figure 9,

[5]Due to the page limit, we leave the interval estimation in Appendix D.



(a) Individual 1 (b) Individual 2

Figure 9: Case study of two generated activity sequences and the learned intensity of different need levels. We select two representative individuals with different activity patterns.

the occurrence of activity not only changes the intensity of the corresponding need level but also affects other levels, indicating that different need levels are interconnected by intensities derived from need states and trigger activities in a cooperative manner. In summary, the above observations demonstrate the interpretability of SAND for simulation outcomes, which is equally important in real-life applications.

5 RELATED WORK

5.1 Human activity simulation

Solutions for activity simulation are mainly agent-based modeling [30] with rule-based methods [22–24, 34, 35, 39]. Specifically, these methods assume that human activities can be described by limited parameters with explicit physical meaning and are governed by transition rules based on psychology and social science theories. With simplified assumptions of human behaviors, agents in the system can be assigned different goals, then they take actions to maximize different attributes. For example, Kim et al. [24] propose that human actions are triggered by a cause and give rise to corresponding effects. Besides, considering the multiple behaviors, the priorities of behaviors are determined based on Maslow's hierarchy of needs [22–24]. Despite the promising performance under some circumstances, rule-based methods fail to capture complicated activity patterns due to relying on simplified assumptions and thus usually fail to simulate activities in reality. The purpose of activity simulation is different from that of activity prediction [18, 33, 53]. The former emphasizes the simulation results to reproduce and reflect characteristics of real data, but should not be too similar to real data with the goal of protecting user privacy, while the latter highlights to what extent the model can recover the real data. Although deep learning approaches are proposed for activity prediction [19, 36], the problem of simulating daily activities has been barely explored.

5.2 Deep generative models for activity simulation

Deep generative models, such as generative adversarial networks (GAN) [12] and variational autoencoder (VAE) [25], are promising solutions to simulation. Previous studies [17, 38, 46, 49, 56] have also explored the ability of Generative adversarial Imitation Learning (GAIL) to simulate human decision process. Besides, a series of neural temporal point process models [8, 32, 55, 57] are proposed to model discrete events. Although these models are mainly for discrete event prediction, the learned probability distribution provides opportunities to perform event generation by the sampling operation. Recently, Gupta et al. [14] propose attention-based temporal point process flows to model goal-directed activity sequences. However, it is not appropriate for our research problems as daily activities cannot be represented as a sequence of actions performed to achieve an explicit goal. We propose a knowledge-driven framework based on GAIL, and the incorporation of psychological knowledge is realized by leveraging an ODE-based temporal point process.

6 CONCLUSION

In this paper, we investigate the individual activity simulation problem by proposing a novel framework SAND, which integrates deep generative models with well-respected psychological theories. Extensive experiments on two real-world datasets show the superior performance of the proposed framework. Our framework is not strictly limited to Maslow's theories, instead, what we highlight is leveraging neural networks to learn the driving force behind human daily activities, and the choice of knowledge or theory related to such driving force is quite flexible. For example, ERF theory [1] claims that there exist three levels behind activities, including existence, relatedness, and growth; some other theories [44] propose that several specific travel purposes lead to daily activities. Importantly, effective modeling of human needs makes it possible to understand human behaviors at a deeper level, which not only benefits the activity simulation in this work but also contributes to many other problems of psychology-informed user modeling. In terms of limitations, we recognize that data-driven models largely depend on high-quality datasets. For example, the shortage of long-term and fine-grained datasets hinders the modeling of needs for esteem and self-actualization.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under grant 2020YFA0711403, the National Nature Science Foundation of China under 61971267, 61972223, 62171260, and U1936217, the Young Elite Scientists Sponsorship Program by CIC under 2021QNRC001, and the Guoqiang Institute, Tsinghua University under 2021GQG1005.

REFERENCES

[1] Clayton P Alderfer. 1969. An empirical test of a new theory of human needs. *Organizational behavior and human performance* 4, 2 (1969), 142–175.

[2] Theo Arentze, Frank Hofman, Henk van Mourik, and Harry Timmermans. 2000. ALBATROSS: multiagent, rule-based model of activity pattern decisions. *Transportation Research Record* 1706, 1 (2000), 136–144.

[3] Haytham Assem, Lei Xu, Teodora Sandra Buda, and Declan O'Sullivan. 2016. Spatio-temporal clustering approach for detecting functional regions in cities. In *ICTAI*. IEEE, 370–377.

[4] Joshua Auld and Abolfazl Kouros Mohammadian. 2012. Activity planning processes in the Agent-based Dynamic Activity Planning and Travel Scheduling (ADAPTS) model. *Transp Res Part A Policy Pract* 46, 8 (2012), 1386–1403.

[5] John J Bartko. 1966. The intraclass correlation coefficient as a measure of reliability. *Psychological reports* 19, 1 (1966), 3–11.

[6] John L Bowman and Moshe E Ben-Akiva. 2001. Activity-based disaggregate travel demand model system with activity schedules. *Transportation research part a: policy and practice* 35, 1 (2001), 1–28.

[7] Wei-Lun Chang and Soe-Tsyr Yuan. 2008. A synthesized model of Markov chain and ERG theory for behavior forecast in collaborative prototyping. *JITTA* 9, 2 (2008), 5.

[8] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366* (2018).

[9] Kae H Chung. 1969. A Markov chain model of human needs: An extension of Maslow's need theory. *Academy of Management Journal* 12, 2 (1969), 223–234.

[10] Dick Ettema, Aloys Borgers, and Harry Timmermans. 1993. Simulation model of activity scheduling behavior. *Transportation Research Record* (1993), 1–1.

[11] Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. 2020. Learning to simulate human mobility. In *KDD*. 3426–3433.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *NIPS* 27 (2014).

[13] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

[14] Vinayak Gupta and Srikanta Bedathur. 2022. ProActive: Self-attentive temporal point process flows for activity sequences. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 496–504.

[15] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016), 4565–4573.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[17] Yuchen Wu Depeng Jin Lina Yao Huandong Wang, Changzheng Gao and Yong Li. 2023. PateGail: A Privacy-Preserving Mobility Trajectory Generator with Imitation Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[18] Vikramaditya Jakkula and Diane J Cook. 2007. Mining sensor data in smart environment for temporal activity prediction. *KDD* (2007).

[19] Neziha Jaouedi, Francisco J Perales, José Maria Buades, Noureddine Boujnah, and Med Salim Bouhlel. 2020. Prediction of human activities based on a new structure of skeleton features and deep learning model. *Sensors* 20, 17 (2020), 4944.

[20] Junteng Jia and Austin R Benson. 2019. Neural jump stochastic differential equations. *arXiv preprint arXiv:1905.10403* (2019).

[21] Sidney Katz. 1983. Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living. *Journal of the American Geriatrics Society* 31, 12 (1983), 721–727.

[22] Hamdi Kavak, Joon-Seok Kim, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2019. Location-based social simulation. In *SSTD*. 218–221.

[23] Joon-Seok Kim, Hyunjee Jin, Hamdi Kavak, Ovi Chris Rouly, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2020. Location-based social network data generation based on patterns of life. In *MDM*. IEEE, 158–167.

[24] Joon-Seok Kim, Hamdi Kavak, Umar Manzoor, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2019. Simulating urban patterns of life: A geo-social data generation framework. In *SIGSPATIAL*. 576–579.

[25] Diederik P Kingma, Max Welling, et al. 2019. An introduction to variational autoencoders. *Found. Trends Mach. Learn.* 12, 4 (2019), 307–392.

[26] Ryuichi Kitamura, Eric I Pas, Clarisse V Lula, T Keith Lawton, and Paul E Benson. 1996. The sequenced activity mobility simulator (SAMS): an integrated approach to modeling transportation, land use and air quality. *Transportation* 23, 3 (1996), 267–291.

[27] Patrick J Laub, Thomas Taimre, and Philip K Pollett. 2015. Hawkes processes. *arXiv preprint arXiv:1507.02822* (2015).

[28] Yuxuan Liang, Kun Ouyang, Hanshu Yan, Yiwei Wang, Zekun Tong, and Roger Zimmermann. 2021. Modeling Trajectories with Neural Ordinary Differential Equations.. In *IJCAI*. 1498–1504.

[29] Nikolaos Limnios and Gheorghe Oprisan. 2012. *Semi-Markov processes and reliability*.

[30] Charles M Macal and Michael J North. 2005. Tutorial on agent-based modeling and simulation. In *WSC*. IEEE, 14–pp.

[31] Abraham Harold Maslow. 1943. A theory of human motivation. *Psychological review* 50, 4 (1943), 370.

[32] Hongyuan Mei and Jason Eisner. 2016. The neural hawkes process: A neurally self-modulating multivariate point process. *arXiv preprint arXiv:1612.09328* (2016).

[33] Bryan Minor, Janardhan Rao Doppa, and Diane J Cook. 2015. Data-driven activity prediction: Algorithms, evaluation methodology, and applications. In *KDD*. 805–814.

[34] Goran Murić, Alexey Tregubov, Jim Blythe, Andrés Abeliuk, Divya Choudhary, Kristina Lerman, and Emilio Ferrara. 2020. Massive Cross-Platform Simulations of Online Social Networks. In *AAMAS*. 895–903.

[35] Michael J North, Charles M Macal, James St Aubin, Prakash Thimmapuram, Mark Bragen, June Hahn, James Karr, Nancy Brigham, Mark E Lacy, and Delaine Hampton. 2010. Multiscale agent-based consumer market modeling. *Complexity* 15, 5 (2010), 37–47.

[36] Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-Garadi, and Uzoma Rita Alo. 2018. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications* 105 (2018), 233–261.

[37] Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. 2018. A Non-Parametric Generative Model for Human Trajectories.. In *IJCAI*. 3812–3817.

[38] Menghai Pan, Weixiao Huang, Yanhua Li, Xun Zhou, and Jun Luo. 2020. xGAIL: Explainable Generative Adversarial Imitation Learning for Explainable Human Decision Analysis. In *KDD*. 1334–1343.

[39] Andreas Züfle Pfoser and Carola Wenk. [n. d.]. Towards Large-Scale Agent-Based Geospatial Simulation. ([n. d.]).

[40] Jan-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D Chodera, Christof Schütte, and Frank Noé. 2011. Markov models of molecular kinetics: Generation and validation. *The Journal of chemical physics* 134, 17 (2011), 174105.

[41] Lawrence Rabiner and Biinghwang Juang. 1986. An introduction to hidden Markov models. *ieee assp magazine* 3, 1 (1986), 4–16.

[42] John Rauschenberger, Neal Schmitt, and John E Hunter. 1980. A test of the need hierarchy concept by a Markov model of change in need strength. *Administrative Science Quarterly* (1980), 654–670.

[43] WW Recker and GS Root. 1981. Toward a dynamic model of individual activity pattern formulation. (1981).

[44] Ilan Salomon. 1985. Telecommunications and travel: substitution or modified mobility? *Journal of transport economics and policy* (1985), 219–235.

[45] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[46] Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. 2019. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In *AAAI*, Vol. 33. 4902–4909.

[47] Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*. Vol. 135.

[48] Chen-Ya Wang, Yueh-Hsun Wu, and Seng-Cho T Chou. 2010. Toward a ubiquitous personalized daily-life activity recommendation service with contextual information: a services science perspective. *INF SYST E-BUS MANAG* 8, 1 (2010), 13–32.

[49] Hua Wei, Dongkuan Xu, Junjie Liang, and Zhenhui Li. 2021. How Do We Move: Modeling Human Movement with System Dynamics. In *AAAI*.

[50] Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang. 2017. Modeling trajectories with recurrent neural networks. IJCAI.

[51] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *TSMC* 45, 1 (2014), 129–142.

[52] Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. 2015. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *TSMC* 45, 1 (2015), 129–142.

[53] Jihang Ye, Zhe Zhu, and Hong Cheng. 2013. What's your next move: User activity prediction in location-based social networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 171–179.

[54] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, Vol. 31.

[55] Yuan Yuan, Jingtao Ding, Huandong Wang, Depeng Jin, and Yong Li. 2022. Activity Trajectory Generation via Modeling Spatiotemporal Dynamics. In *KDD*. 4752–4762.

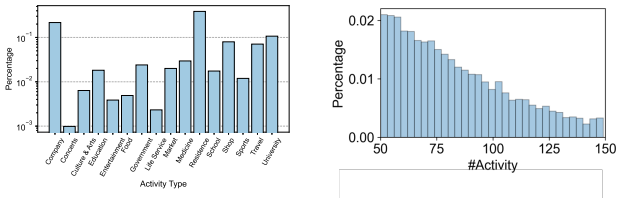
[56] Xin Zhang, Yanhua Li, Xun Zhou, and Jun Luo. 2019. Unveiling taxi drivers' strategies via cgail: Conditional generative adversarial imitation learning. In *ICDM*. IEEE, 1480–1485.

[57] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer hawkes process. In *ICML*. PMLR, 11692–11702.

A DATASET DESCRIPTION

A.1 Mobile Operator Dataset

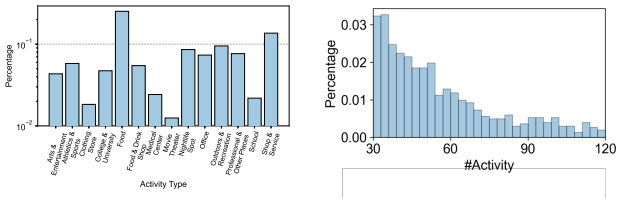
It was collected in Beijing by a major mobile network operator in China for a duration of one month from 2016-09-17 to 2016-10-17. Specifically, we filter out users with less than 50 activities in this period. The data records the anonymous user ID, the visited location category, and the timestamp of each visit. As activity data are characterized by visits to different types of locations, we use the location category as the activity type. Specifically, the activity types in this dataset contain the following categories: Company, Concerts, Culture and Art, Education, Entertainment, Food, Government, Life Service, Market, Medicine, School, Shop, Sports, Travel, and University[6]. Figure 10 shows the statistical distribution.



(a) Distribution of activity type (b) Distribution of the number of activities

Figure 10: The statistics of the mobile operator dataset. (a) Distribution of activity type with y-axis in log scale. (b) Distribution of the number of activities.

As we can see, individuals do not distribute uniformly across different activities. There are more visits to companies, residences, and schools, which are consistent with daily life patterns. In addition, we preprocess the used datasets by filtering out users with less than 50 activities.



(a) Distribution of activity type (b) Distribution of the number of activities

Figure 11: The statistics of the Foursquare dataset. (a) Distribution of activity type with y-axis in log scale. (b) Distribution of the number of activities.

A.2 Foursquare Dataset

This dataset records check-in behaviors in New York City, which has been utilized to model user activity preference [3, 52]. The dataset is originally collected for analyzing people's spatial-temporal regularities of daily activities[7]. Specifically, we consider the duration from 2012-05-01 to 2012-06-01. Similar to the preprocessing of the Mobile Operator Dataset, we also filter out users with less than 30 activities during this period. Each check-in behavior is equivalent to an occurred activity, with the information of the anonymous user ID, timestamp, and venue category. In the same way, we use the venue category as the activity type. Therefore, the activity types in this dataset contain the following categories: Arts and Entertainment, Athletics and Sports, Clothing Store, College and University, Food, Food and Drink Shop, Medical Center, Movie Theater, Nightlife Spot, Office, Outdoors and Recreation, Professional and Other places, School, Shop and Service. Figure 10 shows the statistical distribution.

It should be pointed out that the activity of going home (visiting residential venues) is so sparse in the Foursquare data that we do not consider it, which is different from the Mobile Operator Dataset. Actually, flexible characterization of needs with activities also exhibits the generalization ability of the proposed framework to different scenarios. We also filter out users whose activity sequences are too sparse with less than 30 activities in this dataset.

[6]It is worth pointing out that the number of activity categories is usually less than 20, thus it is scalable to real-world applications and scenarios.

[7]https://sites.google.com/site/yangdingqi/home/foursquare-dataset

B SIMULATION ALGORITHM

Algorithm 1 Activity simulation with need dynamics (generating one individual as an example)

Input: Parameter of the policy function π, start time t_0, end time t_n, initial need state $\mathbf{z}_i(t_0), i \in \{1, 2, 3\}$.
Output: Activity sequence H

initialize $t = t_0, H = \{\}, \mathbf{z}_i = \mathbf{z}_i(t_0), i \in \{1, 2, 3\}$
while $t < t_n$ **do**
  dt = AdpativeStepSize($\mathbf{z}_i$, t, π)
  $(t_{\text{next}}, k_{\text{next}})$ = SimulateNextActivity($\mathbf{z}_i$, t, π)
  if $t_{\text{next}} > t + dt$ **then**
    $\mathbf{z}_i$ = SpontaneousFlow($\mathbf{z}_i$, dt, π), $i \in \{1, 2, 3\}$
  else
    $H = H \cup \{(t_{\text{next}}, k_{\text{next}})\}$
    $\mathbf{z}_i$ = SpontaneousFlow($\mathbf{z}_i$; dt; π), $i \in \{1, 2, 3\}$
    $\mathbf{z}_i$ = InstantaneousJump ($\mathbf{z}_i$, $(t_{\text{next}}, k_{\text{next}})$, π), i corresponds to the activity type k_{next}
  end if
  $t = t + dt$
end while

C IMPLEMENTATION DETAILS

C.1 Scalable Method of Need Annotation

We would like to offer scalable methods if more fine-grained activities are considered. First, we can ask experts to provide detailed definitions, descriptions, and examples of relationships between activities and need levels. Then, we can use Amazon's Mechanical Turk[8] to obtain a reliable need-level label for each activity. Specifically, we assign a need-level label to an activity if two annotators give consistent labels, and ask another annotator to judge it if annotators disagree on the label. To assess the annotation reliability, we can compute the intra-class correlation [5]. In this way, the utilization of expert knowledge is scalable.

[8]It is a crowdsourcing website to perform discrete on-demand tasks that computers are currently unable to do. https://www.mturk.com/

C.2 Model Training

GAIL Training. The training procedure of GAIL is an iterative process of learning a non-linear policy function π_θ and a non-linear discriminator D_ϕ. At each iteration, the policy network generates activity sequences $\mathcal{T}_G$, and then each state-action pair is assigned a reward provided by the discriminator. Then the parameters of the policy network are updated with the reward via the PPO algorithm, which is widely used in reinforcement learning [45]. Subsequently, the generated sequences $\mathcal{T}_G$ and real-world sequences $\mathcal{T}_E$ are used as the training data to optimize discriminator parameters D_ϕ with gradient ascent on the following loss function:

$$\mathcal{L}_D = \mathbb{E}_{(s,a)\in\mathcal{T}_E} \log D_\phi(s,a) + \mathbb{E}_{(s,a)\in\mathcal{T}_G} \log(1 - D_\phi(s,a)). \quad (8)$$

Therefore, the optimization of policy π_θ and discriminator $D\phi$ is to solve a min-max problem with the following objective:

$$\max_{\phi} \min_{\theta} -\lambda H(\pi_\theta) + \mathbb{E}_\pi[\log D_\phi(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D_\phi(s,a))], \quad (9)$$

where $\mathbb{E}_\pi$ represents the expected reward of the sequences under the policy π, and π_E represents the policy under the expert sequences. $H(\pi_\theta)$ is an entropy regularization term, which controls to find the policy π with maximum causal entropy.

Pre-training Mechanism. In order to accelerate the training procedure and improve the performance of the adversarial framework, we first pre-train the embedding network of need dynamics and policy network. In this way, we can enable the generator to preview the important interactions between human needs and activities before GAIL training. To achieve this goal, we utilize the log-likelihood loss function [20] as follows:

$$\mathcal{L} = -\sum_j \log\lambda(\mathbf{z}(\tau_j)) - \sum_j \log p(\mathbf{k_j}|\mathbf{z}(\tau_j)) + \int_{t_0}^{t_N} \lambda(\mathbf{z}(t))dt, \quad (10)$$

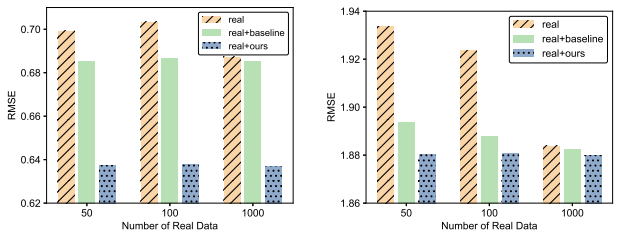
Besides, we also pretrain the discriminator, where we randomly generate activity sequences as fake data. With the fake data and real data, the discriminator is trained to distinguish them with a binary classification loss.

D ADDITIONAL EXPERIMENTAL RESULTS

To validate the utility of our framework, we also use the synthetic data of the best baseline, *i.e.*, NJSDE, for comparison. As a result, for fully synthetic experiments, we have three types of training data: (i) real data, (ii) generated synthetic data of SAND, and (iii) generated synthetic data of the best baseline; for hybrid experiments, the training data includes: (i) real data, (ii) data augmented by SAND, and (iii) data augmented by the best baseline. Correspondingly, the above three trained model is evaluated on the same test set. To obtain a comprehensive view of the augmentation performance, we vary the amount of data used in the training procedure for fully synthetic scenarios. We also vary the amount of real-world data for hybrid scenarios and fix the synthetic data as 1000 sequences.

D.1 Interval Estimation

In addition to the activity type prediction, we also provide experimental results of the activity interval estimation.



(a) Mobile Operator Dataset. (b) Foursquare Dataset.

Figure 12: Interval estimation with data augmentation. For a different number of real-world sequences, *i.e.*, 50, 100, 1000, we all add 1000 generated sequences for data augmentation.

We utilize the neural Hawkes model to estimate the real-valued interval between activities, and the performance is shown in Figure 12. As we can observe, the model trained on the data augmented by our framework not only outperforms real-world data, but also outperforms the data augmented by the best baseline. In addition, on the Foursquare dataset, we can observe that the model trained with only 50 real sequences and 1000 synthetic sequences of our framework has achieved competitive performance by using 1000 real-world sequences, which demonstrates the usability of the simulated data and the effectiveness of the proposed framework.