Understanding In-home Routines through Spatio-temporal Object Tracking for Proactive Assistance

Maithili Patel[1] and Sonia Chernova[1]

***Abstract*— Proactivity in robotic assistance is valued in domains like elder-care and caring for patients with mental disorders, but robot assistants today require exact task or goal descriptions, and can proactively assist only the user's ongoing activity. We propose to formulate the problem of long-horizon proactive assistance as one of learning temporal patterns in object movements resulting from the user's daily routines, then using this learnt model to predict future object movements, which the robot can do instead. In this paper, we create a unified spatio-temporal object dynamics model, based on a generative graph neural network to learn a predictive model using temporal sequences of object arrangements, represented as scene graphs. We identify the lack of a dataset to train and evaluate such a model on, and collect a behavioral dataset in simulation, which reflects object-interaction over normal daily routines. Our model outperforms the baselines on predicting future object locations, with a 42% average increase in F-1 score on predicting which objects will move in a certain predictive window, and 20% average increase in precision on predicting the correct object destination, implying that our method would enable an assistive robot to proactively help the user by moving objects in accordance with the user's needs.**

I. Introduction

In today's aging society, the care-giving burden is rapidly growing, and robots have the potential to not only shoulder some of that burden, but to boost a sense of self-sufficiency among users, enabling elders to live in their homes independently for longer [1]. Towards being more effective caregivers, robots should be able to anticipate user needs, proactively assist without being asked, and adapt to changing user abilities. Studies on companion service robots for the elderly [2] and those suffering from mild cognitive impairments [3], [4] show that proactive behavior and initiative from the robot are highly valued by the users.

To provide proactive assistance, a robot needs to have a generalizable semantic understanding of various objects and their usage, and other agents' intentions, actions and preferences. For instance, if a robot understands that the user likes to have cereal for breakfast at around 8am, and that they need a cereal box and bowl to do so, the robot can take those objects out on the counter ready for the user to use, and afterwards, if the user forgets to put the cereal box back into the cabinet, a similar understanding that the user should be done using the cereal box can enable the robot to put it back for the user. Our aim through this work is to understand patterns in temporal daily routines of users to plan assistive actions.

This work is sponsored in part by NSF IIS 2112633.

Georgia Institute of Technology, Atlanta, Georgia, United States. maithili,chernova@gatech.edu

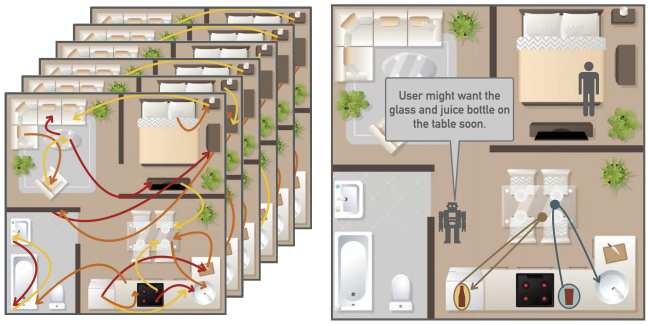


Fig. 1: The robot learns patterns in user behavior, and their effects on object movements from past observations. At runtime, the robot predicts object relocations, such that the objects would be in locations where the user will need them in the near future.

Proactive assistance based on user observations has been developed in prior work using activity recognition, action prediction, and planners [5], [6], [7]. For example, a robot might recognize that the user is making a salad, predict that they will need lettuce, and grab it for them. However, such systems are limited to assisting with user's current activity, and assume the ability to recognize each user action. They are difficult to scale to longer time scales, as it is difficult for a robot to observe each human action and intractable to recognize them with all their nuances.

We propose a novel perspective on this problem. As humans go through their daily routines, their activities affect the objects around them, leaving footprints in the location and state changes of these objects. For example, in the morning, someone who has cereal for breakfast might take out cereal and milk to the counter and afterwards leave the used bowl in the sink as shown in Figure 2. As such, patterns in our daily routines are reflected in the movement of objects in the household, and can be learned by observing such movements. We propose to leverage the learned dynamics of objects resulting from the user's routine, to develop an actionable understanding of these routines. Information from objects in the environment has been shown to be a useful indicator towards reconstructing past agent actions [8], and in Inverse Reinforcement Learning to train policies based on human preferences detectable in the environment state [9]. Through an object-based perspective on understanding routines, we overcome the constraint of having to constantly observe the human, and the need to classify nuanced actions.

Explicitly modeling object movements, and learning pat-

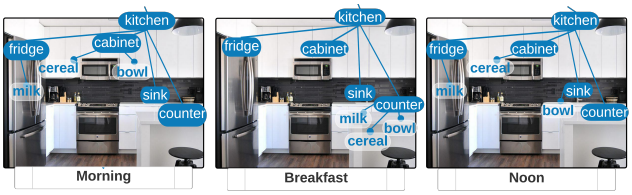


Fig. 2: Evolution of object arrangement through breakfast

terns therein, has the benefit of providing actionable object-level information for downstream assistive action planners. If an embodied agent can predict a future state of object arrangement, it can infer the expected changes in object locations, and relocate the objects in accordance with it. Hence, we need to create a unified spatial and temporal object dynamics model, which can learn patterns from observed object movements. A lack of datasets representing such object dynamics to learn from and/or evaluate on makes this problem trickier. In this work, in addition to formulating our problem in this way, we address both these gaps. Our primary contributions are as follows:

- Formulation of proactive assistance through object dynamics modeling
- A behavioral simulation dataset reflecting normal routines and their effect on household objects
- A generative graph neural network based spatio-temporal object dynamics model

Our object dynamics model outperforms the baselines on predicting future object locations, with a 42% average increase in F-1 score on predicting which objects will move in a certain predictive window, and 20% average increase in precision on predicting the correct object destination.

II. Prior Work

Our work involves generating an activity dataset, as well as creating a predictive model that can learn from object movement observations, for which we employ a graph neural network. In this section, we introduce literature relevant to each of these facets of our work.

A. Activity Datasets

The smart homes and intelligent devices community have datasets [10] extending to order of months and scaling to entire houses or offices. However, they are usually collected using smart sensors which only provide coarse user location and are annotated with high-level activity labels like reading, sleeping, cooking etc. They lack object interaction information needed for our work.

Datasets created for learning to recognize activities from videos [11] contain detailed object interaction-level information. These datasets however are limited to the specific activity, extending to time scales of minutes, and only a small subset of the objects in an environment. The VLOGs dataset [12] extend to the duration of a part of a day, like a morning routine, but because the videos are mined from youtube, these are not continuous, thus missing some actions. For instance, application of makeup is covered in great detail, but the act of putting away cosmetics is skipped. The missed actions are typically associated with putting things away after use, which are crucial for an assistive robot to learn.

Datasets have been created in simulation environments for different activities [13], but these also are limited to the extent of specific activities. We use the same VirtualHome simulator to create a dataset that accurately reflects normal daily routines, is complete in its cataloguing of object interactions, and extends to a longer time horizon of the order of days.

B. Object Modeling

Spatio-temporal object modeling closely relates to and extends prior approaches for semantic mapping [14], which enrich traditional metric and topological maps of households and other environments by supplementing their description with high-level information about space purpose, utilization, object locations, object state etc. Semantic mapping, however, focuses on consolidating past observations into a single estimate, as opposed to trying to predict how things will change in the future.

Informed object search applications have led to methods on modeling beliefs over object locations. Such methods usually leverage probabilistic methods combining prior knowledge, observations, and known constraints [15]. Newer methods [16] leverage correlational information from datasets to generate semantic priors on the likelihood of inter-object relations, to inform the belief over possible object locations. Other methods [17], [18] leverage past experience in the environment to model a temporal function of existence of objects of interest in a location.

The information obtained from inter-object relationships is complementary to the observed location history, and hence we seek to combine both into a unified spatio-temporal model of the object dynamics. Spatio-temporal graphs have been modeled in the robotics community by indirect methods such as flattening into vectors that can be learnt using RNNs [19]. We maintain the graphical structure of our data, and represent the object arrangement at a given time using a scene graph, which is a directed graph with objects as nodes, and inter-object relations as edges. Scene graphs have been used for semantic reasoning over images [20], representing objects in a map [21], etc. Scene graph representations of objects in video frames have also been used to identify activities [22]. We learn the temporal dynamics over this space of scene graphs, using a Graph Neural Network.

C. Graph Neural Networks

Our problem requires a generative graph network, conditioned on a graph as well as some global context to encode time. We derive inspiration from existing work on generative graph methods conditioned on graphs, mainly in the domain of molecular biochemistry. These methods either use encoder-decoder frameworks [23], [24], [25], or step-wise modification-based methods to convert the input graph to the output [26], [27], [28]. Since, predicting object locations involve very few changes on consecutive time steps, we

deem the step-wise edit networks suitable for our application. Some modification-based networks employ domain-specific heuristics and rewards for using reinforcement learning to learn a modification policy [26], [28]. Due to lack of such heuristics, we derive inspiration from a general graph translation method, Node-Edge Co-evolving Deep Graph Translator (NEC-DGT) [27] to learn our predictive model only based on data. The focus of such methods on edge attributes is important for our use, since we need to leverage existing relational information in the scenes to be able to predict future relations. Recent work towards emphasizing information on edges define a dual-graph by flipping the edges and nodes [29], to incorporate message passing between edges that have a node in common. Our method and prior work [27] perform a similar operation through edge-to-edge message passing without explicitly defining a dual-graph.

Graph neural networks are traditionally static in time, and were only recently adapted to model temporally evolving data [30]. Existing networks have been extended to include time, by including LSTM units [31], alternating message passing along the graph and discrete time axis for each node [32], using a memory structure to retain past information [33], and learning functional time embeddings similar to Time2Vec [34] for using self attention [35]. To explicitly account for known periodicities in data, and directly influence predictions using data from previous day, week, etc. in addition to the recent information, prior work [36] maintain LSTMs over all these cadences. Such a system has periodicities rigidly defined in its structure, and hence its ability to learn patterns is strictly limited to known periods.

For traditional graph neural network applications, only the relative value of timestamps of two graphs is important to determine how they relate, whereas our model also needs to derive information from the absolute value of time. For instance, we aim to learn not only that the user usually has breakfast after brushing their teeth, but also that they have breakfast around 8am. We provide our model an embedding of the absolute time inspired by Time2Vec [34], but with known frequencies.

III. Problem Formulation

We model an environment as consisting of a fixed set of objects $\mathcal{O}$ and locations $\mathcal{L}$. At any given time t, we model the state of the environment, X_t, as an unordered list of object-location pairs (o_i, l_i) representing the placement of object $o_i \in \mathcal{O}$ at location $l_i \in \mathcal{L}$. We assume each object can exist in only one location at a time, and that objects of the same class (e.g., one of multiple *cereal* instances) are uniquely identifiable. The state of the environment can be modified by an embodied agent (human or robot) by performing a *relocation* action $r(o, l_1, l_2)$ to move object $o \in \mathcal{O}$ from location l_1 to location l_2.

Based on the above formulation, we model the object relocation problem as consisting of two parts. First, given a set of previous observations of the environment $X_{0:M}$[1] over some time span M, learn the model $\Phi(X_t, t) \rightarrow \hat{X}_{t+\delta}$ that takes the current time t and the state of the environment X_t and predicts the future state of the environment $\hat{X}_{t+\delta}$ some fixed δ timesteps in the future. Second, we desire the function $\Psi(X_t, \hat{X}_{t+\delta}) \rightarrow \mathcal{R}$ which returns the set of relocations $\mathcal{R}$ required to transition the environment from X_t to $\hat{X}_{t+\delta}$.

[1]Which we expect to span multiple periods of the periodic patterns in object movements.

IV. Spatio-Temporal Object Dynamics Model

We aim to create a unified spatial and temporal model of object dynamics, which, given sequential observations of object locations can learn to predict future object movements. We represent the environment state, X_t, at a given time t as a directed graph, $G_t = \{V, E\}$, consisting of vertices/nodes representing objects in the environment, $V = \{v_i\}$, and edges capturing spatial inter-object relationships, $E = \{e_{i,j}\}$. We represent each object instance as a node, and use the terms object and node interchangeably.

Our spatial directed graph consists of edges connecting objects to their locations. We ensure that the graph contains only non-redundant information by removing edges that can be deduced from other edges. For instance, if we know *apple in bowl*, and *bowl on table* to be true, then we can deduce *apple on table*, hence we remove that edge from our formulation. This leaves our scene graph with a single edge originating from every node, representing its most specific location, for instance *bowl* for *apple* in the above example. Such a structure is called an in-tree, or anti-arborescence, since every constituent node has a single out-edge, signifying every object having a single location: $N_i^+ = \{v_j | e_{i,j} \in E\}$, $|N_i^+| = 1$.

Having represented a discrete time-slice of the object dynamics as an in-tree, $X_t = G_t$, we model Φ with respect to this graph representation. Specifically, we learn

$$\Phi(G_t, t) \rightarrow p(\tilde{G}_{t+\delta})$$

which predicts a probabilistic graph representing the state at a future time step given the current graph and time. $p(\tilde{G}_{t+\delta})$ is fully connected, with edge weights representing probabilities of edge existence, and hence the weights of all edges originating at a node sum to one. This graph represents the probability distribution over all possible in-trees, and we can infer the posterior, $\hat{G}_{t+\delta}$, by picking the most likely out-edge for each node:

$$\hat{G}_{t+\delta} = \arg\max_{\tilde{G}_{t+\delta}} \; p(\tilde{G}_{t+\delta})$$

A. Model Architecture

To model the dynamics Φ over scene graphs, we create a graph translation model inspired by prior work [27]. The complete architecture is outlined in Figure 3. Traditionally, GNNs operate over graph topologies built to express similarity between nodes, where message passing serves to smooth out information across nearby nodes. Our edges, however, represent physical relationships, and hence, a crucial insight from such a system performing well is that a message passing formulation over edges representing physical relationships can help inference on spatial graphs.

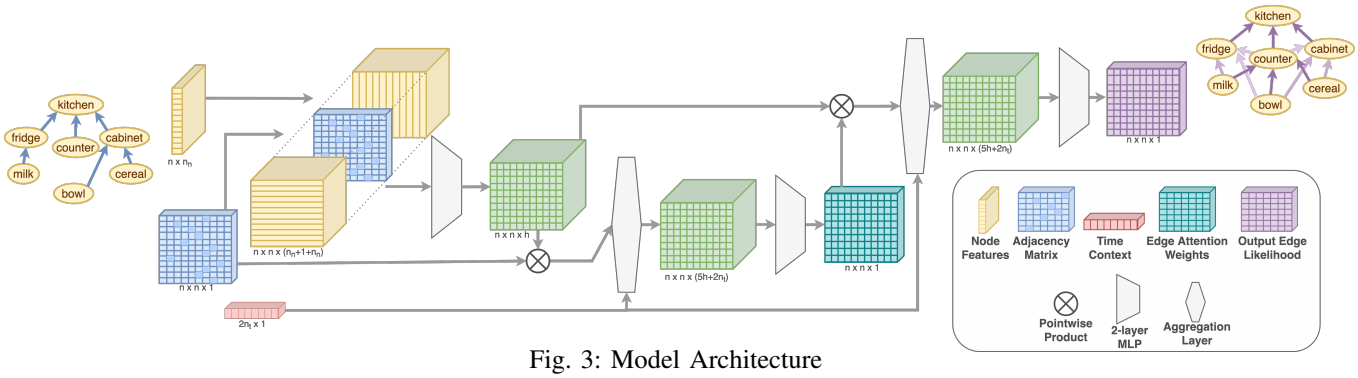


Fig. 3: Model Architecture

Our model produces a probabilistic directed graph as the prediction for next state. The inputs to our model are the embeddings for each node and the edge existence/adjacency matrix of the input graph, as well as an encoding of time. We use one-hot encodings for each object as the node embeddings, resulting in a vector with length equal to the total number of objects ($|\mathcal{O}|$). Our time encoding is similar to that proposed by Time2Vec [34], which learns a given number of frequencies, and uses sinusoidal functions of time over those frequencies. However, rather than having the network learn the frequencies from data, we use pre-specified frequencies, and use both sines and cosines for our periodic functions. We use frequencies corresponding to time periods τ_i of 1-day, 12 hours, 6 hours, 3 hours, 1 hour, 30 mins, and 10 mins to induce meaningful priors on periodicity of human activities, to generate the time encoding (T) of length $2n_\tau$

$$T = \left[sin\left(\frac{2\pi t}{\tau_i}\right), cos\left(\frac{2\pi t}{\tau_i}\right), ...\forall \tau_i \right]$$

With the node features and edge existence as inputs, our model first generates latent edge features by passing the edge existence alongwith features of the nodes it connects through a two layer MLP. The resulting edge features are then passed into aggregation layers to collect information from the neighboring edges, and the time context. The first aggregation step operates on the input graph topology. For every edge e_i, the neighboring edges in the input graph are divided into four categories: edges that share the same origin node, that share the same destination node, that originate from the destination of edge e_i, and that end at the origin of edge e_i. Input features from edges belonging to each of these four categories are aggregated by summation, resulting in four vectors, which are concatenated along with the feature of edge e_i and the time context, to produce the output feature for edge e_i. In this manner, the aggregation layer generates features for every edge containing information about itself, its neighbors and time. These features are passed through an MLP to generate attention weights for every edge. In the second round, aggregation is done in a similar fashion, but on a fully connected graph topology, with the neighboring edge features being weighed by the attention weights before being summed. The output of this aggregation is passed through an MLP to predict the final likelihood of each edge's existence.

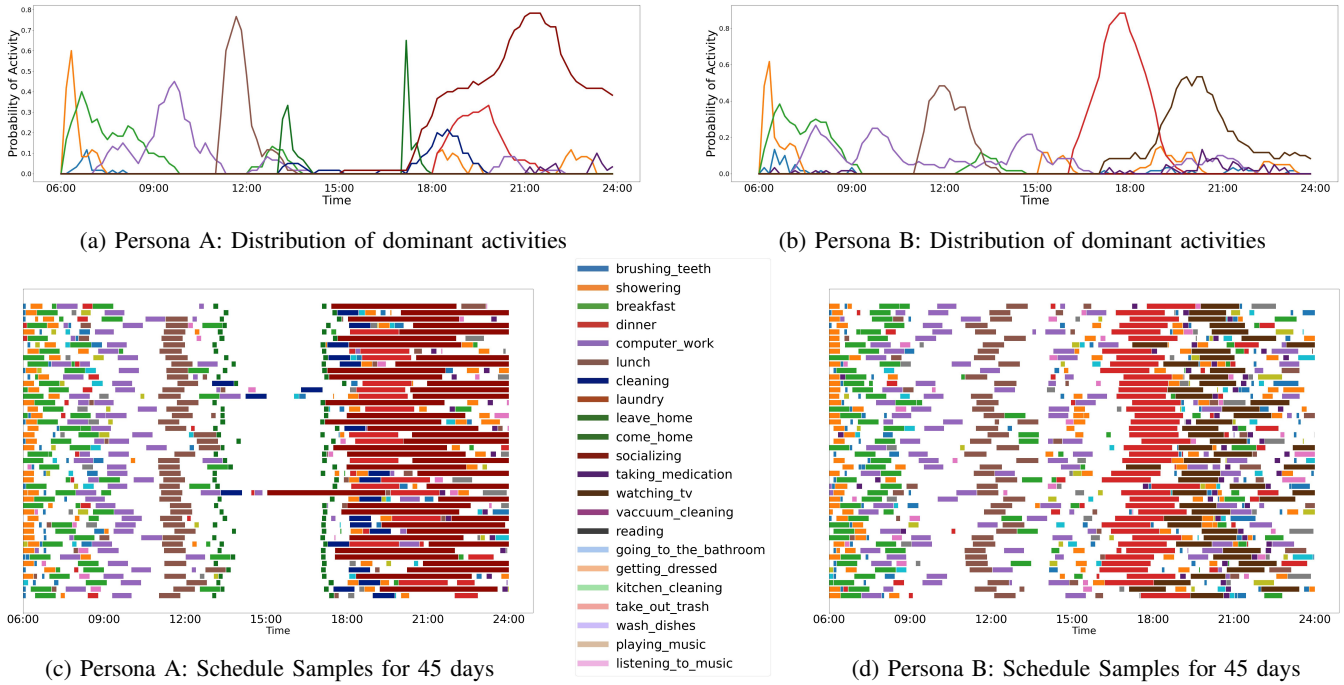
The model in NEC-DGT [27] operates directly on a fully connected graph topology, without any weights, to propagate information by edge-to-edge aggregation. This causes several incoming messages to be summed on each edge, thus diluting each neighbor's contribution. We found this to be sub-optimal for scaling to the number of objects in a house, so we employed the above attention mechanism. By predicting the attention weights based on the sparse input graph topology, we allow the signal from those edges to be stronger. To avoid limiting the model's receptive field to existing neighbors, we allow all edges to contribute in the second step of aggregation, expecting the learned weights to emphasize important neighbors.

V. Behavioral Simulation Dataset

To validate our approach to object relocation, we introduce a novel dataset which captures a diverse set of everyday household activities that humans perform as they go about their daily routines, including information on how objects move throughout a household as a result of those activities. We used the VirtualHome simulator to collect the data [13], as it supports human agents, object interaction, and high-level semantic commands without the need to control low-level motions. To compile the dataset, we first obtained a list of activities of daily living relevant to in-home routines from the activity recognition literature [37]. Specifically, we used the following list:

bathe or shower	brush teeth
clean	clean kitchen
come home	computer work
connect with friends	do laundry
get dressed	leave home
listen to music	play music
prepare and eat breakfast	prepare and eat dinner
prepare and eat lunch	read
take medication	take out trash
use restroom	vacuum clean
wash dishes	watch TV

We then sourced our dataset using a two-tier strategy, separately sourcing high level activity schedules comprising of the above activities, and the low level action sequences to perform each activity.



(a) Persona A: Distribution of dominant activities

(b) Persona B: Distribution of dominant activities

(c) Persona A: Schedule Samples for 45 days

(d) Persona B: Schedule Samples for 45 days

Fig. 4: Activity distributions and schedule samples of two personas are compared here. Notice how common activities like showering, breakfast, dinner are present at roughly the same time for both, expressing common activities that everyone would do. But there are activities specific to that persona, for instance Persona A does cleaning and prefers to socialize in the evening, and Persona B doesn't leave the house, takes medicines regularly and prefers watching TV in the evenings.

A. Activity Schedules

To obtain realistic activity schedules, we surveyed various workers on Amazon Mechanical Turk. First, we asked each worker about the most likely activities they would be doing during each hour of the day from 6am to 12am, from which we obtain a set of activities for every hour of the day for that individual. We obtain probabilities for activities in each hour by normalizing with the maximum number of activities in an hour, and assign the remaining probability to an 'idle' activity. In this manner, we get a probabilistic temporal model of activities that the individual does through the day.

We induce more variability by generating personas from this data. We find semantically meaningful traits for every activity, like having early v.s. late dinner, or brushing twice v.s. once a day, and, by combining data from participants who follow that habit, we generate a temporal distribution for that activity characterizing the specific habit. We then compose four diverse fictitious personas as a combination of such habits for every activity. These persona distributions lead to more stochastic schedule samples, and exhibit both: common activities that they all do, as well as specific activities that are unique to each persona. These similarities and differences can be observed in the final datasets generated for these personas, as shown in Figure 4, where the activity distributions, as well as schedule samples are compared for two of the personas.

B. Action Sequences

We use the VirtualHome simulator to collect scripts for the implementation of each activity from the above list. We recruited 23 participants to compose step-by-step action sequences, defining movement of the avatar and the avatar's interactions with various objects, to recreate each activity in simulation. For example, for the breakfast activity a participant might utilize actions like *walk to cabinet*, *grab cereal*, *put cereal on counter*, and so on. Ensuring that the action sequences start from a fixed home state of the environment allows us to later compose them into complete routines. Along with the actions, we also source the estimated duration ranges needed to do those actions. The outcome is a set of action sequences, along with time duration ranges needed to perform each action, for all activities.

C. Schedule Sampling

From the obtained temporal activity distributions and action sequences, we use Monte Carlo sampling to generate complete daily routines executable on VirtualHome simulator. We set the start time for our daily routine as 6am. Starting at that time, we first sample an activity from the schedule distribution. We then choose an action sequence for that activity, and sample durations for each of those actions. From the action durations, we calculate the end time of that activity, and sample the next activity at that end time. By iteratively sampling the activity and action sequences in such a manner, we compose complete daily schedules, examples of which are shown in Figure 4. These complete

routines, composed of action sequences, when executed on the simulator, provide a sequence of object arrangements in the environment, from which we derive our states X_t.

VI. Evaluation

We test our predictive model on all four of our persona datasets, and compare against the following baselines on metrics measuring the ability to correctly predict changes. We use 20 hidden layers with ReLU activation to construct each of the our MLPs. We use Adam optimizer with a learning rate of 10^{-3}. Starting with a known graph G_t at time t, we use our network to predict the probabilistic graph one step into the future $p(\tilde{G}_{t+1})$. We feed the probabilistic model back into the model, and run it iteratively to predict further into the future, and finally derive our posterior estimate $\hat{G}_{t+\delta}$ for the desired future timestep.

A. Baselines

We compare the performance of our predictive model against two baselines:

- **Static Semantic** baseline, adopted from [16], employs static priors on object-object relations. This baseline calculates prior probabilities of existence of each object-object relation using the training set. Given a state at time t, the model adds noise to the belief over object locations, and then updates the resulting belief using the prior likelihood. To adapt to our topological map formulation, we use a tunable probability of change for the noise model and spread belief uniformly over all other topological locations, as opposed to the nearby areas in metric space as done in the original work.
- **FreMEn** baseline, adapted from [17], uses past experience to model the probability of existence of relationships between object pairs as periodic functions in time. We maintain beliefs over topological relations instead of the metric occupancy grid formulation in the original work. The final belief is a combination of the prior graph and the learned periodic temporal priors, with a tunable time-decaying weight as suggested in their implementation.

The priors for both these baselines are derived from the same training data that is used to train our model.

B. Metrics

For assistive applications, we primarily care about capturing where objects will move in the future, so we focus our evaluations on accurately predicting such changes. We express such changes as object relocations, $r(o_i, l_1, l_2)$, signifying the movement of object o_i from its original location l_1 to destination l_2. The set of such relocations, $\hat{R}_{t:t+1}$, predicted to happen between time t and $t+1$ can be written as

$$\hat{R}_{t:t+1} = \{r(o_i, l_1, l_2) | e_{i,l_2} \in \hat{G}_{t+1}, e_{i,l_1} \in \hat{G}_t, l_2 \neq l_1\}$$

We would however want our robot to predict farther into the future to predict changes earlier in time. Based on a proactivity parameter δ, the agent can try to predict changes δ-steps into the future to predict relocations, $\hat{R}_{t:t+\delta}$, by collecting the first predicted relocation for every object while sequentially predicting scenes from time t to $t+\delta$

$$\begin{aligned}\hat{R}_{t:t+\delta} = \hat{R}_{t:t+\delta-1} \\ \cup \{r(o_i, l_1, l_2) | r(o_i, l_1, l_2) \in \hat{R}_{t+\delta-1:t+\delta}, \\ r(o_i, l_1, l_3) \notin \hat{R}_{t:t+\delta-1}\}\end{aligned}$$

In addition, we can extract the set of objects that are relocated as a part of relocations R as

$$\mathcal{O}(R) = \{o_i | r(o_i, l_1, l_2) \in R\}$$

We evaluate the relocations predicted by our model against the actual relocations from the ground truth sequence ($R_{t:t+\delta}$). Our objective is for the robot to be proactive, and as a result we seek to make relocation predictions ahead of the actual relocation actions that would be carried out by the human. The exact order and precise timing of the robot's relocations is not critical as long as they occur before the human-generated event (e.g, the cereal or bowl could be taken out first). Hence, we measure the predictive performance of changes over the entire proactivity window, using the following metrics to capture how well the model predicts the objects that are relocated in that window and their destinations.

- **Precision** : The fraction of objects predicted to relocate in the δ-step window ($\mathcal{O}(\hat{R}_{t:t+\delta})$) that are correct.

$$P_t = \frac{|\mathcal{O}(\hat{R}_{t:t+\delta}) \cap \mathcal{O}(R_{t:t+\delta})|}{|\mathcal{O}(\hat{R}_{t:t+\delta})|}$$

- **Recall** : The fraction of objects that actually relocated in the p-step window ($\mathcal{O}(R_{t:t+p})$) that were correctly predicted

$$R_t = \frac{|\mathcal{O}(\hat{R}_{t:t+\delta}) \cap \mathcal{O}(R_{t:t+\delta})|}{|\mathcal{O}(R_{t:t+\delta})|}$$

- **Destination Accuracy** : This metric to measures how well the model predicts the destinations of the relocated objects. For this we compare the predicted set of relocations ($\hat{R}_{t:t+\delta}$) against the ground truth ($R_{t:t+\delta}$).

$$D_t = \frac{|\hat{R}_{t:t+\delta} \cap R_{t:t+\delta}|}{|R_{t:t+\delta}|}$$

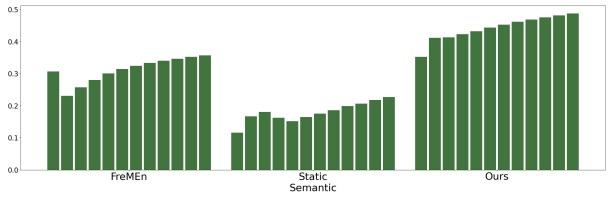
VII. Results

We compare our model against the baselines on all three metrics of precision, recall and destination accuracy defined above, as well as F-1 scores calculated using the precision and recall. We compare performance for proactivity of 10 minutes to 2 hours with increments of 10 minutes, across four persona datasets. We also show the importance of the attention mechanism in our model, and our time encoding through ablations.

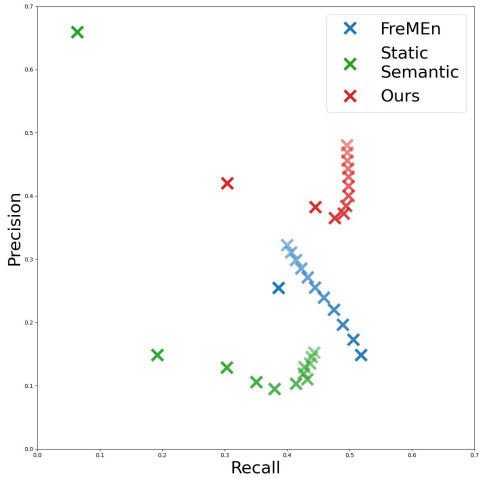
For all levels of proactivity, our method consistently predicts a larger fraction of the relocated objects and their destinations, achieving higher recall and destination accuracy, with fewer false positives, compared to the baselines.

Method	F-1 score	Precision	Recall	Destination Accuracy
Stat.Sem.	0.1695	0.3591	0.2175	0.1798
FreMEn	0.2481	0.4461	0.3098	0.3122
Ours	**0.4185**	**0.4741**	**0.3713**	**0.4423**

TABLE I: Comparison of average metrics for our method against baselines



(a) F-1 scores of our method compared against baselines on predicting objects that relocate in the proactivity window. Each column represents a proactivity window starting from 10 mins to 2 hrs in increments of 10 mins.



(b) Precision-Recall comparison of our method against baselines with lighter markers depicting longer proactivity windows. Each column represents a proactivity window starting from 10 mins to 2 hrs in increments of 10 mins. Our method beats the baselines on both precision and recall for most of the proactivity windows.

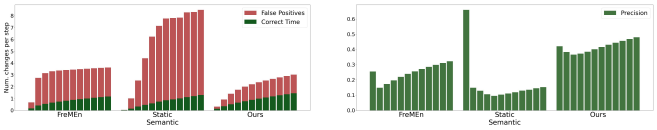
Fig. 5: F-1 score and precision-recall

Average metrics for our method compared against baselines are shown in Table I.

Our method consistently gets a better F-1 score than each baseline for all proactivity steps, as shown in Figure 5a. Our model outperforms the baselines nearly consistently on both precision as well as recall, as shown in Figure 5b. In using such predictions to provide assistance, a higher recall enables the robot to predict more movements correctly, which allows for a better level of assistance, while fewer false positives makes the user more likely to continue using the system.

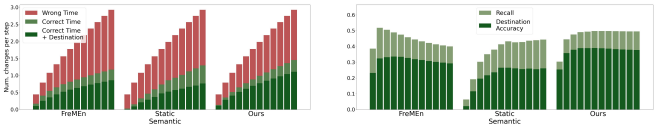
Our method is significantly superior to the baselines on the precision metric. This is because the baselines make a larger number of relocation predictions, including a large number of false positives, which outweigh the correct predictions, as can be seen in Figure 6.

Our method shows superior recall and destination accuracy, which are expanded in Figure 7. The destination accuracy metric is a stricter version of the recall metric as it measures the fraction of relocated objects as well as their destinations that are correctly predicted in the δ-step window. As we increase the proactivity steps, more flexibility is allowed in predicting the exact time of relocations, but it becomes harder to predict changes further into the future. These changes are reflected in the performance of both our method and the FreMEn baseline, where an improvement in recall and destination accuracy is seen initially as we increase proactivity, and as we look further ahead, the uncertainty in predictions causes a drop in these metrics. The Static Semantic baseline predicts a very large number of relocations in total, which could explain why it gets a larger recall, with a widening gap between recall and destination accuracy, as we increase the proactivity steps.



(a) (b)

Fig. 6: Figure (a) shows the objects correctly predicted to relocate relative to the total predictions. The ratio of these quantities is the precision, which is shown in Figure (b). Each column represents a proactivity window starting from 10 mins to 2 hrs in increments of 10 mins.



(a) (b)

Fig. 7: Figure (a) shows the correctly predicted relocated objects and their destinations relative to the total objects that were relocated. The ratio of these quantities are the recall and destination accuracy, which are shown in Figure (b). Each column represents a proactivity window starting from 10 mins to 2 hrs in increments of 10 mins.

We perform an ablation to measure the impact of our attention mechanism. We compare against a version of our model with message passing over all edges without any weighting, similar to NEC-DGT [27]. Our model achieves a significantly better precision and also better recall and destination accuracy as shown in Figure 8.

We also perform an ablation using a linear representation of time in our model instead of the periodic functions. We simply feed in a single number representing the timestamp in minutes. This causes a significant drop in performance across all three metrics as shown in Figure 9

VIII. Conclusion

In this paper, we present an object-centric perspective to providing proactive assistance in robotics. We present a method to create a behavioral dataset representing natural

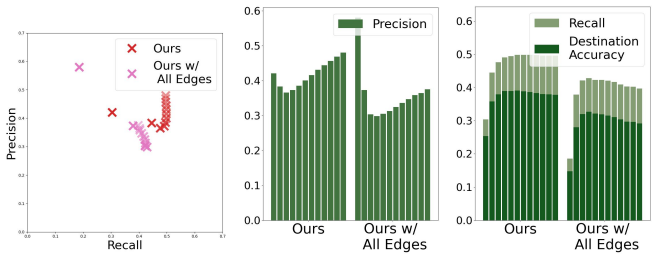


Fig. 8: Comparison of our method against a version without the attention mechanism over evaluation metrics of precision, recall and destination accuracy

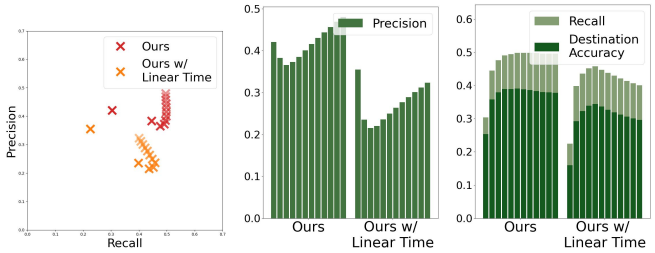


Fig. 9: Comparison of our method against a version without the periodic time encoding over evaluation metrics of precision, recall and destination accuracy

daily routines and their effect on objects, and propose a method that can leverage such data to learn a predictive model which can be used to plan assistive actions. We show that our method exhibits better predictive performance than our baselines on a variety of metrics motivated by the end goal of planning assistive actions.

Our method's ability to outperform both semantic and temporal priors lends confidence to our hypothesis that temporal patterns in our daily activities reflect in the combined spatial and temporal evolution of objects involved. The ability to pick up such patterns is useful in scenarios where rule-based goals are hard to pre-define and it is difficult for the user to command specific goals, both of which are often the case in assisting an elderly user or a user with cognitive impairments in their day-to-day life.

References

[1] G. Mois and J. M. Beer, "https://doi.org/10.1007/s13670-020-00314-wThe role of healthcare robotics in providing support to older adults: a socio-ecological perspective," *Current Geriatrics Reports*, vol. 9, no. 2, pp. 82–89, 2020.

[2] H.-M. Gross, S. Mueller, C. Schroeter, M. Volkhardt, A. Scheidig, K. Debes, K. Richter, and N. Doering, "Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5992–5999, 2015.

[3] G. Peleka, A. Kargakos, E. Skartados, I. Kostavelis, D. Giakoumis, I. Sarantopoulos, Z. Doulgeri, M. Foukarakis, M. Antona, S. Hirche, E. Ruffaldi, B. Stanczyk, A. Zompas, J. Hernandez-Farigola, N. Roberto, K. Rejdak, and D. Tzovaras, "Ramcip - a service robot for mci patients at home," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018.

[4] C. Schroeter, S. Mueller, M. Volkhardt, E. Einhorn, C. Huijnen, H. van den Heuvel, A. van Berlo, A. Bley, and H.-M. Gross, "Realization and user evaluation of a companion robot for people with mild cognitive impairments," in *2013 IEEE International Conference on Robotics and Automation*, pp. 1153–1159, 2013.

[5] X. Puig, T. Shu, S. Li, Z. Wang, Y.-H. Liao, J. B. Tenenbaum, S. Fidler, and A. Torralba, "Watch-and-help: A challenge for social perception and human-{AI} collaboration," in *International Conference on Learning Representations*, 2021.

[6] N. Oh, J. Park, J. Ho Kwak, and S. Jo, "A robot capable of proactive assistance through handovers for sequential tasks," in *2021 18th International Conference on Ubiquitous Robots (UR)*, pp. 296–301, 2021.

[7] H. Harman and P. Simoens, "Action graphs for proactive robot assistance in smart environments," *Journal of Ambient Intelligence and Smart Environments*, vol. 12, no. 2, pp. 79–99, 2020.

[8] M. Lopez-Brau, J. Kwon, and J. Jara-Ettinger, "Social inferences from physical evidence via bayesian event reconstruction," 2021.

[9] R. Shah, D. Krasheninnikov, J. Alexander, P. Abbeel, and A. Dragan, "The implicit preference information in an initial state," in *International Conference on Learning Representations*, 2019.

[10] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, 2012.

[11] R. Singh, A. Sonawane, and R. Srivastava, "Recent evolution of modern datasets for human activity recognition: A deep survey," *Multimedia Systems*, vol. 26, no. 2, pp. 83–106, 2020.

[12] D. F. Fouhey, W. Kuo, A. A. Efros, and J. Malik, "From lifestyle vlogs to everyday interactions," in *CVPR*, 2018.

[13] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "Virtualhome: Simulating household activities via programs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8494–8502, 2018.

[14] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.

[15] M. Lorbach, S. Höfer, and O. Brock, "Prior-assisted propagation of spatial information for object search," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2904–2909, IEEE, 2014.

[16] Z. Zeng, A. Röfer, and O. C. Jenkins, "Semantic linking maps for active visual object search," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1984–1990, IEEE, 2020.

[17] T. Krajnik, J. P. Fentanes, J. M. Santos, and T. Duckett, "Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments," *IEEE Transactions on Robotics*, vol. 33, pp. 964–977, aug 2017.

[18] R. Toris and S. Chernova, "Temporal persistence modeling for object search," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3215–3222, IEEE, 2017.

[19] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 5308–5317, 2016.

[20] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "Scene graphs: A survey of generations and applications," *arXiv preprint arXiv:2104.01111*, 2021.

[21] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *arXiv preprint arXiv:2002.06289*, 2020.

[22] E. E. Aksoy, A. Abramov, F. Wörgötter, and B. Dellen, "Categorizing object-action relations from semantic scene graphs," in *2010 IEEE International Conference on Robotics and Automation*, pp. 398–405, 2010.

[23] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," in *International conference on machine learning*, pp. 2323–2332, PMLR, 2018.

[24] W. Jin, K. Yang, R. Barzilay, and T. Jaakkola, "Learning multimodal graph-to-graph translation for molecule optimization," in *International Conference on Learning Representations*, 2019.

[25] D. Zhou, L. Zheng, J. Xu, and J. He, "Misc-gan: A multi-scale generative model for graphs," *Frontiers in big Data*, vol. 2, p. 3, 2019.

[26] M. Sacha, M. Błaz, P. Byrski, P. Dabrowski-Tumanski, M. Chrominski, R. Loska, P. Włodarczyk-Pruszynski, and S. Jastrzebski, "Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits," *Journal of Chemical Information and Modeling*, vol. 61, no. 7, pp. 3273–3284, 2021.

[27] X. Guo, L. Zhao, C. Nowzari, S. Rafatirad, H. Homayoun, and S. M. P. Dinakarrao, "Deep multi-attributed graph translation with node-edge

co-evolution," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 250–259, IEEE, 2019.

[28] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, and P. Riley, "Optimization of molecules via deep reinforcement learning," *Scientific reports*, vol. 9, no. 1, pp. 1–10, 2019.

[29] J. Jo, J. Baek, S. Lee, D. Kim, M. Kang, and S. J. Hwang, "Edge representation learning with hypergraphs," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[30] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, "Representation learning for dynamic graphs: A survey.," *J. Mach. Learn. Res.*, vol. 21, no. 70, pp. 1–73, 2020.

[31] Y. Ma, Z. Guo, Z. Ren, J. Tang, and D. Yin, "Streaming graph neural networks," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 719–728, 2020.

[32] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "Dysat: Deep neural representation learning on dynamic graphs via self-attention networks," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 519–527, 2020.

[33] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," *arXiv preprint arXiv:2006.10637*, 2020.

[34] V. Peñaloza, "Time2vec embedding on a seq2seq bi-directional lstm network for pedestrian trajectory prediction.," *Res. Comput. Sci.*, vol. 149, no. 11, pp. 249–260, 2020.

[35] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," *arXiv preprint arXiv:2002.07962*, 2020.

[36] H. Peng, H. Wang, B. Du, M. Z. A. Bhuiyan, H. Ma, J. Liu, L. Wang, Z. Yang, L. Du, S. Wang, *et al.*, "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting," *Information Sciences*, vol. 521, pp. 277–290, 2020.

[37] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition—a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254, 2018.