

ORACLE: Orchestrate NPC Daily Activities using Contrastive Learning with Transformer-CVAE

Seong-Eun Hong
Korea University
Seoul, South Korea
seong_eun@korea.ac.kr

JuYeong Hwang
Korea University
Seoul, South Korea
05judy02@korea.ac.kr

RyunHa Lee
Korea University
Seoul, South Korea
bluerish@korea.ac.kr

HyeongYeop Kang*
Korea University
Seoul, South Korea
siamiz_hkang@korea.ac.kr

Abstract

The integration of Non-player characters (NPCs) within digital environments has been increasingly recognized for its potential to augment user immersion and cognitive engagement. The sophisticated orchestration of their daily activities, reflecting the nuances of human daily routines, contributes significantly to the realism of digital environments. Nevertheless, conventional approaches often produce monotonous repetition, falling short of capturing the intricacies of real human activity plans. In response to this, we introduce ORACLE, a novel generative model for the synthesis of realistic indoor daily activity plans, ensuring NPCs’ authentic presence in digital habitats. Exploiting the CASAS smart home dataset’s 24-hour indoor activity sequences, ORACLE addresses challenges in the dataset, including its imbalanced sequential data, the scarcity of training samples, and the absence of pre-trained models encapsulating human daily activity patterns. ORACLE’s training leverages the sequential data processing prowess of Transformers, the generative controllability of Conditional Variational Autoencoders (CVAE), and the discriminative refinement of contrastive learning. Our experimental results validate the superiority of generating NPC activity plans and the efficacy of our design strategies over existing methods.

Keywords: Generative model, Indoor daily activity plan, CASAS smart home dataset, Contrastive Learning

1. Introduction

Non-player characters (NPCs) are fundamental components of digital environments such as video games and vir-

tual reality (VR), acting as interactive agents that enhance user experiences. Beyond simple task execution, NPCs that exhibit realistic daily routines contribute to deeper immersion by making digital worlds feel more lifelike and dynamic [29, 43]. However, generating diverse, natural, and contextually coherent daily activity sequences for NPCs remains a significant challenge [4].

Traditional approaches to NPC activity modeling have relied on scripted rule-based systems, hierarchical task networks (HTN) [7], and belief-desire-intention (BDI) models [17]. While these methods allow for structured behavior generation, they suffer from limited variability and require extensive manual design to adapt to new environments. More recently, learning-based sequence modeling techniques have been explored, leveraging recurrent and generative models to infer NPC activity sequences from data. Despite these advancements, existing approaches still struggle with three challenges. First, it is challenging to capture long-term dependencies in daily routines, as many models rely on recurrent architectures such as LSTMs, which suffer from vanishing gradients and limited memory when processing long sequences, leading to short-term planning that fails to account for realistic, temporally coherent activity patterns. Second, generating diverse yet realistic schedules remains a challenge, since deterministic models often produce repetitive outputs and fail to introduce controlled variability. Many sequence modeling approaches optimize for likelihood maximization, which biases them toward generating frequent patterns rather than exploring underrepresented but plausible variations in human activity. Third, ensuring coherence when integrating partially defined activity constraints is difficult, as traditional autoregressive and transformer-based models struggle to reconcile predefined activities with generated sequences. These models lack mechanisms to balance adherence to constraints

*Corresponding author

while preserving natural transitions between activities, leading to inconsistencies such as unrealistic gaps or abrupt shifts in daily routines [38].

In parallel, recent work has explored using large language models (LLMs) to plan multi-step behaviors for simulated or embodied agents. However, LLM-based pipelines typically impose substantial inference latency and GPU cost, and reliance on commercial APIs can further introduce per-request fees and governance constraints, which limit adoption in resource-constrained or latency-sensitive industrial settings. Moreover, while LLMs encode broad world knowledge, they are not trained on activity-plan-specific data (e.g., 24-hour routine corpora), often yielding generic or inconsistent schedules unless heavily fine-tuned. These considerations motivate the need for a compact, domain-tailored generative model for activity planning.

To address these limitations, we introduce **ORACLE** (**OR**chestrated NPC Daily Activities using **C**ontrastive Learning with **CV**AE), a generative framework that learns to synthesize realistic human-like daily routines for NPCs. ORACLE is trained on the **CASAS smart home dataset** [15, 16], which provides 24-hour indoor activity sequences. Unlike previous sequence modeling approaches, which primarily focus on direct sequence prediction, ORACLE incorporates. 1) A latent variable model to introduce diversity in generated schedules while maintaining coherence. 2) A contrastive learning framework to refine the generative process, ensuring activity sequences adhere to realistic patterns. 3) A conditional generative structure that supports both full-schedule generation and augmentation of partially defined schedules.

Unlike deterministic sequence generation methods, ORACLE leverages **probabilistic modeling** to create more flexible and adaptable schedules, reducing the tendency for repetitive or unrealistic behavior. The contrastive learning mechanism further enhances the model’s ability to recognize and correct **implausible activity sequences**, ensuring that generated NPC routines remain **coherent and lifelike**.

In summary, the contributions of this paper are as follows:

- We introduce a generative model for NPC daily activity synthesis, integrating latent variable modeling and contrastive learning to enhance sequence diversity and realism.
- Our model is flexible, supporting both full-schedule generation and augmentation of partially defined schedules, making it adaptable to different application scenarios.
- We propose a contrastive learning framework tailored to human activity modeling, improving the model’s ability to generate contextually coherent daily routines.

- We present extensive experiments, including comparisons with existing sequence modeling approaches, a user study, and an ablation analysis, to validate the effectiveness of our approach.

2. Related work

2.1. Plausible NPC Activity Planning

The development of NPCs that inhabit the digital realm like humans spans the fields of AI, gaming, and VR. Achieving this involves either assigning predefined activities to NPCs or enabling them to autonomously decide their actions based on their goals and environmental context.

Plan-based methodologies allow NPC to determine actions in alignment with their objectives and the current environmental conditions. STRIPS [22], an early AI planning system, introduces a formal language for planning, defining actions, and state transitions. STRIPS is enhanced by Planning Domain Definition Language (PDDL) [2, 23] to cover more complex scenarios. In the realm of video gaming, Goal-Oriented Action Planning (GOAP) [34, 3, 49] enable NPCs to devise dynamic strategies tailored to their immediate circumstances, while Hierarchical Task Network (HTN) [24, 36] decomposes tasks into subtasks to create actionable plans. However, these methods often lead to predictable and overly deterministic NPC behavior, reducing their realism.

Motivation-based methodologies use internal drives, such as hunger, safety, and social needs, to generate more varied and sometimes unpredictable NPC behaviors. These approaches emulate human biological and psychological imperatives to create realistic activity plans [10, 17, 39, 4, 25]. The Beliefs-Desires-Intentions (BDI) model [9, 20, 46, 18, 26] further integrates human-like cognitive processes into NPC design, guiding their actions through a matrix of beliefs and intentions. However, building and maintaining a BDI system is labor-intensive and computationally demanding, particularly in balancing diverse and conflicting intentions.

In recent advancements, learning-based methodologies have emerged. Contemporary research increasingly adopts reinforcement and deep learning techniques, empowering virtual agents to refine their decision-making through accumulated experiences [44, 33, 35, 42, 45]. Despite the potent capabilities of these methods, they require significant computational resources and extensive datasets for training. Particularly, the scarcity of comprehensive datasets recording human activity in naturalistic settings poses a substantial bottleneck [51, 15, 38].

2.2. Sequential data analysis

The primary focus of sequential data analysis is developing models adept at delineating the temporal dynamics and

dependencies inherent in data sequences. Historically, techniques such as the Hidden Markov Model (HMM) [5] and dynamic time warping [6] have been extensively studied.

Sequential data analysis has been substantially evolved with the advent of Natural Language Processing (NLP) and deep learning methodologies. Recurrent Neural Networks (RNN) [50], including variants such as Long Short-Term Memory (LSTM) [32] networks and Gated Recurrent Unit (GRU) [12, 13], emerged as foundational models for processing sequential data. These architectures demonstrated exceptional capability in capturing long-range dependencies, thereby significantly advancing tasks such as machine translation [12], text generation [48], and speech recognition [27].

Despite their success, RNN and LSTM encountered notable training challenges over extended sequences, attributed primarily to vanishing and exploding gradient issues [31]. RNN and LSTM have been widely used to deal with sequence data. Koushik [38] conducted an activity-based travel demand modeling using bidirectional LSTM [28]. The introduction of the Transformer [52] model innovatively circumvented these challenges by leveraging self-attention mechanisms, enabling the parallel processing of sequence data and markedly enhancing model efficiency. Consequently, the Transformer architecture has become a cornerstone in the field, with foundational models such as Bidirectional Encoder Representations from Transformers (BERT) [19], Generative Pre-trained Transformer (GPT) [47], and their variants establishing new standards across a wide range of sequential data analysis applications.

In parallel, there has been an increasing interest in leveraging generative models for sequential data analysis. Variational Autoencoder (VAE) [37], known for their proficiency in learning latent data representations, have shown promise in generating high-quality, diverse sequences that closely mimic the statistical properties of real-world data. Bowman [8] combine LSTM and VAE to provide diversity in sequence data as a generative model. Additionally, Conditional variational Autoencoder (CVAE) achieves controllable generation through exploiting learned latent representations [53]. Le Fang [21] suggests Transformer-based CVAE to bring the advantages of Transformer and CVAE.

On the other hand, contrastive learning has emerged as a pivotal technique in large-scale unsupervised learning, focusing on learning feature spaces by bringing similar samples (positive samples) closer together and pushing dissimilar ones (negative samples) apart [30, 11]. This method has been particularly successful in representation learning for sequential data [41].

3. Dataset

To train ORACLE on human activity planning within indoor home environments, we use the CASAS smart home

Categorized Classes	Criteria
Sleep	5 to 12 hours a day, Once a day
Outing	Up to 12 hours a day, No restrictions
Rest	Up to 12 hours a day, No restrictions
Work	Up to 12 hours a day, No restrictions
Hygiene	Up to 1.5 hours per occurrence, At least once a day
Toilet	Up to 0.5 hours per occurrence, At least once a day
Dress	Up to 1 hour per occurrence, At least once a day
Cook	Up to 2 hours per occurrence, No restrictions
Meal	Up to 2 hours per occurrence, At least once a day
Chore	Up to 2 hours per occurrence, At least once a day
Snack	Up to 2 hours per occurrence, No restrictions
Medicine	Up to 0.3 hours per occurrence, No restrictions

Table 1. The criteria for data filtering.

dataset, which consists of two subsets: the Apartment dataset and the Home dataset. These datasets contain continuous 24-hour activity sequences recorded through ambient sensors installed in residential settings. The Apartment dataset includes 42 activity classes, while the Home dataset contains 46 classes. Data was collected from 20 participants (8 males, 12 females, aged 21-62, $\mu = 33$), resulting in a total of approximately 57,216 hours of recorded activity in the Apartment dataset and 26,112 hours in the Home dataset.

Despite the extensive data collection, several challenges arise: 1) Unlabeled periods categorized as *Other activity*, reducing the reliability of activity segmentation. 2) Severe class imbalance, where activities such as *Entertain Guests* appear over 1.27 million times, while *Exercise* appears only 93 times. 3) Incomplete 24-hour cycles due to sensor interruptions, leading to inconsistencies in activity sequences.

3.1. Preprocessing

To enhance the dataset’s usability for training, we applied the following preprocessing steps. First, the duration of the first and last daily activities, typically sleep, was adjusted to span the entire 24-hour period from 00:00:00 to 23:59:59. Next, for segments labeled as *Other activity*, the initial half was replaced with the preceding activity, and the latter half with the subsequent activity. This was also applied to shorter intermittent gaps by extending neighboring activities. To mitigate class imbalance, the original 42 and 46 activity categories were consolidated into 12 major activities by grouping semantically similar activities.

The dataset was then reformatted into uniform sequences of 86,400 seconds (24 hours). Due to GPU memory constraints, the sequences were further segmented into 5-minute intervals, with each interval labeled based on the predominant activity. This resulted in sequences of 288 intervals per day, making training computationally feasible.

In the CASAS smart home dataset, 42 activity classes in Apartment dataset and 46 activity classes in Home dataset were used to label the human daily activities. During the

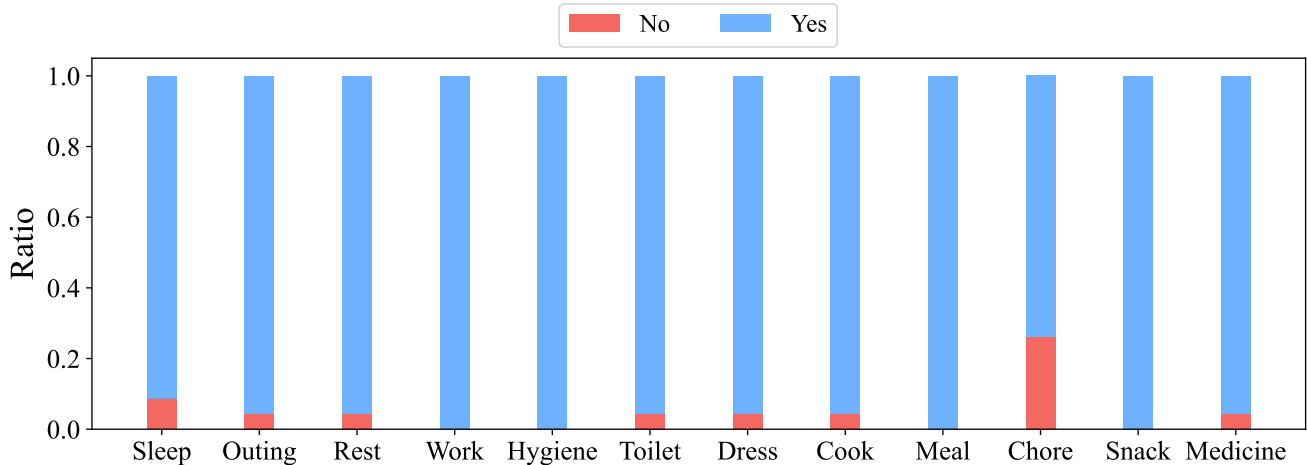


Figure 1. The majority of users responded that the proposed filtering criteria for each activity are suitable for use as the minimum conditions to evaluate the activity’s plausibility.

Major activity classes	Original activity classes
Sleep	sleep, sleep out of bed, go to sleep, nap, wake up
Outing	leave home, enter home, step out, single leave, single enter, staff leave, staff enter, single step out
Rest	watch TV, entertain guests, read, relax, phone, exercise, pet activity, sew
Work	work, work on computer, work at table, work at desk
Hygiene	personal hygiene, groom, bathe, shower
Toilet	toilet, bed toilet transition
Dress	dress
Cook	cook, cook breakfast, cook lunch, cook dinner,
Meal	eat breakfast, eat lunch, eat dinner
Chore	wash dishes, wash breakfast dishes, wash lunch dishes, wash dinner dishes, laundry, housekeeping, put groceries away
Snack	drink, eat
Medicine	take medicine, morning meds, evening meds

Table 2. 42 activity classes in Apartment dataset and 46 activity classes in Home dataset are merged into 12 major activities.

preprocessing, we consolidated 42 and 46 activity categories into 12 major activities to enhance the learning by

grouping similar activities, as shown in Table 2. For example, the major activity class “Sleep” encompasses specific behaviors related to sleeping, such as “sleep”, and “sleep of bed”.

3.2. Data Filtering

Following preprocessing, we applied a filtering step to remove unrealistic activity patterns based on duration and frequency constraints. Filtering criteria were established through a combination of: 1) Common sense reasoning using ChatGPT [1], 2) A user survey involving 23 participants (16 males, 7 females, aged 22-26, $\mu = 23$), 3) Expert validation by the authors.

To ensure high-quality training data, activity sequences that deviated significantly from typical human behavior were excluded. After filtering, the dataset retained approximately 57,216 hours of recorded activity from the Apartment dataset and 26,112 hours from the Home dataset. The filtered data was then split into training, validation, and test sets using an 8:1:1 ratio, ensuring a robust evaluation framework.

By emphasizing *total recorded hours* rather than individual sequence counts, we highlight the scale and depth of human activity patterns captured in the dataset.

Filtering criteria were delineated from the ChatGPT’s knowledge, human knowledge, and the author’s intuition. Exploiting GPT’s knowledge base, we identified “Duration” and “Frequency” as universal numerical parameters for defining 12 activities and determined their desired values considering the plausible human daily activities. The text prompts and GPT outputs are presented in Appendix.

Subsequently, a survey involving human participants was conducted to examine the plausible duration and frequency for each of the 12 activities. Integrating insights from both

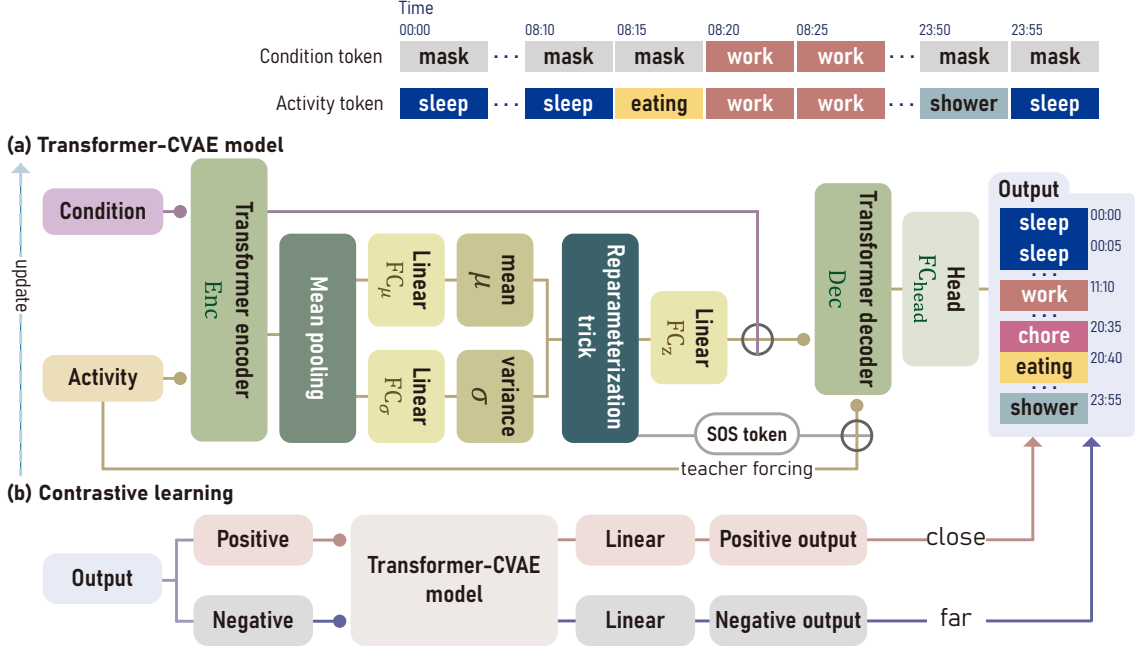


Figure 2. The workflow of ORACLE framework.

GPT outcomes and human responses, the authors established the filtering criteria utilizing their intuitive judgment.

To validate these criteria, a subsequent survey assessed human consensus regarding their appropriateness for evaluating the plausibility of activities. As shown in Figure 1, the participants predominantly affirmed the suitability of these criteria for such evaluations. The survey questions are presented in Appendix. These results suggest that the proposed criteria are reasonable as conservative minimum plausibility checks, and that they capture broadly acceptable constraints rather than idiosyncratic rules tied to a specific individual or household. Consequently, the criteria provide a more general and reproducible basis for filtering and evaluation across different sequences and settings within our indoor daily-routine domain.

4. Methodology

The ORACLE framework, as illustrated in Figure 2, incorporates Transformer architecture, CVAE, and contrastive learning techniques. The utilization of Transformer architecture targets the issue of class imbalance within daily activity datasets, where activities like sleeping may dominate others like restroom usage. This architecture aims to appropriately weigh activities based on their occurrence rates, enhancing the model’s capability to produce balanced and plausible daily plans.

Furthermore, the CVAE is employed within ORACLE to enable the generation of daily activity plans under two scenarios: scenarios with NPCs having predetermined activities and those without any pre-scheduled activities. This

flexibility allows ORACLE to tailor daily plans that seamlessly adapt to various contexts. Our activity plans are represented as discrete tokens over 12 activity classes, and the generation objective focuses on producing coherent token sequences under optional masking constraints. In this discrete, long-horizon setting, a Transformer-based CVAE provides a natural way to model stochasticity and controllability in sequence space, while remaining simple and stable to train on limited data. Although diffusion models are a promising alternative, they typically require additional design choices (e.g., discrete diffusion formulations or continuous relaxations) and higher computational cost; thus we adopt a Transformer-CVAE as a more direct fit to our tokenized activity representation.

Lastly, contrastive learning is exploited to enhance the learning efficacy and quality of generated samples. By distinguishing between similar and dissimilar samples, contrastive learning facilitates a more nuanced understanding and representation of the data, contributing significantly to the model’s overall performance.

4.1. Transformer-CVAE model

The inputs for training consist of activity and condition tokens. Activity tokens form a sequence of 288-length daily activities $A = \{a_1, a_2, \dots, a_{288}\}$, with each token a_i corresponding to a distinct activity class spanning a 5-minute duration. These tokens are processed through the Transformer encoder Enc , as defined by:

$$Enc(A) = \text{SelfA}(A) \in \mathbb{R}^{288 \times C}, \quad (1)$$

where $\text{SelfA}(\cdot)$ represents the self-attention blocks within the Transformer encoder, and $C = 768$ represents the embedding dimension. The self-attention mechanism, adept at capturing both local and global dependencies within the sequence, is effective for long-sequence management. Then, mean-pooling is applied to the encoder’s output to produce a single 768-dimensional vector $\bar{V}(A)$, followed by a linear projection to delineate the latent representations into their *mean* (μ) and *variance* (σ) via $FC_\mu(\cdot)$ and $FC_\sigma(\cdot)$ layers:

$$\begin{aligned}\bar{V}(A) &= \text{MeanPooling}(\text{Enc}(A)), \\ FC_\mu(\bar{V}(A)) &= \{\mu_j \mid j \in [1, 2, \dots, C_0]\}, \\ FC_\sigma(\bar{V}(A)) &= \{\log \sigma_j^2 \mid j \in [1, 2, \dots, C_0]\},\end{aligned}\quad (2)$$

where C_0 represents the embedding dimension, and FC_μ and FC_σ represent two linear projection layers that transition data from \mathbb{R}^C to \mathbb{R}^{C_0} . Although C_0 is potentially smaller than C , the model performance is optimized when $C_0 = C$ in our internal test.

Subsequently, we employ a reparametrization technique to sample the latent variable z_j , constructing a set of latent variables $Z = \{z_1, z_2, \dots, z_{C_0}\}$:

$$z_j = \mu_j + \sigma_j \cdot \epsilon, \quad (3)$$

where $\epsilon \sim \mathcal{N}(0, 1)$. To ensure that the distribution of the latent variables remains close to a standard normal distribution, thereby facilitating efficient and meaningful latent space exploration, Kullback-Leibler(KL) regularization, represented as the regularization loss term L_{reg} , is applied with a weighting of 1×10^{-5} :

$$L_{reg}(\bar{V}(A)) = \frac{1}{C_0} \sum_{j=1}^{C_0} \frac{1}{2} (\mu_j^2 + \sigma_j^2 - \log(\sigma_j^2) - 1). \quad (4)$$

Another training input is the sequence of condition tokens $M = \{m_1, m_2, \dots, m_{288}\}$ where each m_i represents whether the corresponding activity token a_i is masked. For example, in scenarios where the activities of a_3 and a_4 are pre-scheduled, the token m_3 and m_4 denote an unmasked state, whereas the remaining elements in M indicate a masked state.

To optimize parameter efficiency within the model, we employ the same Encoder Enc embedding both activity and condition tokens. Then, the latent variable set Z undergoes transformation via a linear layer $FC_z(\cdot)$ and is then concatenated with the condition token embeddings $\text{Enc}(M)$:

$$M' = \text{Concat}(FC_z(Z), \text{Enc}(M)), \quad (5)$$

M' serves as both the key and value within the Transformer decoder’s cross-attention layer:

$$\text{Dec}(A, M) = \text{CrossA}(E_{MSA}(A), M') \in \mathbb{R}^{288 \times C}, \quad (6)$$

where $E_{MSA}(A)$ represents the output from masked self-attention blocks, facilitating auto-regressive decoding. The notation $\text{CrossA}(\cdot)$ refers to cross-attention blocks within the Transformer decoder architecture.

Then, the probabilities for each of the 12 possible activities, denoted as P_{act} , are calculated through the last fully connected layer FC_{head} :

$$P_{act} = FC_{head}(\text{Dec}(A, M)). \quad (7)$$

This model training follows the Maximum Likelihood Estimation (MLE) methodology, leveraging $FC_z(Z)$ as the start of the sequence (SOS) token. The primary focus is on reconstructive accuracy, which is enhanced by minimizing the negative log-likelihood through reconstruction loss term L_{recon} :

$$L_{recon} = - \sum_{A \in D} \log P(A|A, M; \theta), \quad (8)$$

where θ denotes the model parameter of the model, $P(\cdot)$ denotes the probabilistic output of the ORACLE model, and D denotes a training dataset.

4.2. Contrastive Learning

Our approach is anchored in a Transformer-CVAE structure, designed for a reconstruction task that processes input data to generate new data closely mimicking the original. While contrastive learning is widely used for representation learning, it has rarely been tailored to long-horizon *discrete daily schedule generation* where realism must be judged at the sequence level (24 hours, 288 tokens) and the model is trained without task-specific pretraining. In ORACLE, we adapt contrastive learning to this setting by *defining positives and hard negatives in the space of generated schedules* using plausibility criteria (Table 1), rather than relying on instance identity or data augmentation as in standard contrastive pipelines. This design explicitly pushes the decoder away from near-miss but unrealistic routines, providing weak supervision aligned with our planning objective and serving as the key ingredient that makes contrastive learning effective for NPC routine synthesis.

Here, we employ contrastive learning to consider the divergence between generated data A_{gen} , either semantically or structurally, even when they remain within close proximity to the original data’s distribution in the latent space. In cases where the generated data A_{gen} aligns with the evaluation criteria as presented in Table 1, it is classified as a positive sample A_{pos} ; otherwise, it is deemed a negative sample A_{neg} . This scheme facilitates the model’s enhanced learning from “hard” negative samples. Loss functions for contrastive learning loss are defined as follows:

Datasets	Model	WD ↓	LLM score ↑	REAL-Random ↑	Distinct-10-Random ↑	Distinct-15-Random ↑	SAM ↓	SAM-90 ↓	REAL-Condition ↑	Distinct-10-Condition ↑	Distinct-15-Condition ↑
Apartment dataset	Bowman	0.6263	0.4750	0.4715	0.1899	0.3262	-	-	-	-	-
	Vaswani	-	-	-	-	-	0.5423	1.0491	0.3430	0.1760	0.2809
	Koushik	-	-	-	-	-	0.3390	0.7099	0.6131	0.2633	0.4032
	ORACLE	0.4835	0.5240	0.8416	0.2106	0.3401	0.4218	0.6899	0.9423	0.2831	0.4327
Home dataset	Bowman	1.0470	0.4600	0.1267	0.1699	0.2782	-	-	-	-	-
	Vaswani	-	-	-	-	-	0.5976	1.0960	0.2000	0.2305	0.3553
	Koushik	-	-	-	-	-	0.4437	0.9157	0.5666	0.3396	0.5109
	ORACLE	0.8256	0.5137	0.7433	0.1969	0.3414	0.4495	0.7647	0.9133	0.3276	0.4973

Table 3. Performance benchmark of ORACLE against three existing models.

Datasets	Model	WD ↓	LLM score ↑	REAL-Random ↑	Distinct-10-Random ↑	Distinct-15-Random ↑	SAM ↓	SAM-90 ↓	REAL-Condition ↑	Distinct-10-Condition ↑	Distinct-15-Condition ↑
Apartment dataset	ORACLE -C	0.7161	0.4879	0.5942	0.1582	0.2518	0.4233	0.6964	0.9058	0.2662	0.4098
	ORACLE -T	0.5281	0.4849	0.4985	0.1861	0.3211	0.4278	0.6867	0.7153	0.2731	0.4237
	ORACLE	0.4835	0.5240	0.8416	0.2106	0.3401	0.4218	0.6899	0.9423	0.2831	0.4327
Home dataset	ORACLE -C	1.2898	0.4220	0.4833	0.1300	0.2188	0.4744	0.8262	0.8500	0.2984	0.4482
	ORACLE -T	1.0463	0.4430	0.4200	0.1719	0.2719	0.4813	0.7896	0.5633	0.3300	0.4934
	ORACLE	0.8256	0.5137	0.7433	0.1969	0.3414	0.4495	0.7647	0.9133	0.3276	0.4973

Table 4. Ablation study results.

$$\begin{aligned}
L_{positive} &= 1 - \text{sim}(\text{Dec}(A), \text{Dec}(A_{pos})), \\
L_{negative} &= \text{sim}(\text{Dec}(A), \text{Dec}(A_{neg}))^2 \\
L_{contrastive} &= L_{positive} + L_{negative}.
\end{aligned} \tag{9}$$

Implementing contrastive learning introduces non-ground truth tokens during training, reducing the gap between training and inference. This strategy alleviates the exposure bias commonly encountered in autoregressive models using teacher forcing.

Furthermore, the generation of positive and negative samples as part of the contrastive learning process not only augments the data diversity but also acts as a mechanism for knowledge injection. Such an approach is particularly beneficial for models like ours that commence without a pre-trained foundation, lacking the extensive knowledge typically derived from large datasets.

5. Experiments

5.1. Implementation details

Our model uses the BERT-base-uncased architecture with 768 hidden units, 12 attention heads, 12 transformer layers, and Gaussian Error Linear Unit (GELU) activation. This setup balances computational efficiency with the ability to capture complex data patterns, leveraging the proven effectiveness of BERT-base in sequential data processing. All experiments were conducted using a single RTX 3090 GPU.

5.2. Comparative Evaluation

This section benchmarks ORACLE against three models in two scenarios: random generation and conditional generation. For a fair comparison with prior activity-sequence generation approaches on CASAS-style data, we include representative LSTM-based and Transformer-based baselines that are trained/evaluated on the same preprocessed CASAS splits used in our experiments. Specifically, Bowman et al. [8] serves as an LSTM-based latent generative baseline for unconditional generation, while Vaswani et al. [52] (Transformer) and Koushik et al. [38] (biLSTM) provide strong non-latent sequence-modeling baselines for the masked completion setting. In the random generation scenario, where condition tokens are fully masked, ORACLE is compared to the VAE-LSTM model by Bowman [8]. The LSTM’s sequential processing constraints limit its ability to handle long sequences and complex activity relationships, where ORACLE shows advantages. In the conditional generation scenario, where activities are partially masked, ORACLE is compared to models by Vaswani [52] and Koushik [38]. These models lack generative flexibility, producing identical outputs for the same inputs, which limits diversity. ORACLE, by contrast, excels in generating more diverse and contextually appropriate plans, even under data sparsity. All evaluations were conducted using a test dataset, with the generation process repeated 10 times for statistical robustness.

5.3. Metrics

For the random generation scenario, our model’s performance is evaluated using the Wasserstein distance (WD),

LLM score, REAL-Random, and Distinct-10/15-random. WD measures differences between distributions while the LLM score evaluates the realism and human likeness on a scale from 0 to 100, leveraging the ChatGPT’s comprehensive knowledge base. To ensure consistency with other metrics, the LLM score is normalized by dividing by 100. The used text prompts and their outcomes are presented in Appendix. REAL-Random score quantifies the proportion of randomly generated outputs that meet the selection criteria detailed in Table 1, indicating how well the model adheres to realistic constraints. Distinct-10/15-Random provides our primary quantitative *diversity* analysis by measuring the uniqueness of long-range n -grams [40] across multiple generations, with $n=10$ and 15. Higher Distinct-10/15 indicates that the model produces a broader set of non-redundant activity subsequences (i.e., less mode collapse), which is particularly important for evaluating the diversity benefit of latent-variable generation.

For the conditional generation scenario, our model’s performance is evaluated using the Sequence Alignment Method (SAM) distance, REAL-Condition, and Distinct-10/15-Condition. SAM distance assesses how closely the generated sequence matches the ground truth by applying penalties across the entire sequence of 288 elements: “1” for additions or deletions, and “2” for substitutions. Here, a higher penalty indicates less similarity. These cumulative penalties are then normalized by the sequence length. REAL-Condition and Distinct-10/15-Condition metrics are identical to the REAL-Random and Distinct-10/15-Random metrics used in the random generation scenario, but they are specifically tailored to evaluate model performance under our conditional generation scenarios.

5.4. Evaluation Results

The results of the evaluation are presented in Table 3. For the random generation scenario, ORACLE outperforms the hybrid LSTM-VAE model by Bowman. Firstly, a lower WD signifies a reduced discrepancy between two probability distributions. In generative model contexts, this suggests that the model’s generated data distribution closely mirrors the true data distribution, implying more accurate learning of the data’s distribution. Secondly, a higher LLM score signifies that the generated outputs are more plausible. This indicates that the model effectively captures the relationships between activity classes, leading to daily activity plans that appear more human-like. Additionally, ORACLE achieves higher REAL-Random, Distinct-10-Random, and Distinct-15-Random scores, demonstrating that the generated outputs are not only realistic but also diverse. The high REAL-Random score indicates that ORACLE generates sequences that better meet the criteria in Table 1, while the higher Distinct-10-Random and Distinct-15-Random scores reflect the model’s ability to produce a wide variety of unique and

non-repetitive sequences.

For the conditional generation scenario, the comparative performance of ORACLE is mixed. In terms of SAM distance, ORACLE outperforms the Transformer model by Vaswani yet falls short compared to the bidirectional LSTM-based model by Koushik. This is attributed to the ORACLE’s generative nature, which promotes diversity, consequently placing ORACLE at a comparative disadvantage when evaluated using SAM distance. Nevertheless, when the masking ratio exceeds 90%, as assessed by SAM-90, ORACLE is observed to outperform the aforementioned two models. Moreover, ORACLE generally exhibits higher REAL-Condition, Distinct-10-Condition and Distinct-15-Condition scores, indicating that it generates sequences that are not only more aligned with the given conditions but also maintain greater diversity, producing a broader range of unique sequences under specific constraints.

Additionally, we examined whether ORACLE simply memorizes and reproduces training samples by performing a kNN analysis between generated sequences and the training set. Since each day is discretized into a 288-length sequence with 5-minute activity tokens, we use the Hamming mismatch (the number of different tokens) as the distance. Across all settings, we observed zero exact matches (distance = 0), indicating that ORACLE does not directly copy any training sequence. Concretely, for the Apartment dataset, ORACLE random generation yields a top-1 nearest-neighbor mismatch of 99.50/288 (34.55%) on average (median 98, minimum 67), and a top-5 average mismatch of 105.38/288 (36.59%). For Apartment conditional generation, the top-1 mismatch is 96.47/288 (33.50%) on average (median 98, minimum 40), with a top-5 average mismatch of 102.14/288 (35.46%). For the Home dataset, ORACLE random generation shows a top-1 mismatch of 117.50/288 (40.80%) on average (median 118, minimum 94) and a top-5 average mismatch of 123.37/288 (42.84%). Home conditional generation produces comparatively closer sequences, with a top-1 mismatch of 62.30/288 (21.63%) on average (median 54, minimum 7), which is expected because conditional generation can converge to training-distribution days that satisfy the given constraints; nevertheless, no exact reproductions occur (distance = 0 for all samples).

5.5. User Evaluation

This section evaluates the subjective quality of daily activity plans generated by ORACLE in comparison to three benchmark models, as discussed in section 5.2. The evaluation adopts the same dual scenarios of random generation and conditional generation.

Eleven participants (7 males, 4 females, age range 22-29, $\mu = 24.36$, $\sigma = 2.062$) participated in the study. Participants evaluated a total of 36 data samples, randomly se-

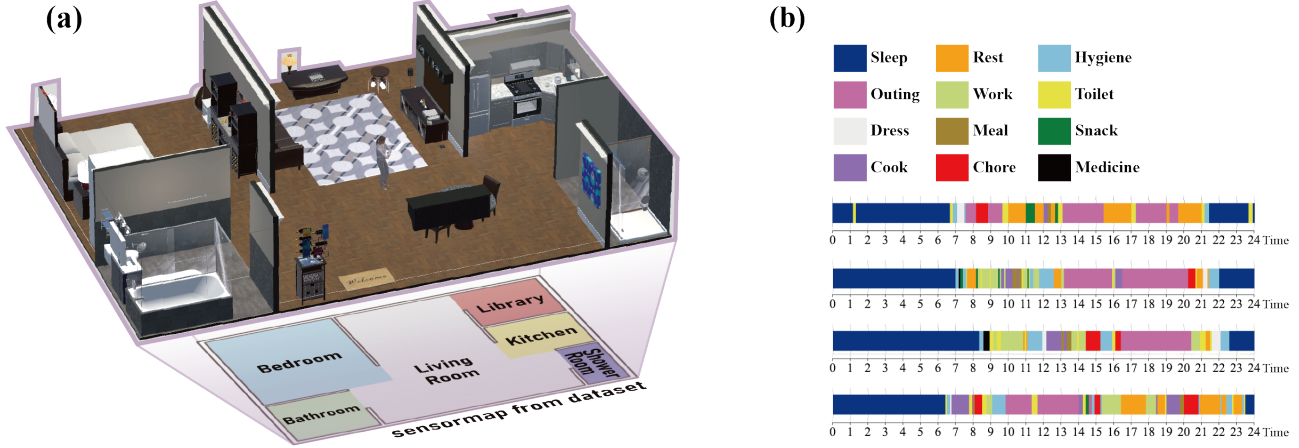


Figure 3. (a) Unity virtual environment where an NPC executes activities. (b) Samples of the visualized plan.

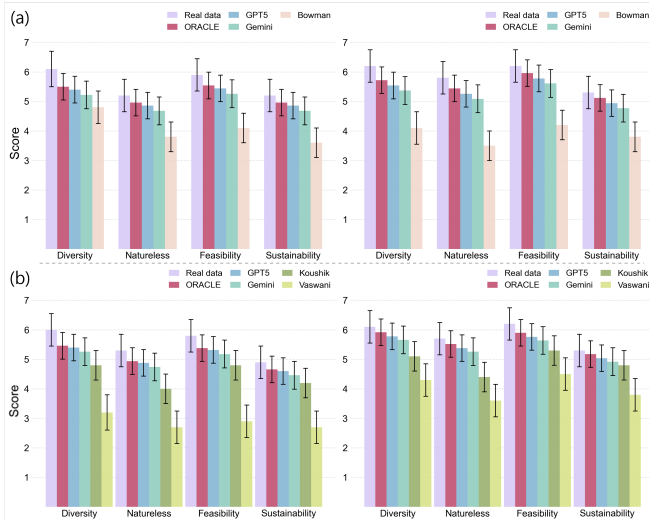


Figure 4. User study evaluation results. (a) Random generation scenario results in Apartment and Home datasets. (b) Conditional generation scenario results in Apartment and Home datasets. Real data is sampled from the test dataset.

lected and visualized, combining both the random generation and conditional generation scenarios in both Apartment and Home datasets. The evaluation involved data visualization and a 2.5-minute virtual environment simulation for each sample. The virtual environment is illustrated in Figure 3(a), and the data visualization is shown in Figure 3(b). They then rated four research questions on a 7-point Likert scale: “*Diversity*: Are these activities diverse and varied?”, “*Naturalness*: Do these activities resemble how a person would do within a day?”, “*Feasibility*: Do you think these activities could be done by someone?”, and “*Sustainability*: Could you see yourself living comfortably for a week or more following these plans?”

The analysis results of the random and conditional gen-

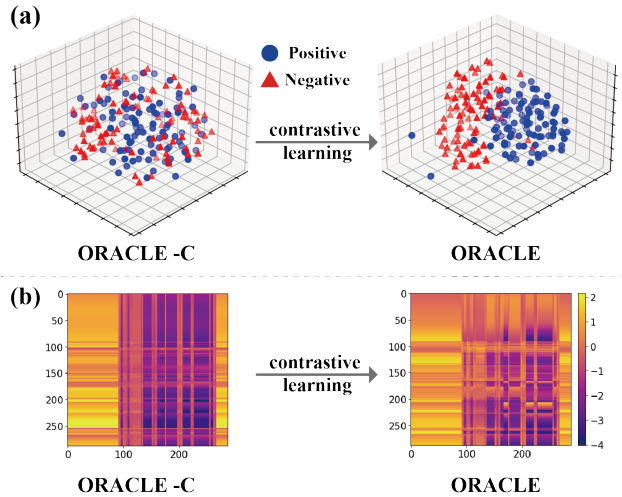
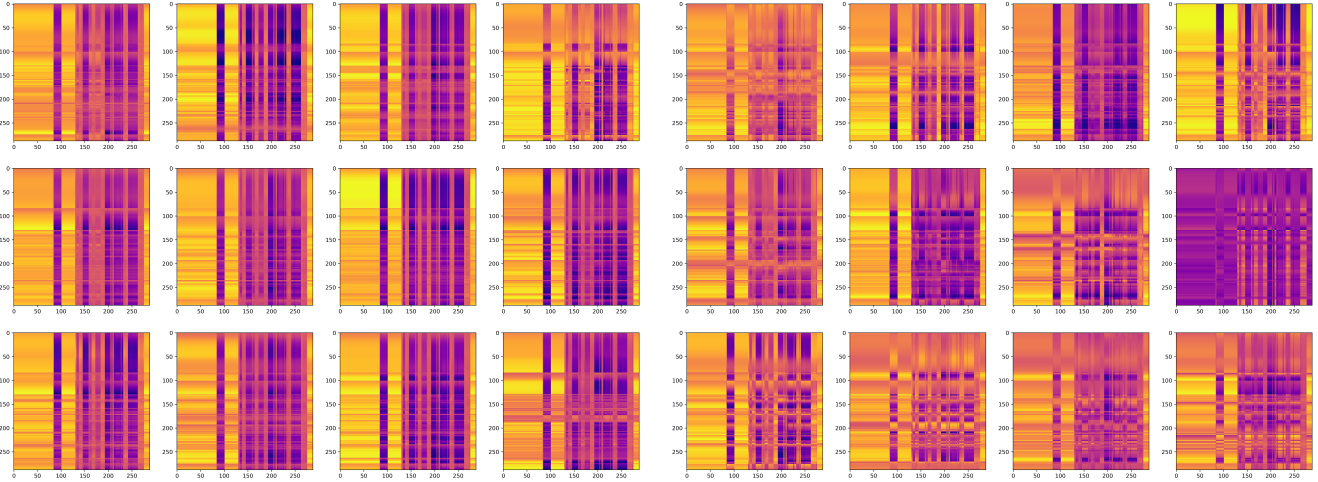


Figure 5. (a) Latent visualization of ORACLE and ORACLE -C. (b) Attention map of ORACLE and ORACLE -C.

eration scenarios in Apartment and Home datasets are presented in Figure 4. When comparing ORACLE to Bowman., ORACLE achieves slightly higher scores across all metrics. These findings align with the LLM score’s validity tested in section 5.2 and demonstrate that ORACLE generates more human-like and credible activity plans. Against the models by Vaswani and Koushik, ORACLE demonstrates superior performance, indicating its proficiency in crafting plausible schedules, even when they are partially pre-scheduled. We additionally compared ORACLE with two recent LLMs [1, 14] using the same prompt: “*Generate a 24-hour human-like daily behavior sequence in 5-minute increments (288 values total) using Sleep, Outing, Rest, Work, Hygiene, Toilet, Dress, Cook, Meal, Chore, Snack, Medicine, output in the format HH:MM~HH:MM [activity].*” As these LLMs were not trained on activity datasets, only qualitative evaluation was performed,



(a) Attention maps for ORACLE -C. (b) Attention maps for ORACLE.
 Figure 6. Attention map visualizations of ORACLE and ORACLE -C.

where participants rated ORACLE higher.

Our user study provides a subjective perspective on the generated plans, although its scale and scope are inherently limited. We therefore present it as a complementary evaluation alongside our main quantitative results. In particular, we report plausibility-based REAL scores, distributional similarity via WD, and long-range diversity via Distinct- n , which capture different aspects of realism and variability in daily routines. The overall trends observed in the user study are broadly consistent with these objective metrics, providing additional support that ORACLE generates realistic and diverse activity plans.

5.6. Ablation Study

We conducted an ablation study to evaluate the impact of contrastive learning and Transformer architecture on ORACLE. We compared the standard ORACLE model to a variant without contrastive learning (ORACLE -C) and another with an LSTM instead of a Transformer (ORACLE -T).

Table 4 shows that standard ORACLE outperforms all variants, presenting the effectiveness of contrastive learning and Transformer architecture. To further investigate the effect of contrastive learning, we visualized latent space and attention maps for both ORACLE and ORACLE -C, as depicted in Figure 5(a) and (b).

In latent space visualization, test dataset samples were identified as positive samples, whereas generated data failing to meet the established criteria were identified as negative samples. These samples were then rendered using t-SNE. The visualization revealed overlaps between positive and negative samples in ORACLE -C, indicating potential exposure bias. In contrast, ORACLE’s contrastive learning improved sample discrimination, reducing bias and enhancing generalization across diverse data scenarios.

In attention map visualization, test dataset samples were selected at random during the task of reconstruction, focusing particularly on the outputs from the final cross-attention layer’s 12 heads. The visualization revealed ORACLE’s maps to be more varied than ORACLE -C, indicating superior generalization to unseen scenarios during training.

5.7. Further Visualization Results

Figure 6 showcases the attention maps obtained from the 12 heads of the final cross-attention layer. Prior to the application of contrastive learning, attention maps exhibit a uniform focus. Conversely, after the incorporation of contrastive learning, the model exhibits a more diverse focus, with each head concentrating on different aspects of the data. This indicates that the model analyzes the data from various perspectives at the level of individual heads, suggesting the potential of contrastive learning to enhance the richness and diversity of the outcomes.

Figure 7 provides additional daily activities generated by our model. Analysis of these results reveals that both random and conditional generations exhibit similarities to the reference data, indicating that our model has effectively learned the data’s distribution. This resemblance serves as evidence of the model’s successful training.

We provide an example video in the supplementary material to demonstrate a minimal proof-of-concept pipeline that maps each predicted activity token to a simple in-engine action and location transition in a Unity apartment scene. The purpose of this visualization is to show the usability of ORACLE as an activity-plan generator that can drive an NPC controller, rather than to present a complete animation set or high-fidelity character behaviors. Thus, the limited action variety in the current video reflects the simplicity of the demonstration mapping, not a limitation of the

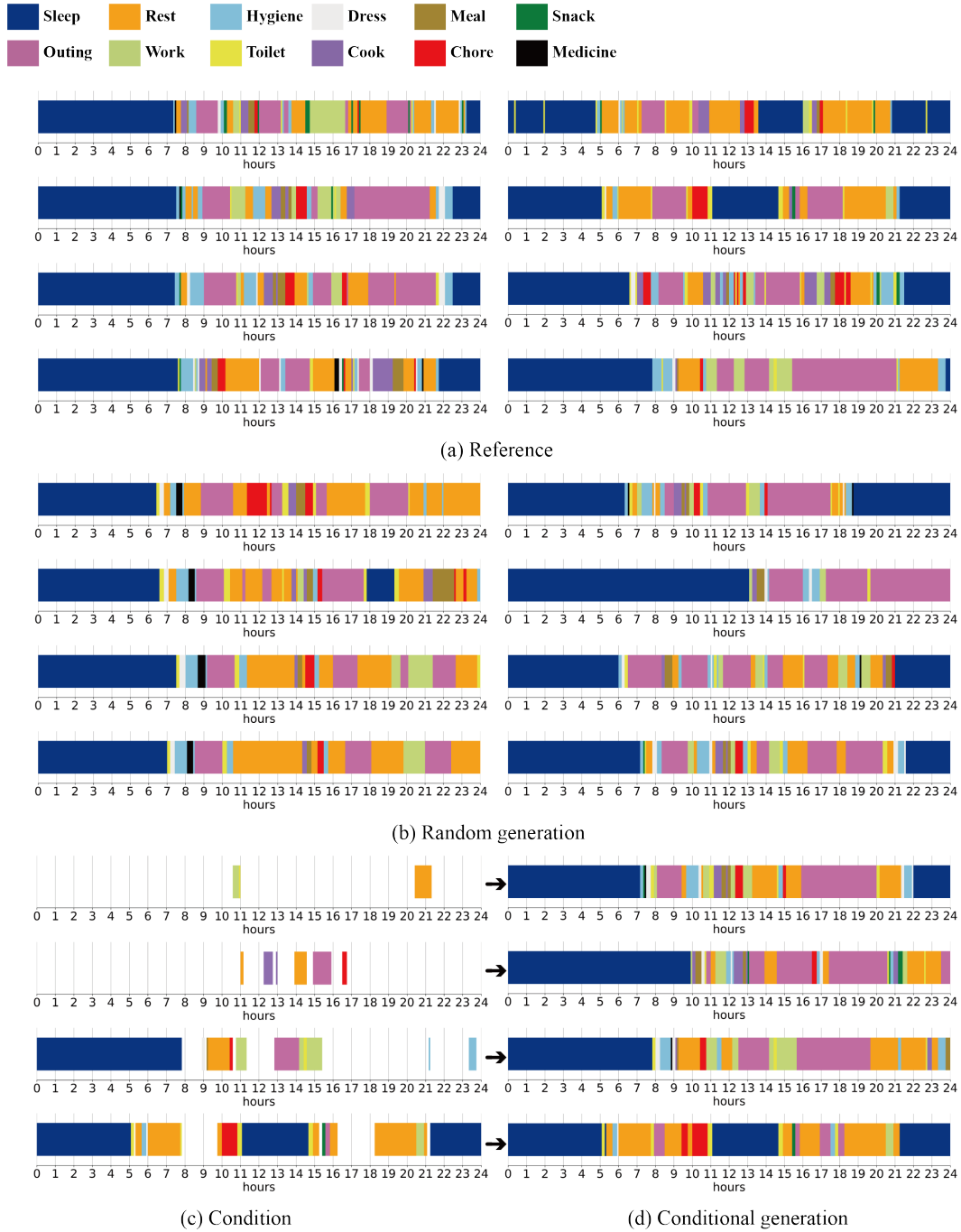


Figure 7. The visualization of daily activities generated by our model.

generated plans. Importantly, the activity-to-action mapping is modular and can be readily extended with richer interaction primitives (e.g., object-level affordances, state machines, motion libraries, or behavior trees), enabling more diverse and convincing executions without changing the underlying planning model.

6. Conclusion

In this paper, we introduce ORACLE, a novel approach for generating plausible daily activity plans for NPC utilizing a Transformer-CVAE model with contrastive learning. Our experiments confirm that ORACLE outperforms existing models in generating both complete schedules from scratch and enhancing existing ones with additional activi-

ties.

ORACLE offers significant potential for applications in enhancing smart home automation, supporting systems for the elderly, aiding in urban planning, and streamlining facility management through its predictive capabilities for personal daily routines. Furthermore, the training framework of ORACLE has broader applicability in other domains requiring sequential data generation, such as aligning events in games, simulating urban designs, and automating production processes in manufacturing settings.

Despite its achievements, our approach has a limitation. ORACLE primarily facilitates the planning of generalized behaviors inferred from the smart home dataset, thus restricting its capacity to devise personalized activities or predict behaviors occurring outside the home environment. The most direct strategy to address this challenge would involve sourcing datasets that encapsulate a wider array of human behaviors. However, the acquisition of such comprehensive data presents significant challenges, rendering this solution impractical. An alternative method to overcome this may entail the integration of reinforcement and deep learning strategies to simulate human decision-making processes, thereby accommodating a wider variety of situational decisions. We will direct our future research to simulate a broader range of human behaviors across various environments.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) (project management number granted from IRIS) (No. RS-2025-24802983, 50%). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2025-00518643, 50%).

References

- [1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 4, 9
- [2] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. SRI, A. Barrett, D. Christianson, et al. Pddl—the planning domain definition language. *Technical Report, Tech. Rep.*, 1998. 2
- [3] E. F. Anderson, L. McLoughlin, F. Liarokapis, C. Peters, P. Petridis, and S. De Freitas. Developing serious games for cultural heritage: a state-of-the-art review. *Virtual reality*, 14:255–275, 2010. 2
- [4] N. Avradinis, T. Panayiotopoulos, and G. Anastassakis. Behavior believability in virtual worlds: agents acting when they need to. *SpringerPlus*, 2(1):1–11, 2013. 1, 2
- [5] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966. 3
- [6] R. Bellman and R. Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9, 1959. 3
- [7] A. Bogdanovych and T. Trescak. To plan or not to plan: lessons learned from building large scale social simulations. In *Intelligent Virtual Agents: 17th International Conference, IVA 2017, Stockholm, Sweden, August 27-30, 2017, Proceedings 17*, pages 53–62. Springer, 2017. 1
- [8] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*, pages 10–21. Association for Computational Linguistics (ACL), 2016. 3, 7
- [9] M. Bratman. Intention, plans, and practical reason. 1987. 2
- [10] L. Chen, K. Bechkoum, and G. Clapworthy. Equipping a lifelike animated agent with a mind. In *Intelligent Virtual Agents: Third International Workshop, IVA 2001 Madrid, Spain, September 10–11, 2001 Proceedings 3*, pages 72–85. Springer, 2001. 2
- [11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3
- [12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 3
- [13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 3
- [14] G. Comanici, E. Bieber, M. Schaeckermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 9
- [15] D. Cook, A. Crandall, and B. Thomas. Human Activity Recognition from Continuous Ambient Sensor Data. UCI Machine Learning Repository, 2019. DOI: <https://doi.org/10.24432/C5D60P>. 2
- [16] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan. Casas: A smart home in a box. *Computer*, 46(7):62–69, 2012. 2
- [17] E. de Sevin and D. Thalmann. A motivational model of action selection for virtual humans. In *International 2005 Computer Graphics*, pages 213–220. IEEE, 2005. 1, 2
- [18] L. De Silva, F. R. Meneguzzi, and B. Logan. Bdi agent architectures: A survey. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI), 2020, Japan.*, 2020. 2
- [19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3
- [20] F. Dignum, D. Morley, E. A. Sonenberg, and L. Cavedon. Towards socially sophisticated bdi agents. In *Proceedings*

- fourth international conference on multiagent systems, pages 111–118. IEEE, 2000. 2
- [21] L. Fang, T. Zeng, C. Liu, L. Bo, W. Dong, and C. Chen. Transformer-based conditional variational autoencoder for controllable story generation. *arXiv preprint arXiv:2101.00828*, 2021. 3
- [22] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971. 2
- [23] M. Fox and D. Long. Pddl2. 1: An extension to pddl for expressing temporal planning domains. *Journal of artificial intelligence research*, 20:61–124, 2003. 2
- [24] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: theory and practice*. Elsevier, 2004. 2
- [25] L. Gramoli, J. Lacoche, A. Foulonneau, V. Gouranton, and B. Arnaldi. Needs model for an autonomous agent during long-term simulations. In *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 134–138. IEEE, 2021. 2
- [26] L. Gramoli, J. Lacoche, A. Foulonneau, V. Gouranton, and B. Arnaldi. Control your virtual agent in its daily-activities for long periods. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 203–216. Springer, 2022. 2
- [27] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013. 3
- [28] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052. IEEE, 2005. 3
- [29] M. F. Hasani and Y. Udjaja. Immersive experience with non-player characters dynamic dialogue. In *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, volume 1, pages 418–421, 2021. 1
- [30] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 3
- [31] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998. 3
- [32] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. 3
- [33] H. Jang, S. Hao, P. M. Chu, P. K. Sharma, Y. Sung, and K. Cho. Deep q-network-based multi-criteria decision-making framework for virtual simulation environment. *Neural Computing and Applications*, 33:10657–10671, 2021. 2
- [34] O. Jeff. Applying goal-oriented action planning to games. *AI game programming wisdom*, 2:217–228, 2003. 2
- [35] N. S. Johansen, L. B. Kær, J. A. B. Stolberg, R. G. Tollund, N. Hyldeg, P. Oktober, and A. Torralba. Towards believable non-player characters with domain-independent planning. In *2022 workshop on scheduling and planning applications workshop*, 2022. 2
- [36] J.-P. Kelly, A. Botea, and S. Koenig. Offline planning with hierarchical task networks in video games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 4, pages 60–65, 2008. 2
- [37] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022. 3
- [38] A. Koushik, M. Manoj, N. Nezamuddin, and A. Prathosh. Activity schedule modeling using machine learning. *Transportation Research Record*, page 03611981231155426, 2023. 2, 3, 7
- [39] P. Krümpelmann, M. Thimm, G. Kern-Isberner, and R. Fritsch. Motivating agents in unreliable environments: A computational model. In *Multiagent System Technologies: 9th German Conference, MATES 2011, Berlin, Germany, October 6-7, 2011. Proceedings 9*, pages 65–76. Springer, 2011. 2
- [40] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In K. Knight, A. Nenkova, and O. Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016. Association for Computational Linguistics. 8
- [41] Y. Liu and P. Liu. Simcls: A simple framework for contrastive learning of abstractive summarization. *arXiv preprint arXiv:2106.01890*, 2021. 3
- [42] M. Maroto-Gómez, Á. Castro-González, M. Malfaz, and M. Á. Salichs. A biologically inspired decision-making system for the autonomous adaptive behavior of social robots. *Complex & Intelligent Systems*, pages 1–19, 2023. 2
- [43] M. Mateas. An oz-centric review of interactive drama and believable agents. In *Artificial intelligence today: Recent trends and developments*, pages 297–328. Springer, 2001. 1
- [44] R. J. McCall, S. Franklin, U. Faghihi, J. Snaider, and S. Kugele. Artificial motivation for cognitive software agents. *Journal of Artificial General Intelligence*, 11(1):38–69, 2020. 2
- [45] J. S. Park, J. C. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulators of human behavior. *arXiv preprint arXiv:2304.03442*, 2023. 2
- [46] S. Parsons and P. Giorgini. An approach to using degrees of belief in bdi agents. *Information, Uncertainty and Fusion*, pages 81–92, 2000. 2
- [47] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018. 3
- [48] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015. 3
- [49] G. Robertson and I. Watson. A review of real-time strategy game ai. *Ai Magazine*, 35(4):75–104, 2014. 2
- [50] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 3

- [51] M. Vanhulsel, D. Janssens, G. Wets, and K. Vanhoof. Simulation of sequential data: An enhanced reinforcement learning approach. *Expert Systems with Applications*, 36(4):8032–8039, 2009. [2](#)
- [52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. [3](#), [7](#)
- [53] T. Zhao, R. Zhao, and M. Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In R. Barzilay and M.-Y. Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada, July 2017. Association for Computational Linguistics. [3](#)

Input prompts

I'd like to design daily human activities with a defined set {Sleep, Outing, Rest, Work, Hygiene, Toilet, Dress, Cook, Meal, Chore, Snack, Medicine}. I need you to establish numerical parameters for these activities. These parameters must be commonly used for all activities. These parameters should relate to the sequence of activities throughout the day without being tied to specific times. Please recommend parameters and their desired values, drawing on general knowledge about realistic human daily activities.

GPT outputs

Given the scope of your request, I will focus on two parameters that can be universally applied to the activities listed: Duration and Frequency. These parameters are crucial for planning and can accommodate the variability of daily routines. Below are the recommended values based on general knowledge and assumptions about an average adult's daily activities. Please note that these are guidelines and actual values may vary based on individual needs and lifestyle.

Sleep; Duration: 7-9 hours (Per night), Frequency: 1 time per day

Outing; Duration: 1-4 hours (Per outing), Frequency: 1-2 times per day

...

Snack; Duration: 5-15 minutes (Per snack), Frequency: 2-3 times per day

Medicine; Duration: 1-5 minutes (Per session), Frequency: 1-3 times per day (based on prescription)

These values provide a foundational framework for designing a daily schedule, considering a balance between obligations and personal care. Adjustments may be necessary to tailor the schedule to individual needs, preferences, and specific life circumstances.

Table 5. Input text prompts and GPT outputs for data filtering.

Question 1: Do you consider the criteria presented to be the minimum standards for sleep time and frequency for a normal person's sustainable life?

Answer 1: People should sleep for at least 5 hours and up to 12 hours per day.

Question 2: Do you consider the criteria presented to be the minimum standards for outing time and frequency for a normal person's sustainable life?

Answer 2: People should go out for up to 12 hours a day.

Question 3: Do you consider the criteria presented to be the minimum standards for rest time and frequency for a normal person's sustainable life?

Answer 3: People take a rest for up to 12 hours a day.

Question 4: Do you consider the criteria presented to be the minimum standards for work time and frequency for a normal person's sustainable life?

Answer 4: People work for up to 12 hours a day.

Question 5: Do you consider the criteria presented to be the minimum standards for hygiene activities time and frequency for a normal person's sustainable life?

Answer 5: People do hygiene activities such as taking a shower at least once a day, for up to 1 hour and 30 minutes at a time.

Question 6: Do you consider the criteria presented to be the minimum standards for toilet time and frequency for a normal person's sustainable life?

Answer 6: People use the toilet at least once a day, for up to 30 minutes at a time.

Question 7: Do you consider the criteria presented to be the minimum standards for dress time and frequency for a normal person's sustainable life?

Answer 7: People change clothes at least once a day, for up to 1 hour at a time.

Question 8: Do you consider the criteria presented to be the minimum standards for cooking time and frequency for a normal person's sustainable life?

Answer 8: People cook for up to 2 hours at a time.

Question 9: Do you consider the criteria presented to be the minimum standards for meal time and frequency for a normal person's sustainable life?

Answer 9: People have meals at least once a day, for up to 2 hours at a time.

Question 10: Do you consider the criteria presented to be the minimum standards for chore time and frequency for a normal person's sustainable life?

Answer 10: People perform household chores such as cleaning and dishwashing at least once a day, for up to 2 hours at a time.

Question 11: Do you consider the criteria presented to be the minimum standards for snack time and frequency for a normal person's sustainable life?

Answer 11: People snack for up to 2 hours at a time.

Question 12: Do you consider the criteria presented to be the minimum standards for medicine time and frequency for a normal person's sustainable life?

Answer 12: People take medication for up to 2 hours at a time.

Table 6. Survey questions assessing the filtering criteria for each activity.

Input prompts

We want to evaluate if the following [number] indoor daily schedules resemble a human's daily routine. Please rate it on a scale from 0 to 100 for each. Tell me the score with further explanation. And find the average of those scores.

00:00~00:05 Rest 00:05~00:15 Chore 00:15~00:20 Dress 00:20~00:35 Hygiene 00:35 08:25 Sleep 08:25~08:30 Toilet 08:30~12:10 Sleep 12:10~12:25 Hygiene 12:25~12:35 Dress 12:35~12:40 Rest 12:40 13:10 Outing 13:10~13:50 Rest 13:50~14:40 Outing 14:40~16:00 Rest 16:00~16:20 Work 16:20~17:55 Rest 17:55~18:10 Hygiene 18:10~18:20 Work 18:20~18:45 Outing 18:45~20:40 Rest 20:40~24:00 Outing

...

GPT outputs

To evaluate these schedules on a scale from 0 to 100 for how closely they resemble a human's daily routine, I considered several factors:

Sleep Patterns: Regularity and total duration of sleep. Activity Distribution: Balanced and realistic distribution of work, chores, meals, and rest. Consistency: Predictability and normalcy in daily routines, avoiding erratic or excessive repetition of certain activities. Social and Leisure Activities: Time allocated for social interactions, leisure, and outings. Overall Realism: How the entire schedule feels when compared to an average person's day.

Example Scores:

00:00~00:05 Rest 00:05~00:15 Chore 00:15~00:20 Dress 00:20~00:35 Hygiene 00:35~08:25 Sleep...

Score: 40/100: Unusual sleep pattern with significant sleep during morning hours, minimal time for meals and work, and unrealistic hygiene frequency.

00:00~07:10 Sleep 07:10~07:20 Hygiene 07:20~07:25 Snack...

Score: 65/100: A more realistic distribution of activities with decent sleep and work patterns, though still some odd timing for hygiene and meals.

00:00~06:00 Sleep 06:00~06:10 Hygiene 06:10~06:20 Dress...

Score: 70/100: A fairly standard schedule with adequate sleep, hygiene, and work balance, though some activities like multiple short rests are slightly odd.

00:00~09:45 Sleep 09:45~09:55 Hygiene 09:55~10:10 Cook...

Score: 75/100: A good distribution with realistic blocks for sleep, work, and meals, though a bit too much time in outings.

...

an estimated average score across all schedules could be around **50/100**, reflecting a mix of schedules that somewhat resemble human routines

...

Table 7. Input text prompts and GPT outputs for LLM score.