Министерство науки и высшего образования Российской Федерации
Московский физико-технический институт
(национальный исследовательский университет)
Физтех-школа прикладной математики и информатики
Магистерская программа
«Методы и технологии искусственного интеллекта»

Трунова Анастасия Михайловна

Прогнозирование движения объектов с применением графа трехмерной сцены

09.04.01 Информатика и вычислительная техника

Литературный обзор для Магистерской диссертации

Научный руководитель:
ктн.
Юдин Дмитрий Александрович

Москва
2024 г.

Содержание

1 Введение **3**

2 Постановка задачи и цели работы **3**

3 Методы прогнозирования изменения графа трёхмерной сцены **5**

3.1 3D VSG 5

3.2 MDESGM 9

3.3 STGNNOT 13

3.4 LP2 20

3.5 PROUPRA 25

4 Заключение **28**

1 Введение

Современные задачи анализа и предсказания изменений графов сцены становятся все более актуальными в связи с развитием робототехники, систем поддержки принятия решений и интеллектуальных систем взаимодействия с человеком. Графы сцены представляют собой структурированное описание окружающей среды, включающее объекты, их взаимосвязи и семантические характеристики. Решение задач долгосрочного прогнозирования изменений в таких графах требует комплексного подхода, учитывающего временные и пространственные зависимости, динамическую природу среды, а также активность пользователей.

Рассматриваемая задача включает в себя предсказание будущих состояний или перемещений объектов в динамических и сложных средах. Данная возможность крайне важна для мобильных роботов, работающих в совместных пространствах, таких как дома, офисы или общественные места, где взаимодействия с людьми и другими агентами вызывают постоянные изменения в окружающей сцене. В отличие от традиционных методов, предполагающих статичность среды, современные подходы используют трехмерные (3D) графы сцены, которые учитывают пространственные, семантические и динамические отношения между объектами. Это исследование направлено на улучшение прогнозирования перемещения объектов с использованием структуры 3D графов сцены для моделирования и прогнозирования изменений во времени.

Данный обзор посвящен анализу архитектурно различных современных методов, предложенных для решения задач предсказания изменений графов сцены.

2 Постановка задачи и цели работы

Основной мотивацией для исследования задачи прогнозировани перемещения объектов является её применение в задачах трекинга (позволяет эффективно отслеживать перемещение ключевых элементов в динамической среде, что может быть полезно для мониторинга состояния сложных систем) и в задачах построения траекторий для автомобилей или автономных агентов (здесь способность учитывать изменения в окружающей среде, такие как внезапное появление препятствий или перемещение объектов, является критически важной). То есть учет изменений среды позволяет системам предвидеть потенциальные сценарии развития событий, адапти-

руя свое поведение в реальном времени.

Рассмотрим стандартную задачу прогнозирования состояния динамического графа $G_{t+1} = (V_{t+1}, E_{t+1})$ на основе накопленной информации о предыдущих состояниях среды из графов $G_0, G_1, \ldots, G_t$ в предыдущие моменты времени.

Состояние графа в момент времени t задаётся как:

$$G_t = (V_t, E_t),$$

где:

- V_t — множество объектов в момент времени t;
- E_t — множество связей между объектами, присутствующих в момент времени t.

Обновление состояния памяти осуществляется следующим образом:

$$h_t = f_h(h_{t-1}, G_t, a_t),$$

где:

- h_t — скрытое состояние памяти в момент времени t;
- h_{t-1} — скрытое состояние памяти на предыдущем шаге;
- G_t — текущий граф;
- a_t — действие (активность) агента в момент времени t;
- f_h — функция обновления состояния памяти.

Цель состоит в прогнозировании следующего состояния графа $G_{t+1} = (V_{t+1}, E_{t+1})$ на основе текущего состояния памяти h_t:

$$P(G_{t+1} \mid h_t) = P((V_{t+1}, E_{t+1}) \mid h_t).$$

и оптимизации метода за счёт минимизации функции потерь:

$$\mathcal{L} = \mathbb{E}_t\Big[L_v(V_{t+1}, V_{t+1}^*) + L_e(E_{t+1}, E_{t+1}^*)\Big]$$

где:

- (V_{t+1}^*, E_{t+1}^*) — истинные состояния графа.
- L_v и L_e — функции потерь для узлов и рёбер.
- $\mathbb{E}_t$ — математическое ожидание по времени.

3 Методы прогнозирования изменения графа трёхмерной сцены

3.1 3D VSG

Рассматриваемый метод [1] сосредоточен на моделировании дискретных долгосрочных изменений. Для такого прогнозирования предлагается концепция Variable SGs (VSG). VSG в дополнении к традиционным SGS моделируют вероятность изменений, происходящих для отдельных объектов, что можно назвать изменчивостью. Основная цель данной работы - это построение моделей, которые оценивают семантическую изменчивость сцены.

VSG определяется как набор вершин V и набор ребер E, где каждая запись $v_i \in V$ представляет экземпляр объекта в сцене.

Семантическая информация экземпляра представлена семантическим классом объекта $o_i \in O$ и набором атрибутов $a_i \in A$. Набор возможных атрибутов A включает статические свойства (например, цвет, жесткость), которые вряд ли изменятся, динамические свойства (например, заполненый, открытый/закрытый, включенный/выключенный), которые могут изменяться в результате взаимодействия, они называются состояниями и возможные взаимодействия, которые допускает объект (например, возможность сидеть на объекте), они называются доступностью.

Семантические отношения между объектами определяются множеством R и направленным отображением ребер $\mathrm{e}(v_i\ ,\ v_j) : (\mathrm{V} \times \mathrm{V}) \rightarrow \mathrm{R}$. Такие отношения включают в себя отношения поддержки (например, положение стоя, лежа на), отношения близости (например, рядом, перед) и сравнительные отношения (например, более темный цвет, та же форма).

Поскольку сцены обычно меняются семантически непротиворечивым образом, т.е. изменяется весь объект целиком, а не отдельный фрагмент пространства, изменчивость y_i может быть интегрирована в VSG в качестве дополнительного атрибута каждой вершины v_i. Можно выделить три основных способа изменения сцены:

- Изменчивость позиции (y_P): Связана с движением объектов в пространстве. Пример: объект перемещается за определённый предел $\mathrm{r}(v_i^t, v_i^{t+1})$;
- Изменчивость состояния (y_S): Изменение состояния объектов (дверь меняет состояние с "открыто"на "закрыто") $a_i^t \neq a_i^{t+1}$;

- Изменчивость экземпляра (y_I): Топологические изменения графа. Пример: объект исчезает из сцены ($v_i^t \notin V_{t+1} \vee v_i^{t+1} \notin V_t$).

Таким образом, задача долгосрочной оценки вариативности семантической сцены была резюмирована следующим образом: оценить $y_i = \{y_P, y_S, y_I\}_i \forall n_i \in N.,,.$

DeltaVSG Заявленный авторами метод DeltaVSG базируется на марковской гипотезе, то есть предполагается, что состояние системы в следующий момент времени зависит только от её текущего состояния, а не от её истории. В контексте решаемой задачи это предположение используется для упрощения модели и фокусируется на том, что будущее состояние объектов и их связей можно предсказать, опираясь исключительно на их текущий граф сцены.

Оценка долгосрочных изменений в сценах моделируется как задача обучения с учителем, с использованием графовых нейронных сетей (GNN).

Постановка задачи Работа со сценой сводится к рассмотрению текущего G_c и будущего графа сцены G_f. Оценка изменения происходит следующим образом: все объекты $V_i^{(c)}$ в текущем графе сцены G_c сопоставляются с их аналогами $V_i^{(f)}$ в будущем графе сцены G_f Для каждого объекта $v_i^{(c)}$ вычисляется его метка изменений — ground truth variability:

$$y_i = [y_i^P, y_i^S, y_i^I] \in [0, 1]^3$$

Все метки для объектов сцены агрегируются в вектор $Y = [y_1, \ldots, y_{N_v}] \in \mathbb{R}^{N_v \times 3}$, где N_v — количество объектов в сцене.

Текущий граф сцены представляется в виде трёх матриц, пригодных для обучения нейронной сети:

- Матрица векторных представлений узлов ($M_V \in R^{N_v \times d_v}$) — представление объектов. Здесь для каждого объекта строится бинарный вектор большой размерности (619), каждое значение в векторе указывает, имеет ли узел конкретный атрибут или принадлежит ли он определённому классу. Из-за большого числа классов и атрибутов, бинарные векторы u_y становятся высокомерными и разреженными, для борьбы с этими проблемами используется метод Principal Component Analysis (PCA), он преобразует исходные бинарные векторы размерности 619 в эмбеддинги размерности 120, таким образом, что новые признаки

представляют собой линейные комбинации исходных признаков и компоненты упорядочены по степени важности.

- Матрица векторных представлений отношений ($M_R \in R^{N_e \times d_r}$) — представление связей между объектами. Для двух объектов v_i и v_j, входящих в множество узлов V, формируется вектор эмбеддинга ребра, q_{ij}, описывающий связь между ними. Этот вектор состоит из двух частей: семантические отношения (бинарный вектор, кодирующий наличие конкретного семантического отношения между объектами v_i и v_j) и геометрические отношения (вычисляется относительная позиция $I^r_{(i,j)} = I^r_j - I^r_i$), где I^r_i - положение объекта v_i в общем пространстве и $I^r_{(i,j)}$ - вектор, указывающий на расположение v_j относительно v_i. Так же учитывается пороговое расстояние (τ) для фильтрации локальных связей. Связи рёбер определяются для пар объектов, находящихся ближе, чем на расстоянии τ. После чего все векторы отношений, прошедшие порог по τ объединяются в матрицу M_R.
- Матрица индексов рёбер ($M_E \in N^{N_e \times 2}$) — показывает, каким узлам соответствуют конкретные рёбра.

Эти эмбеддинги используются для формирования обучающего примера

$$X = (M_V^{(c)}, M_E^{(c)}, M_R^{(c)}, Y).$$

Графовая нейронная сеть обучается для генерации прогнозов для объектов:

$$\hat{y}_i = \Phi(M_V^{(i)}, M_E^{(i)}, M_R^{(i)}),$$

где $\hat{y}_i$ — предсказанная метка изменений для объекта i.

В архитектуре сети используются слои Message-Passing Convolution (MP-Conv), которые рассчитывают латентные векторы признаков для объектов.

$$z_i^{l+1} = f_\theta(z_i^l) + \sum_{j \in \text{Neighbors}(i)} z_j^l \cdot h_\psi(q_{i,j}),$$

где:

- z_i^l — вектор признаков объекта i на слое l,
- f_θ и h_ψ — это feed-forward нейронные сети, которые параметризованы весами θ и ψ соответственно,
- $q_{i,j}$ — эмбеддинг отношения между объектами i и j (ребро).

Сеть обучается на реальных сценах, где для каждой пары текущего и будущего состояния объектов вычисляется вероятность различных типов изменений. В итоге модель предсказывает вероятности изменений и на их основе перестраивается входной граф сцены.

Для обучения и теста используются датасеты 3DSSG и 3RScan. Временной интервал изменений варьируется от нескольких часов до нескольких дней.

Метрики оценки включают точность (Accuracy) и F1-Score для категорий: изменчивость состояния, позиции и экземпляров объектов.

Преимущества

- Простой подход, не требующий сложных вычислений;
- Учёт разных типов изменчивости объектов;
- Модель правильно предсказывает наиболее вероятные изменения, например, перемещения часто используемых объектов, что отражает реалистичное понимание взаимодействия людей с окружающей средой;
- Компактные эмбеддинги.

Недостатки

- Недостаток контекста для предсказаний (без учета истории изменений модель теряет информацию о ранее наблюдавшихся динамических закономерностях, что делает предсказания менее точными);
- Учет дальних объектов может снижать производительность модели из-за менее значимой или шумной информации, а игнорирование этих объектов не позволяет учитывать глобальный контекст достаточно эффективно, поэтому предоставляется не полная информация о сцене;
- Отсутствие учёта человеческой активности;
- Конкретные типы изменений объектов не предсказываются моделью напрямую. Модель оценивает, какие объекты могут измениться в сцене и вероятности трёх типов изменений, исходя из их семантических и геометрических характеристик, а также контекста окружающих объектов. Однако она не пытается точно предсказать, что именно произойдет с каждым объектом.

3.2 MDESGM

В статье [2] рассматривается предсказание связей в динамических, частично наблюдаемых графах. Здесь решается стандартная задача предсказания временных связей с некоторыми дополнениями:

- Изменяющееся множество узлов. Множество узлов может изменяться между временными шагами. То есть $V_t \neq V_{t+1}$.
- Частичное наблюдение. Вместо того чтобы иметь полный доступ к состояниям графов в прошлом $G_0, \ldots, G_t$, агент имеет доступ только к частичным наблюдениям $O_0, \ldots, O_t$. Каждое наблюдение O_t включает подмножество узлов $V_t^O \subseteq V$ и рёбер $E_t^O \subseteq E_t$. Используется часть графа, от которой зависит запрос.
- Подмножество будущих рёбер. Вместо того чтобы предсказывать все рёбра в будущем графе E_{t+1}, авторы фокусируются на предсказании только подмножества рёбер $E_t^Q \subseteq E_{t+1}$. Это подмножество связано с конкретным набором узлов запроса V_t^Q, которые имеют значение для дальнейшей задачи. Эти рёбра и узлы запроса могут или не могут быть замечены в прошлом, а также могут изменяться между временными шагами.

Авторы применяют свой подход для рассуждений о графах сцен в домашних условиях, где узлы организованы иерархически. На вершине иерархии находятся узлы комнат, затем узлы мебели и далее узлы объектов. Рёбра графа представляют отношения между этими узлами. Динамическая природа среды означает, что объекты перемещаются во времени, и рёбра между узлами мебели и объектами изменяются соответственно. Задача агента состоит в том, чтобы предсказать, какой узел мебели связан с целевым узлом объекта. Агент может выбирать, какие узлы мебели наблюдать, но его наблюдения зашумлены. Некоторые объекты могут быть не замечены, даже если они связаны с выбранным узлом мебели, что имитирует реалистичное восприятие, которое может не заметить объекты из-за закрытия или других факторов. В такой постановке, задача адаптирована к поиску конкретных целевых объектов в среде.

Чтобы изучить и решить проблему прогнозирования динамических связей с частичной наблюдаемостью, данная работа включает в себя несколько компонентов:

1. **Dynamic House Simulator**

   Симулятор моделирует домашние среды с разнообразными расположениями

объектов и динамическими изменениями, включая перемещения объектов, вызванные взаимодействием с другими агентами. Моделируются состояния среды в виде графов сцены с временной динамикой.

2. **Scene Graph Memory (SGM)**

   Представляет собой структуру данных, которая агрегирует наблюдения агента и используется для накопления информации о наблюдаемых сценах. В момент времени t состояние памяти SGM (SGM_t) состоит из узлов и связей, которые могут быть: наблюдаемые ($V_t^{\text{SGM}} = \bigcup_{n=0}^{t} V_n^O$ — все узлы, которые агент видел до момента t. $E_t^{\text{SGM}} = \bigcup_{n=0}^{t} E_n^O$ — все наблюдаемые рёбра), запросные узлы (V_t^Q — узлы, о которых агент должен сделать предсказание), гипотетические рёбра E_t^H (генерируются для узлов запроса с помощью функции f_h, если для этих узлов мало наблюдаемых рёбер).

   Каждому узлу и связи приписываются семантические признаки, отражающие свойства объекта или отношений, и временные признаки (время последнего наблюдения, частота изменения состояния узла/связи, общее число наблюдений). Используются 96-мерные векторы из Tok2Vec, обученные на spaCy, для кодирования узлов и рёбер. Они гененрируются на основе текстовых данных о свойствах объектов или связей. Эти векторы помогают идентифицировать семантическое сходство между объектами или предсказать вероятные связи между узлами.

   То есть в SGM содержатся все узлы и рёбра, которые агент когда-либо наблюдал до текущего момента времени t. Однако для узлов и рёбер учитываются дополнительные временные признаки, которые позволяют различать старые и новые наблюдения.

3. **Node Edge Predictor (NEP)**

   Нейронная сеть, разработанная для предсказания присутствия рёбер запроса (query edges) в графе сцены.

   - Входные данные: Node Features (характеристики каждого узла графа, семантические и временные), Edges (все рёбера в графе), Edge Features (семантические и контекстные признаки рёбер), Query Edges (рёбра, для которых нужно предсказать вероятности наличия), Query Edge Features (признаки запросных рёбер).

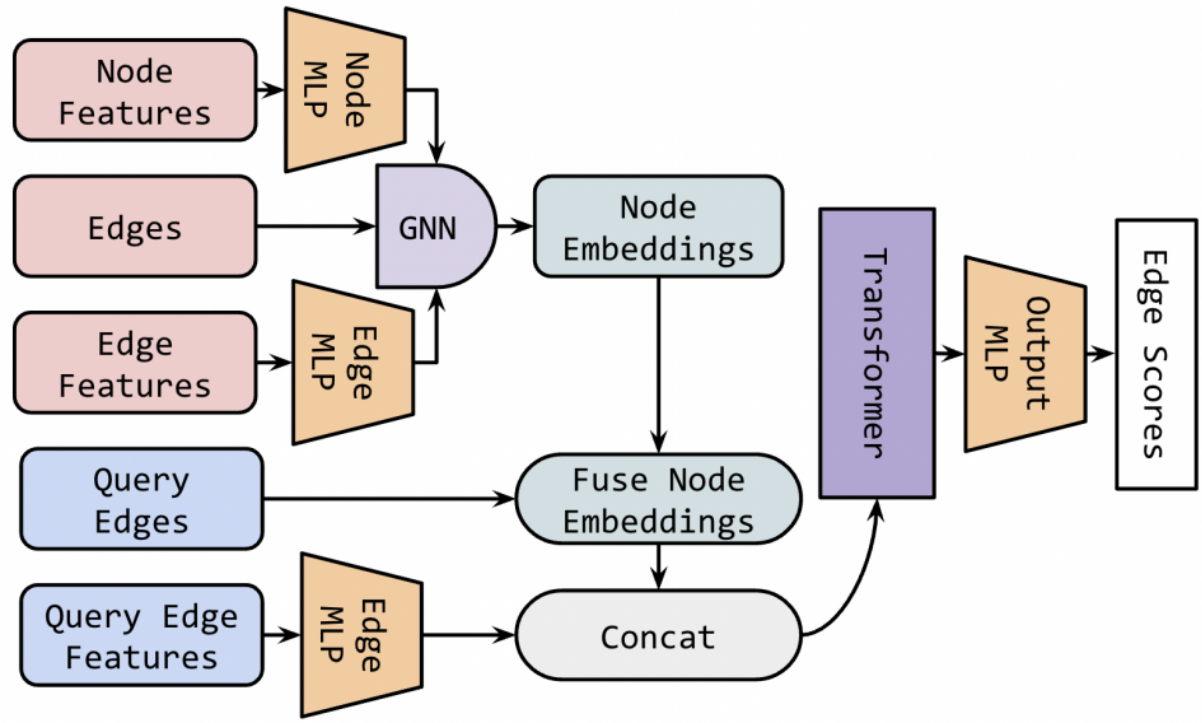

Рис. 1: Архитектура NEP

- Создание эмбеддингов: Node MLP и Edge MLP преобразуют признаки узлов и рёбер в эмбеддинги фиксированной размерности, GNN применяется для улучшения эмбеддингов узлов. Она использует информацию о связях (edges) для распространения признаков между узлами графа, улучшая их представление.
- Объединение признаков для рёбер-запросов: Fuse Node Embeddings (признаки узлов, связанных рёбрами-запросами, объединяются с использованием усреднения или других методов агрегации), Concat (полученные признаки объединяются с признаками рёбер-запросов из Query Edge Features, чтобы создать объединённый набор признаков для каждого рёбра-запроса).
- Обработка объединённых признаков: Transformer (двухслойный трансформер позволяет модели учитывать взаимные зависимости между рёбрами-запросами, что особенно важно, если рёбра взаимно исключают друг друга), Output MLP (выход трансформера передаётся в трёхслойный MLP с активацией сигмоид, который преобразует данные в вероятности для каждого рёбра)
- Выходные данные: Edge Scores (вероятности существования рёбер-запросов в графе).

Основные метрики: Accuracy: для задач предсказания местоположения и вероятностей, количество действий (Actions) и длина пути (Path Length): для оценки эффективности поиска объектов, успешность (Success Rate): для оценки доли успешных попыток.

Преимущества

- Учитывается история прошлых появлений объектов и связей между ними (Представление сцены в виде Scene Graph Memory (SGM), которое агрегирует все наблюдаемые узлы и рёбра до текущего момента, позволяет эффективно учитывать исторические изменения сцены.);
- Эффективное представление векторов объектов на основе Tok2Vec;
- Генерация "гипотетических рёбер"при отсутствии информации о связях запрашиваемых узлов;

- Предсказание изменения графа сцены для любого количества запрашиваемых рёбер.

Недостатки

- Отсутствие учёта человеческой активности;
- Накопление избыточной информации в SGM;
- Вычислительная сложность NEP (GNN и трансформер являются тяжеловесными компонентами, особенно при большом количестве запросных рёбер и узлов.);

3.3 STGNNOT

В статье [3] авторы сосредоточились на проблеме упреждающей помощи роботов в повседневной деятельности в виде размещения объектов. В качестве примеров можно привести робота, который приносит пользователю предметы в ожидании их необходимости, а также восстанавливает предметы после использования, реализация не подразумевает запросов. Этот метод не основывается на детальных моделях человеческой деятельности, требующих распознавания человеческой активности или прогнозирования действий человека, чтобы определить, какое действие выполнить следующим. Такая логика основывается на предположении, что держать агента в поле зрения в течение длительного времени и высчитывать его траекторию может оказаться непрактичным и крайне трудозатратным.

Работа направлена на то, чтобы понять закономерности в повседневной жизни пользователей, предвидеть их потребности и планировать вспомогательные действия. Информация о перемещениях предметов домашнего обихода, связанных с деятельностью человека, была взята из набора данных Everyday Routines (HOMER), в котором были отслежены необходимые объекты в течение более чем 50 дней для пяти имитируемых домов.

Для моделирования данных использовался симулятор VirtualHome, он поддерживает людей-агентов, взаимодействие с объектами и 33 высокоуровневые семантические команды, такие как “найти”, “ходить”, “схватить”, не требуя низкоуровневого управления движением. В процессе создания набора данных был получен список повседневных действий, относящихся к домашней рутине. Затем отдельно составлены расписания высокоуровневых занятий и последовательности действий низкого уровня для выполнения каждого вида деятельности и полученные данные использо-

brushing teeth	cleaning	watching TV	take out trash
showering	laundry	vaccuum cleaning	wash dishes
breakfast	leave home	reading	playing music
dinner	come home	going to the bathroom	listening to music
lunch	socializing	getting dressed	
computer work	taking medication	kitchen cleaning	

Рис. 2: Виды деятельности высокого уровня

вались для составления ежедневных графиков. В повседневной жизни используются 22 вида деятельности высокого уровня, представленные на рисунке **??**.

Чтобы получить реалистичный график работы, были собраны данные от 21 работника Amazon Mechanical Turk о том, в какие часы обычного дня, с 6 до 12 часов, они, скорее всего, будут заниматься каждым видом деятельности. Чтобы построить вероятностную модель, которая реалистично отражает различия в поведении человека, использовалась кластеризация для извлечения различных базовых привычек, связанных с выполнением каждого вида деятельности, и определено домохозяйство, используя комбинацию таких привычек для каждого вида деятельности. Параллельно с описанным выше, использовался симулятор VirtualHome для сбора сценариев, имитирующих каждое действие в повседневной жизни, с подробным описанием пошаговой последовательности действий, а также минимальной и максимальной продолжительности времени, необходимого для выполнения каждого действия. Используя как графики активности, так и последовательности действий вместе, был создан набор данных HOMER, отражающий, как объекты перемещаются по окружающей среде с течением времени. В нём охарактеризовано каждое из пяти зданий по временному распределению активности и сценарию действий для каждого вида деятельности и выборки их повседневного распорядка, созданные по методу Монте-Карло. При выполнении на симуляторе эти процедуры предоставляют последовательность расположения объектов, из которой извлекается состояния X_t для обучения и оценки модели.

Схема модели представлена на рисунке 3.

На вход модели подаётся ориентированный граф в виде мартиц: ***nodes*** (one-hot encodings эмбедиги объектов и местоположений), ***edges*** (матрица смежности, отображающая существование рёбер) и временная метка ***time***. Выход модели: полносвязный ориентированный граф, на рёбрах которого вероятности принадлежности объекта местоположению. Модель итеративно предсказывает изменение сцены через

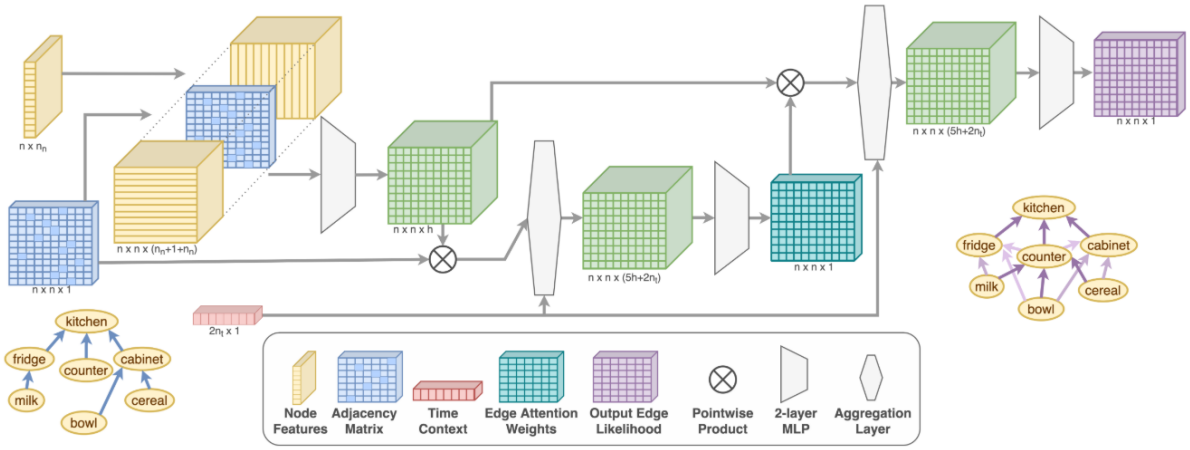

Рис. 3: Архитектура модели

заданный промежуток времени. Модель использует MLP с одним скрытым слоем, состоящим из 20 нейронов и активацией ReLU, и Adam optimizer со скоростью обучения 10-3.

Основные этапы представленного подхода:

- вычисление эмбеддингов рёбер путём конкатенации: для каждого ребра объединяются эмбеддинги узлов, которые это ребро соединяет, и бинарный признак существования ребра;
- слой MLP;
- для каждого ребра e_{ij}, модель агрегирует признаки соседних ребер, относящиеся к четырем категориям:

- ребра, исходящие из того же узла, что и текущее e_{ij} (e_{ik}),
- ребра, входящие в тот же узел, что и текущее e_{ij} (e_{lj}),
- ребра, которые исходят из узла j (e_{jm}),
- ребра, которые заканчиваются в узле i (e_{ni})).

Они объединяются с исходными признаками ребра и информацией о времени, которая кодируется на основе Time2Vec с помощью синусоиды и косинусоиды в вектор размерности $(2n \times 1)$, где n - количество предопределяемых периодов времени;

- далее предполагается, что между любыми 2 узлами существует ребро, и выполняется вычисление весов внимания для каждого ребра (MLP);
- ранее вычисленная агрегация соседей агрегируется с матрицей весов внимания;
- выходная матрица представляет **вероятности существования ребер** между узлами в графе.

В процессе инференса модель оценивается на основе последовательных данных по каждому из представленных пяти наборов данных о домах и результаты высчитываются усреднением по ним. Используется время предсказания продолжительностью 1 день, 12 часов, 6 часов, 3 часа, 1 час, 30 минут и 10 минут. Однако, фиксированный промежуток времени для предсказания - 10 минут. Начиная с известного графа G_t в момент времени t, сеть используется для прогнозирования вероятностного графа на один шаг (10 минут) в будущее $p(G_{t+1})$.

Чтобы предсказать состояние среды в будущем, на время, большее чем 10 минут вперёд, необходимо итеративно вводить выходной граф обратно в модель для дальнейшего прогнозирования. Тогда можно получить желаемую апостериорную оценку

G_{t+g}. Автоматического предсказания на время, большее, чем 10 минут вперёд, в методе не предусмотрено, поэтому возникла необходимость реализации этой задачи. Ниже изображены реальные перемещения объектов и результаты генерации в виде графа. Для более наглядной и понятной интерпретации были выведены только те узлы, которые участвовали в перемещении в один из рассмотренных моментов времени (были объектами перемещения или местами). На полученном графе узлы отмечены разным цветом в зависимости от типа, заданного в исходных данных:

- “Furniture”: ’orange’,
- “Placeable objects”: ’skyblue’,
- “Appliances”: ’green’,
- “Props”: ’gray’,
- “Rooms”: ’red’,
- “Decor”:’purple’.

Так же, серыми рёбрами отмечены наиболее вероятные связи (чем темнее цвет ребра, тем вероятнее связь), отдельно зелёным цветом выделены рёбра, верно предсказанные моделью в текущий момент времени, которые являются новой связью, то есть описывают, например, перемещения объекта на новое местоположение или в другую комнату, а красным - пропущенные (неверно не угаданные)

На рисунке 4 изображены все возможные связи между объектами и положениями в пространстве в момент времени 270 минут от начального. На каждом ребре отмечена вероятность существования связи и видно, что вероятность существования верно предсказанной связи между сумкой и комодом равна 0.6 и она наибольшая из всех существующих в выбранный момент времени.

На рисунке 5 изображены все возможные связи между объектами и местоположениями в момент времени 420 минут от начального. Здесь ни одна из вероятностей связей не превышает значения 0.25, так же, пропущена связь между стаканом и столом, которая в реальной выборке должна существовать в текущий момент времени, поэтому ребро между вершинами “drinking_glass” и “desk” было дополнительно дорисовано красным цветом, обозначающее ошибку предсказания.

Преимущества

- Метод не требует сложных моделей распознавания человеческой активности и детального прогнозирования действий, что упрощает реализацию;

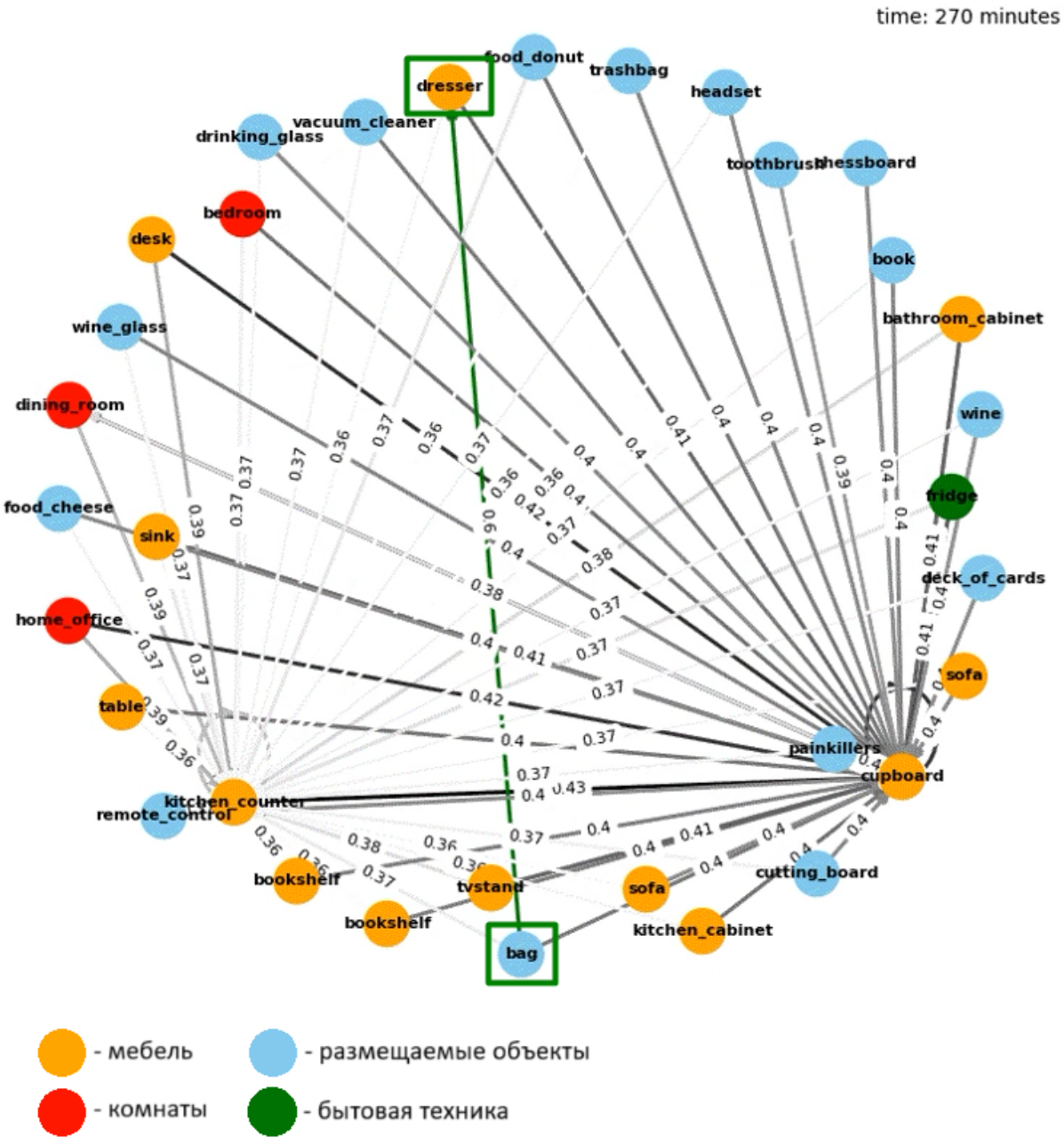

Рис. 4: Верно угаданное перемещение сумки на комод

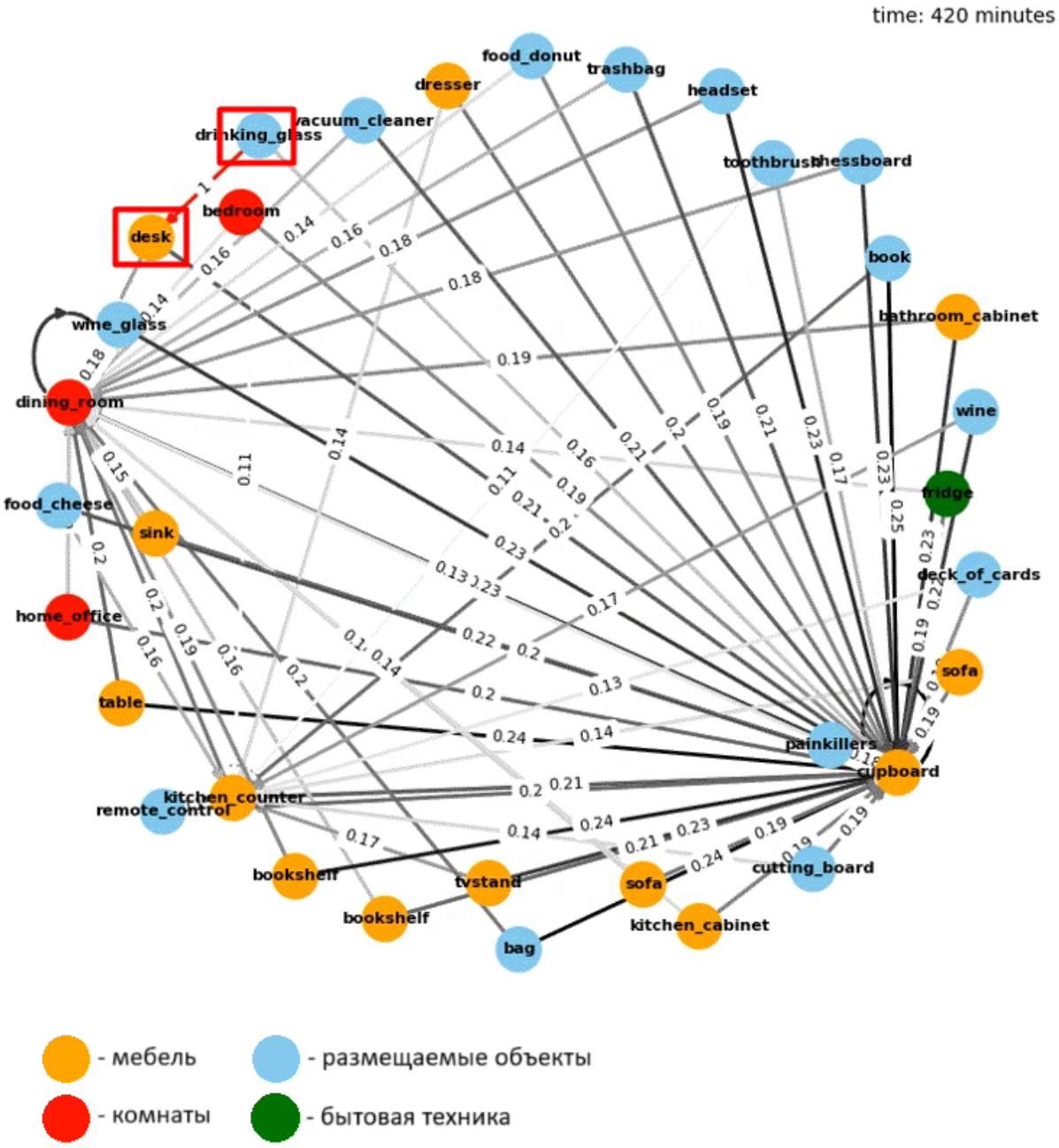

Рис. 5: Пропущенное перемещение стакана (объекта drinking_glass) на стол

- Предсказание вероятности конкретного перемешения для всех объектов;
- Простая архитектура модели.

Недостатки

- Ограничение временного прогноза (метод рассчитан на фиксированный шаг времени (10 минут), и автоматическое предсказание на большие временные интервалы отсутствует, что ограничивает его применимость.);
- Упрощённая архитектура (использование базового MLP (20 нейронов, один скрытый слой) может ограничивать способность модели к захвату сложных паттернов);
- Ресурсоемкость для долгосрочного прогноза (итеративное предсказание для дальних временных интервалов может быть вычислительно затратным, так как ошибки накапливаются с каждым шагом);
- Отсутствие автономного улучшения (модель не может корректировать свои предсказания на основе новых наблюдений в реальном времени);
- Упрощённое представление объектов в графе в виде one-hot вектора.

3.4 LP2

В статье [4] описан метод, который решает задачу долгосрочного прогнозирования траектории движения человека в семантически насыщенных средах, учитывая прошлую траекторию движения человека, его прошлые взаимодействия с окружающей средой и модель этой среды. Основная цель состоит в том, чтобы оценить мультимодальное пространственно-временное распределение $P(T_f|A_p, p, S)$ по будущим траекториям T_f с учетом прошлых взаимодействий A_p, прошлой траектории T_p и представление сцены S.

В процессе изучения метода, выяснилось, что в нём не реализовано изменение графа сцены, и в будущем времени предсказываются только возможные траектории для агента, однако построение траектории основывается на наблюдениях об активностях пользователя и рассуждении языковой модели об использовании объектов из графа сцены. Этот подход кажется перспективным, так же само представление сцены выглядит наиболее подходящим для решения задачи предсказания перемещения объектов.

Набор данных для обучения состоит из объектов и связей между ними и домами, наблюдаемых в двух типичных средах:

- Офис: 13 комнат, 107 объектов, 994 м^2, 45 траекторий;
- Дом: 9 комнат, 120 объектов, общая площадь: 111 м^2, 31 траектория.

Данные о последовательности взаимодействий были созданы людьми, привлеченными через Amazon Mechanical Turk. Аннотаторам показывались изображения сцен, и их просили представить реалистичные последовательности взаимодействий с объектами, выполняя определенные задачи. Типы аннотаций: оценка времени взаимодействия с объектами; описание действий (например, "взять чашку "переместить стул"); описание высокоуровневой цели человека (например, "подготовка к встрече"). Проверка качества: все представленные данные были проверены вручную, чтобы гарантировать их достоверность и логичность. После сбора данных человеческие модели (на основе SMPL) выполняли взаимодействия в симуляторе TESSE.

Вся сцена представлена в виде графа DSGs (3D Dynamic Scene Graph), который включает:

- Узлы объектов (V_o)
- Узлы комнат (V_r)
- Узлы пространственных точек (V_p): точки, описывающие проходимость.

Формально он состоит из набора узлов и ребер $S = \{V, E\}$. Узлы $v \in V$ сгруппированы в слои L, с меж- и внутрислойными ребрами $e_{ij} \in E$, определяющими отношения между узлами v_i и v_j. Слои $L = o, p, r$, кодируют объекты o, места p и комнаты r соответственно.

Изначально граф сцены представляет собой файл формата json, в котором определены характеристики графа, поля *edges* и *nodes*: каждое ребро представлено полями, содержащими информацию (*type*, *weight*, *weighted*), а также *source* и *target* идентификаторами соответствующих вершин. Каждая вершина представлена полями, показанными на рисунке 6, существуют 3 типа вершин: PlaceNodeAttributes, ObjectNodeAttributes, RoomNodeAttributes. Исходя из предоставленной информации, о каждой вершине можно сказать про её расположение, размеры, тип, название, цвет, слой, уникальный номер.

Траектории для предсказания хранятся в наборе данных в 76 файлах типа .csv, каждый файл делится на предыдущую траекторию и траекторию для предсказания, с которой будет сравниваться выход метода, эти данные используются в виде

```
{
  "attributes": {
    "bounding_box": {
      "max": [0.0, 0.0, 0.0],
      "min": [0.0, 0.0, 0.0],
      "type": "INVALID",
      "world_P_center": [0.0, 0.0, 0.0],
      "world_R_center": {"w": 1.0, "x": 0.0, "y": 0.0, "z": 0.0}
    },
    "color": [0, 0, 0],
    "deformation_connections": [],
    "distance": 1.0,
    "is_active": false,
    "last_update_time_ns": 0,
    "mesh_vertex_labels": [],
    "name": "",
    "num_basis_points": 0,
    "pcl_mesh_connections": [],
    "position": [15.692138671875, 28.214796066284 18,
1.20587038993835 45],
    "semantic_label": 0,
    "type": "PlaceNodeAttributes",
    "voxblox_mesh_connections": []
  },
  "id": 8070450532247930544,
  "layer": 3
}
```


Рис. 6: Поля вершин графа

```
{"id": 0,
    "trajectory": [996, 1030, 1585, 1029],
    "names": ["computer", "files", "water bottle", "office chair"],
    "interaction_times": [60.0, 40.0, 3.0, 50.0],
    "actions": [
      "Using the computer to open and review the presentation slides.",
      "Organizing files on the wooden table to ensure all necessary documents are in order",
      "Pouring a glass of water.",
      "Sitting on the office chair at the wooden table with the prepared materials, ready to begin the presentation"
    ],
    "worker_ID": "A1K8GB1TL6FQUM",
    "survey_code": "GOS-89646-LHMP",
    "total_time": 180.0,
    "task": "I am doing a task that involves a sequence of interactions with the objects in the environment",
    "rejected": false,
    "rejection_reason": ""
  },
  {
    "id": 1,
    "trajectory": [890, 296, 551, 430],
    "names": ["office chair", "bookshelf", "files", "wooden chair"],
    "interaction_times": [60.0, 20.0, 40.0, 10.0],
    "actions": [
      "Sitting on the office chair and turning on the computer to review meeting notes and agenda.",
      "Browsing the bookshelf to retrieve relevant reference materials or additional documents for the meeting.",
      "gathering files for the meeting.",
      "Arranging the wooden chair at the meeting table for a collaborative discussion."
    ],
    "worker_ID": "A1K8GB1TL6FQUM",
    "survey_code": "GOS-89646-LHMP",
    "total_time": 220.0,
    "task": "I am doing a task that involves a sequence of interactions with the objects in the environment",
    "rejected": false,
    "rejection_reason": ""
  }
```


Рис. 7: Предварительно определенные действия пользователей

pandas.DataFrame, где указаны взаимодействия с одинаковым временным промежутком, используемые объекты и описание вида взаимодействия с объектами.

Предварительно определенные действия пользователей (Actions) хранятся в .json файле, здесь присутствует описание действия, объекты, с которыми взаимодействует агент, время взаимодействия с каждым из объектов, конечная цель взаимодействия и другие характеристики (рис. 7).

Рассматриваемый метод состоит из двух основных модулей: Interaction Sequence Prediction:

- Генерация текстового описания сцены: вершины V_o, V_r и их связи из DSG преобразуется в текстовый формат, получаются полные предложения описания среды, иерархически описывающие все комнаты и объекты в них).
- Запрос к LLM: на вход подаётся описание сцены, информация о текущих взаимодействиях (A_p) и траекториях (T_p). LLM предсказывает следующие возможные взаимодействия (A_f) и их вероятности $P(A_f A_p, T_p, S)$. Результатом работы LLM является JSON-структура, которая также содержит длительности t_i вза-

имодействий (если они указаны), контекстуальные объекты из сцены V_o, обоснование прогноза LLM (ключ *reasoning*).

- Построение дерева взаимодействий TI: LLM используется итеративно, на каждом шаге узлы A_f добавляются в дерево как "листья в результате дерево будет содержать возможные последовательности действий, начиная с корня (текущей позиции агента).

Probabilistic Trajectory Prediction: (основан на Continuous-Time Markov Chain)

- Дополнение TI информацией о проходимости, полученной в местах v_p. Предполагая, что люди локально эффективны в своих путях, находятся кратчайшие пути, соединяющие каждое предыдущее взаимодействие с возможными последующими.
- Продвижение человека по этому дереву с течением времени моделируется как стохастический процесс и строится цепь Маркова с непрерывным временем (CTMC). Состояния X в CTMC соответствуют узлам в TI, включая места и действия. Для вычисления вероятности нахождения человека в определённом состоянии используется матрица переходов Q, где $d(x_i, x_j)$ - это Евклидово расстояние между вершинами x_i и x_j, а v_walk - средняя скорость ходьбы человека.
- Sampling: Сначала решается уравнение Колмогорова для определения вероятности нахождения человека в каждом состоянии x(t) в момент времени t, что даёт представление о том, с какой вероятностью человек окажется в каждом узле дерева, после чего, на основе этого распределения, с помощью Sampling выбирается путь через узлы графа.

Преимущества

- Мультимодальное представление данных (метод эффективно использует мультимодальное представление данных, сочетая различные источники информации: траектории движения, взаимодействия с объектами, граф сцены, а также текстовые описания этих взаимодействий. Это позволяет лучше учитывать различные аспекты поведения агента в семантически насыщенной среде);
- Использование LLM для прогнозирования и рассуждений (применение модели LLM (Large Language Model) для предсказания действий и взаимодействий позволяет эффективно обрабатывать контекстные и сложные зависимости между объектами, действиями и пространственным расположением. LLM также дает

возможность объяснять свои прогнозы, что повышает объяснимость метода и улучшает взаимодействие с пользователем или исследователем);
- Учёт активности агента в момент наблюдения;

Недостатки

- Отсутствие изменений в графе сцены (в методе предполагается, что граф сцены остаётся фиксированным, и предсказываются только возможные траектории для агента);
- Высокие вычислительные затраты: Использование LLM и вероятностных методов может быть вычислительно затратным в случае необходимости многократных итераций и обработки больших объемов данных);
- Необходимость в предобработке и текстовой генерации (преобразование графа сцены и взаимодействий в текстовый формат для подачи в LLM требует предварительной обработки, что добавляет сложность и потенциальные ошибки в этап подготовки данных);

3.5 PROUPRA

Статья [5] описывает разработку подхода для реализации долгосрочной проактивной помощи, где робот самостоятельно предсказывает и поддерживает действия пользователя, основываясь на изучении его рутинных привычек. Предыдущий подход Spatio-Temporal Object Tracking (STOT) использовал графовые нейронные сети для изучения динамики объектов, но не включал высокоуровневую информацию о действиях пользователя. Это ограничивало способность модели предсказывать сложные или нестабильные активности. Кроме того, авторы считают, что некоторые виды поведения человека по своей сути стохастичны и им не хватает контекстуальных подсказок, которые робот мог бы использовать. Для решения таких проблем был внедрён механизм интерактивных запросов, который можно использовать для получения информации о предполагаемых действиях пользователя и использовании объектов для улучшения прогнозирования.

Рассматривается среда, состоящая из:

- Объектов $O = \{o_i\}$,
- Локаций $L = \{l_i\}$,

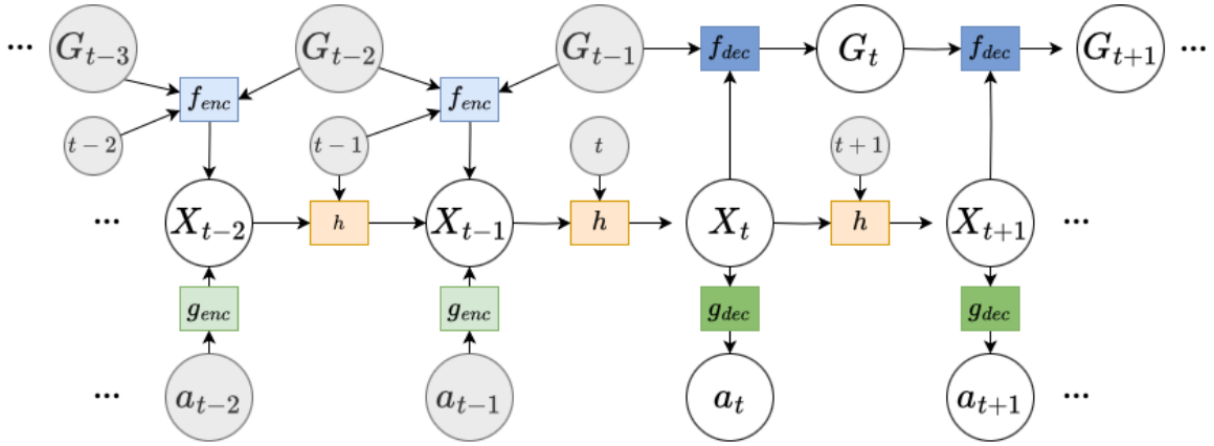

Рис. 8: Схема модели

- Человека, который выполняет действия, перемещая объекты из одной локации в другую.

Объекты могут быть локациями для других объектов. В этом случае объект находится одновременно в O и L.

Состояние среды G_t:

Это набор пар (o_i, l_j), где o_i — объект, а l_j — его текущее местоположение в момент времени t. Робот имеет полный доступ к информации о состоянии среды.

Активность пользователя a_t:

Представляет собой одну из заранее определенных активностей (A). Если активность неизвестна роботу, она обозначается как *unknown*.

Переход состояния среды $G_t \to G_{t+\Delta t}$:

Происходит из-за действий пользователя. Разницу между состояниями можно выразить через множество перемещенных объектов:

$$\Delta G_{t:t+\Delta t} = \{(o_i, l_j) \mid (o_i, l_j) \notin G_t \text{ и } (o_i, l_j) \in G_{t'}, \text{ где } t < t' < (t + \Delta t)\}.$$

Взаимодействие с пользователем: робот может задавать вопросы пользователю о его намерениях, пользователь также может вносить информацию самостоятельно (u_t) о своей активности или использовании объектов.

Графы сцены представлен узлами (объекты o_i и местоположения l_i в виде one-hot векторов), и рёбрами, соединяющими объекты с их соответствующими местоположениями. Метки активности a_t представлены также one-hot векторами, а время суток вектором синусов и косинусов предопределённых частот (t).

На рисунке 8 представлена архитектура модели, основные её части:

- Кодировщик f_{enc} фиксирует изменения окружающей среды из пар графов сцены $f_{enc} : G_{t1}, G_t, (t)ßX\hat{}t$ и получает скрытое представление общего изменения состояния окружающей среды при помощи трансформера, применяя self-attention через входные представления всех объектов, за которым следует cross-attention, обусловленное контекстом времени суток, далее max pooling.
- Декодер расположения объектов f_{dec}: $G_t, X_{t+1} \rightarrow p(G_{t+1})$, смоделированный с использованием GNN на основе edge-message-passing, генерирует вероятностный граф сцены, представляющий расположение объектов на следующем временном шаге от текущего графа сцены и с учетом скрытого вектора. Вместо кодирования и декодирования всего графа сцены подход фокусируется на фиксировании изменений в окружающей среде с помощью скрытого вектора, то есть от модели не требуется запоминания местоположений нерелевантных объектов.
- Кодирование меток действий: предусмотрен обучаемый набор векторов $g_{enc} : a_t \rightarrow X\hat{}t$ для каждой метки действий, а так же декодер меток действий в виде полностью подключенного классификатора прямой связи $g_{dec} : X_t \rightarrow p(A_t)$.

Этот подход так же построен исходя из логики метода Маркова, что позволяет упростить задачу моделирования, сводя её к анализу текущего состояния, а не всей временной последовательности.

Преимущества

- Моделирование сложных зависимостей (использование механизмов self-attention в трансформерах для кодирования информации о графах позволяет эффективно выявлять зависимости между объектами в графе, даже если они далеко друг от друга);
- Учет временного контекста (включение временного контекста через переменную (t) позволяет учитывать зависимость изменений в графе сцены от времени, что делает модель более эффективной для задач, связанных с динамическими системами, где изменения зависят от предшествующих состояний);
- Учёт активности агента в момент наблюдения;
- Латентные представления X_t накапливают информацию о предыдущих состояниях среды, которая учитывается при формировании будущих состояний.
- Предсказывание активности агента и графа сцены в заданный момент времени.

Недостатки

- Высокие вычислительные затраты;
- Проблемы с обработкой больших графов;
- Нет возможности делать предсказания с произвольным временным шагом;

4 Заключение

Анализ показал, что ни один из методов не решает поставленную задачу в полной мере. Наиболее перспективным является PROUPRA благодаря учету активности пользователя и использованию временного контекста. Однако его высокая вычислительная сложность ограничивает применение в реальных системах. Для достижения целей исследования необходимо объединить сильные стороны методов и создать интегративную модель, учитывающую временные, семантические и стохастические аспекты, а также активность пользователя.

Метод	GNN	Временная компонента	Учет активности пользователя	Запросы пользователя	Использование логики цепей Маркова	Учёт истории изменения среды	Год
3D VSG	+	+	-	-	+	-	2023
MDESGM	+	+	-	+	-	+	2023
STGNNOT	+	+	-	-	-	-	2022
PROUPRA	+	+	+	-	+	+	2024
LP2	-	-	+	+	+	-	2023

Таблица 1: Сравнительная таблица алгоритмов

Список литературы

1. 3D VSG: Long-term Semantic Scene Change Prediction through 3D Variable Scene Graphs / S. Looper, J. Rodriguez-Puigvert, R. Siegwart, C. Cadena, L. Schmid //. — IEEE International Conference on Robotics, Automation (ICRA), 2023. — DOI: 10.1109/ICRA48891.2023.10161212.

2. Modeling dynamic environments with scene graph memory / A. Kurenkov, M. Lingelbach, T. Agarwal, E. Jin, C. Li, R. Zhang, L. Fei-Fei, J. Wu, S. Savarese, R. Martın-Martın // International Conference on Machine Learning. — PMLR. 2023. — С. 17976—17993.

3. *Listl M.* Spatio-Temporal Graph Neural Networks for Multiple Object Tracking. — 2022.

4. *Gorlo N.*, *Schmid L.*, *Carlone L.* Long-Term Human Trajectory Prediction Using 3D Dynamic Scene Graphs // IEEE Robotics and Automation Letters. — 2024. — Т. 9, № 12. — С. 10978—10985. — DOI: 10.1109/LRA.2024.3482169.

5. *Patel M.*, *Prakash A.*, *Chernova S.* Predicting Routine Object Usage for Proactive Robot Assistance // arXiv preprint arXiv:2309.06252. — 2023.

6. Structural-rnn: Deep learning on spatio-temporal graphs / A. Jain, A. R. Zamir, S. Savarese, A. Saxena // Proceedings of the ieee conference on computer vision and pattern recognition. — 2016. — С. 5308—5317.

7. *Harman H.*, *Simoens P.* Action graphs for proactive robot assistance in smart environments // Journal of Ambient Intelligence and Smart Environments. — 2020. — Т. 12, № 2. — С. 79—99.