Содержание

1 Введение **3**

2 Основные определения и постановка задачи **4**

3 Обзор современного состояния проблемы **7**

3.1 Экстракторы слотовых представлений 7

3.2 Объектно-центричные модели мира 11

4 COMPAS **17**

4.1 Экстрактор объектов 17

4.2 Модель переходов 18

4.3 Объектно-центричный TD-MPC2 20

5 Эксперименты **22**

5.1 Среды 22

5.2 Модель переходов 26

5.3 Обучение с подкреплением 33

6 Заключение **36**

Аннотация

Композиционное обобщение важно для быстрого обучения и понимания новых концепций. Проблема связывания информации затрудняет обучение нейросетей на неструктурированных данных. Одним из возможных способов решения этой проблемы является использование символьных представлений. В данной работе предлагается динамическая модель мира COMPAS. Данный метод комбинирует символьные представление объектов в среде, известное как слоты или объектно-центрические состояния, с представлениями действий для точного прогнозирования следующего состояния. Другим ключевым аспектов является использованием авторегрессионной SSM (State-space model) модели, которая позволяет эффективно учитывать сложные взаимодействия между объектами и действиями в заданном контексте. Обучение экстрактора слотов в авторегрессионном режиме, позволяет достичь консистентность слотов в заданном временном окне. Эти извлеченные слоты, дополненные информацией о действиях, затем используются в модели динамики, обеспечивая точное предсказание траектории, особенно в средах с частично наблюдаемой динамикой. В секции экспериментов представляются результаты, в которых COMPAS превосходит конкурирующие объектно-центрические модели мира.

1 Введение

Модели мира служат основными компонентами в различных задачах машинного обучения, начиная от робототехники [21, 32] и автономных транспортных средств [26, 16] до искусственного интеллекта в играх и видеосинтеза [28, 4, 6, 33]. Эти модели нацелены на представление состояния и динамики окружающей среды, позволяя агентам понимать, предсказывать и взаимодействовать с окружающими объектами. Однако создание моделей мира, способных обрабатывать данные высокой размерности, непрерывные и изменяющиеся со временем, остается сложной задачей. Традиционные модели мира часто представляют окружающую среду в целом, не различая отдельные объекты. Эти модели испытывают трудности в захвате сложных взаимосвязей между объектами и их динамикой, эволюционирующей со временем. Они часто плохо обобщаются на различные задачи и сталкиваются с проблемами масштабируемости при работе со сложными сценами, содержащими множество взаимодействующих объектов. Однако модели мира подвержены проблеме связывания [9].

Способность композиционного обобщения является ключевой для более быстрого обучения решению новых проблем и пониманию новых концепций при ограниченном опыте [30]. Трудность композиционного обобщения вызвана так называемой проблемой связывания [9] – неспособность современных искусственных нейронных сетей динамически и гибко связывать информацию, распределенную по сети, которая возникает при обучении на неструктурированных входных данных.

Одним из возможных способов решения этой проблемы может быть использование символьных представлений. Такие представления могут быть описаниями объектов [12], где одно скрытое представление не кодирует входные данные, а представлено набором таких представлений (слотов). Слоты конкурируют между собой, описывая часть входных данных. Такие представления успешно применялись для задач, связанных с объектами, таких как предсказание свойств множества и обнаружение объектов в изображении [12], изучение визуальной динамики из видео [31], генерация изображений [24] и обучение представлений объектов без учителя для данных реального мира [27].

Обучение с подкреплением с использованием модели мира в робототехнике [21, 32] привлекает внимание как многообещающий подход к более эффективному обучению агентов. Путем построения модели мира, способной предсказывать будущие состояния и награды, роботы могут планировать свои действия и интеллектуально

исследовать окружающую среду. Этот метод сокращает количество попыток и ошибок, улучшая эффективность использования данных и приводя к более быстрому и экономически выгодному обучению робототехнических систем.

Несмотря на актуальность тематики объектно-центрических моделей мира, большинство работ сосредоточенно либо на моделирование сред без учета действий [10, 31], либо могут работать только в полностью наблюдаемых средах, в которых достаточно легко произвести выделение объектов [11, 13, 19]. С целью решения этих проблем, была разработана модель мира COMPAS, работоспособность которой была проверена в средах с задачами робототехнической манипуляции и в контексте обучения с подкреплением. Во 2 разделе вводятся основные определения и постановка задачи. В 3 разделе приводится описание предложенного метода и алгоритма TD-MPC для объектно-центричной. В 4 секции приводятся описание используемых сред и вычислительные эксперименты показывающие превосходство COMPAS над базовой моделью C-SWM[11] и ее модификациями, а также работоспособность модели мира с использованием TD-MPC2[32] в робототехнической среде. В заключении подводятся итоги проделанной работы и обсуждаются направления дальнейшего исследования.

2 Основные определения и постановка задачи

Определение 2.1. ***Обучение с подкреплением*** *– раздел машинного обучения который состоит в обучении* **агента** *решению определенной задачи путем получением сигнала качества ее решения –* **награды** *при взаимодействии со* **средой**.

В обучении с подкреплением вводится абстракция среды, с которой взаимодействует агент, называемая марковским процессом принятия решений.

Определение 2.2. ***Марковский процесс принятия решений (МППР)*** *– это кортеж* $\langle S, A, P, R, \gamma \rangle$*, в котором* S *– множество состояний среды,* A *– множество действий агента,* P *– условная вероятность* $P(s'|a, s)$ *перехода в новое состояние* $s' \in S$*, зависимое от состояния* $s \in S$ *и действия* $a \in A$*,* R*:* $S \times A \to \mathbb{R}$ *– функция награды,* $\gamma \in [0, 1]$ *– дисконтирующий множитель.*

Для повышения эффективности решения задач в области робототехники, агенту необходимо обладать информацией о динамике и кинематике взаимодействий внутри

среды. Однако, агент не обладает прямым доступом к таким данным при использовании только визуальных наблюдений. В связи с этим целесообразно рассматривать среду как частично наблюдаемую. Для работы с частично наблюдаемыми средами используется частично наблюдаемый марковский процесса принятия решений.

Определение 2.3. ***Частично наблюдаемый марковский процесс принятия решений (ЧНМППР)*** *– это кортеж* $\langle S, A, P, R, \Omega, O, \gamma \rangle$*, в котором* S *– множество состояний,* A *– множество действий,* P *– условная вероятность* $P(s'|a, s)$ *перехода в новое состояние* $s' \in S$*, зависимое от состояния* $s \in S$ *и действия* $a \in A$*,* R*:* $S \times A \to \mathbb{R}$ *– функция награды,* Ω *– множество наблюдений,* P_Ω *– условная вероятность* $P_\Omega(\omega|s', a)$ *получения наблюдения* $\omega \in \Omega$*, при условии достижения состояния* s' *и выполнения действия* a*,* $\gamma \in [0, 1]$ *– дисконтирующий множитель.*

Т.к. агент не имеет прямого доступа к ЧНМППР, то обычно используется убежденное состояние МППР. В качестве такого состояния предлагается использовать объектно-центрическое состояние [1].

Определение 2.4. *Пусть* $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ *– множество классов объектов,* $Att(C) = \{C.a1, \ldots, C.aa\}$ *– множество атрибутов класса* $C \in \mathcal{C}$*,* $Dom(C.a)$ *– диапазон значений атрибута* $C.a \in Att(C)$*,* $O = \{o_1, o_2, ..., o_l\}, o \in C_i$ *– множество объектов в среде. Обозначим* $s_j \in \mathbb{R}^N$ *– эмбединг, характеризующее все аттрибуты объекта* o_j*. Тогда множество* $S = \{s_1, s_2, \ldots, s_l\}$ *–* ***объектно-центричное-состояние среды*** *(в дальнейшем* ***слоты, слотовые представления****).*

В таком случае введем объектно-центричный марковский процесс принятия решений (ОЦМППР), с которым и будет взаимодействовать агент.

Определение 2.5. *Объектно-цетричный МППР – это кортеж* $\left\langle \hat{S}, A, \hat{P}, \hat{R}, \gamma \right\rangle$*, в котором* $\hat{S}$ *– множество объектно-центрических состояний,* A *– Множество действий из ЧНМППР,* $\hat{P}$ *– множество условных вероятностей* $P(\hat{s}'|a, \hat{s})$ *переходов из объектно-центрического состояния* $\hat{s} \in \hat{S}$ *в новое объектно-центричное состояние* $\hat{s}' \in \hat{S}$*, при условии совершения действия* $a \in A$*,* $\hat{R} : \hat{S} \times A \to \mathbb{R}$ *– функция награды над объектно-центрическими состояниями,* $\gamma \in [0, 1]$ *– дисконтирующий множитель из ЧНМППР.*

Для робототехнических задач, в качестве метрики качества часто используют частоту успехов:

$$\texttt{Success Rate} = \frac{e}{E} \tag{2.1}$$

где e – количество эпизодов, в которых агент выполнил задачу, E – общее количество эпизодов.

Однако, т.к. в этой метрики нет прямой явной зависимости от используемой стратегии, чаще для метрики качества самой стратегии используют среднюю отдачу. В такой постановке агент, взаимодействуя с МППР, должен максимизировать следующий функционал:

$$J(\pi) = \mathbb{E}_{\pi,P} \sum_{t=0}^{T} \gamma^t R\left(s_t, a\right) \tag{2.2}$$

Для повышения эффективности использования данных и ускорения процесса обучения в обучения с подкреплением применяются так называемые модели мира. Такой подход объединяет в себе методы обучения с подкреплением и планирования, что позволяет достичь более эффективного выполнения задач при уменьшенных затратах на взаимодействие со средой.

Определение 2.6. ***Обучение с подкреплением на основе модели*** *для ЧНМППР – раздел обучения с подкреплением, в котором агент строит модель мира, предназначенную для аппроксимации убежденного МППР. Модель мира аппроксимирует кортеж $\langle P, R\rangle$ при взаимодействии с которым, агент должен максимизировать функционал* (2.2)

Введем определение объектно-центрической модели мира:

Определение 2.7. ***Объектно-центрическая модель мира*** *– модель мира, которая аппроксимирует объектно центрический МППР. Конкретно, модель мира аппроксимирует кортеж $\left\langle \hat{S}, \hat{P}, \hat{R}\right\rangle$*

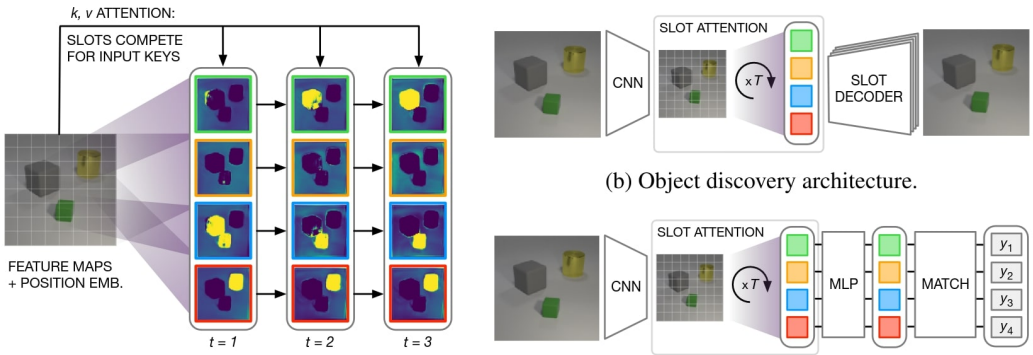
Таким образом, цель данной работы, разработать обучаемую объектно-центрическую модель мира, при взаимодействии с которой, стратегия агента π максимизирует функционал:

$$\hat{J}(\pi) = \mathbb{E}_{\pi,\hat{P}} \sum_{t=0}^{T} \gamma^t \hat{R}\left(s_t, a\right) \to \texttt{max} \tag{2.3}$$

3 Обзор современного состояния проблемы

3.1 Экстракторы слотовых представлений

Ключевой проблемой объектно-центричного обучения является эффективное формирование слотовых представлений. Качество этих представлений непосредственно влияет на эффективность решения различных прикладных задач и общее качество алгоритмов обучения с подкреплением.



(a) Slot Attention module.

(b) Object discovery architecture.

(c) Set prediction architecture.

Рис. 1: Архитектура модуля Slot Attention [12]. a) архитектура модели, b) архитектура для задачи обнаружения объектов, c) архитектура для предсказания классов объектов.

Одной из ключевых моделей для получения объектно-центричных представлений является модель Slot Attention [12]. Этот механизм работает путем итеративной группировки различных частей входных данных по их слотовым представления. Это делается с использованием специального механизма перекрестного внимания и рекуррентной нейронной сети GRU [3]. Этот процесс генерирует абстрактное представление, которое фокусируется на наборе индивидуальных репрезентаций объектов, а не на всей сцене. В качестве входного вектора используются признаки изображения, полученные на выходе энкодера CNN. Слоты для начального временного шага выбираются из обучаемого Гауссовского распределения с параметрами μ и Σ. Схема модуля изображена на Рисунке 1. Работа модуля представлена в алгоритме 3.1

Для получения представлений объектов чаще используется постановка задачи обнаружения объектов. В такой постановке, модель обучается как автоэнкодера. Пусть $h_\varphi : x \mapsto S$ – энкодер с обучаемыми параметрами φ, который преобразует изображение $x \in \mathbb{R}^{3\times H\times W}$ размера $H \times W$ в слотовые представления $S \in \mathbb{R}^{K\times N}$. Введем также декодер $p_\theta : S \mapsto \hat{x}$ с обучаемыми параметрами θ, который восстанав-

Алгоритм 3.1 Slot Attention модуль. На вход подается набор из N векторов размерности $D_{\texttt{inputs}}$ которые в ходе алгоритма назначаются на набор из K слотов размерности $D_{\texttt{slots}}$. На начальном шаге, слоты независимо друг от друга инициализируются путем сэмплирования из Гауссовского распределения с общими, обучаемыми параметрами $\mu \in \mathbb{R}^{D_{\texttt{slots}}}$ и $\sigma \in \mathbb{R}^{D_{\texttt{slots}}}$.

Вход: $\texttt{inputs} \in \mathbb{R}^{N \times D_{\texttt{inputs}}}$, $\texttt{slots}_0 \sim \mathcal{N}(\mu,\ \mathrm{diag}(\sigma)) \in \mathbb{R}^{K \times D_{\texttt{slots}}}$

Выход: $\texttt{slots} \in \mathbb{R}^{K \times D_{\texttt{slots}}}$;

$\texttt{inputs} = \texttt{LayerNorm}\,(\texttt{inputs})$

$\texttt{slots} = \texttt{slots}_0$

для $j = 0 \ldots T$

  $\texttt{slots_prev} = \texttt{slots}$

  $\texttt{slots} = \texttt{LayerNorm}\,(\texttt{slots})$

  $\texttt{attn} = \texttt{Softmax}\,(\frac{1}{\sqrt{D}} k(\texttt{inputs}) \cdot q(\texttt{slots})^T,\ \texttt{axis='slots'})$

  $\texttt{updates} = \texttt{WeightedMean}\,(\texttt{weights=attn} + \epsilon,\ \texttt{values=}v(\texttt{inputs}))$

  $\texttt{slots} = \texttt{GRU}\,(\texttt{state=slots_prev},\ \texttt{inputs=updates})$

  $\texttt{slots} \mathrel{+}= \texttt{MLP}\,(\texttt{LayerNorm}\,(\texttt{slots}))$

return slots

ливает изображение $\hat{x} \in \mathbb{R}^{3 \times H \times W}$ из слотов S. Таким образом, задача оптимизации для обучения формулируется следующим образом:

$$||x - p_\theta(h_\varphi(x))||_2 \to \min \tag{3.1}$$

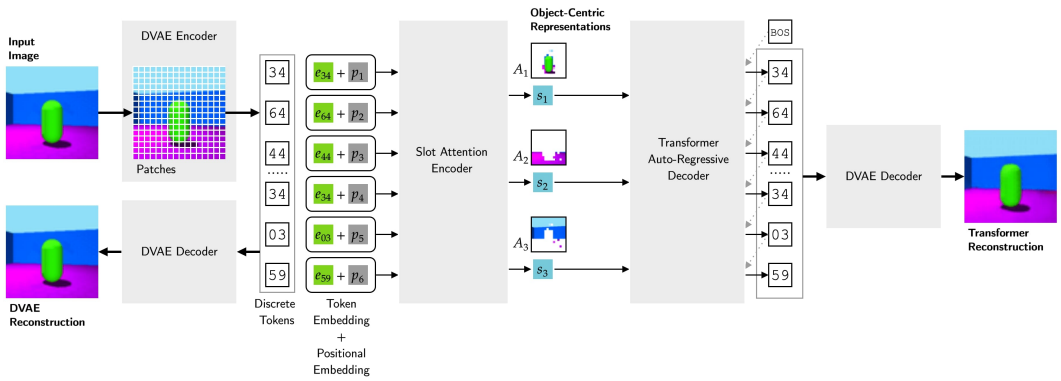


Рис. 2: Схема метода SLATE. Модель состоит из dVAE, кодирующего наблюдения в токены, slot-attention, преобразующий токены в слоты и GPT-подобного трансформера который должен из слотов восстановить токены dVAE.

Существует ряд архитектурных модификаций модуля slot-attention для задачи

обнаружения объектов. Одной из таких модификаций является модель SLATE [24]. Схема этого подхода изображена на Рисунке 2

Помимо Slot-Attention, SLATE включает в себя dVAE [7] для извлечения признаков из изображения и GPT-подобный трансформер [18] для декодирования слотов.

Цель dVAE – сократить входное изображение размером $H \times W$ до представления меньшей размерности в K раз. Сначала, наблюдение s_t подается на вход кодировщику f_φ. f_φ вычисляет логарифмические вероятности o_t для категориального распределения с C классами. Затем эти логарифмические вероятности используются для сэмплирования soft one-hot векторов j_t^{soft} из расслабленного (relaxed) категориального распределения с температурой τ. Каждый токен из j_t^{soft} представляет информацию о патче размером $K \times K$ из общего числа $P = HW/K^2$ патчей на изображении. Затем, вектор j_t^{soft} используется для восстановления наблюдения $\tilde{s}_t$ по патчам с помощью декодера g_θ.

$$\begin{cases} o_t = f_\varphi(s_t) \\ j_t^{\text{soft}} \sim \text{RelaxedCategorical}(o_t; \tau); \\ \tilde{s}_t = g_\theta(j_t^{\text{soft}}) \ . \end{cases} \tag{3.2}$$

Обучение dVAE заключается в минимизации среднеквадратичной ошибки между входными изображениями $\mathbf{s_t}$ их реконструкциями $\tilde{\mathbf{s}}_\mathbf{t}$:

$$L_{dVAE} = \text{MSE}(\mathbf{s_t}, \tilde{\mathbf{s}}_\mathbf{t}), \tag{3.3}$$

Дискретные токены j_t, полученные из категориального распределения, отображаются в эмбеддинги из обучаемых словарей. Эти эмбеддинги суммируются с обучаемым позиционным эмбеддингом p_φ для объединения информации о патчах на изображении. Затем полученные эмбеддинги $u_t^{1:P}$ подаются на вход slot-attention. Модуль slot attention возвращает N слотов $z_t^{1:N}$ вместе с их картами внимания $A_t^{1:N}$.

$$\begin{cases} o_t = f_\varphi(s_t); \\ j_t \sim \text{Categorical}(o_t); \\ u_t^{1:P} = \text{Dictionary}_\varphi(j_t) + p_\varphi; \\ z_t^{1:N}, A_t^{1:N} = \text{SlotAttention}_\varphi(u_t^{1:P}) \ . \end{cases} \tag{3.4}$$

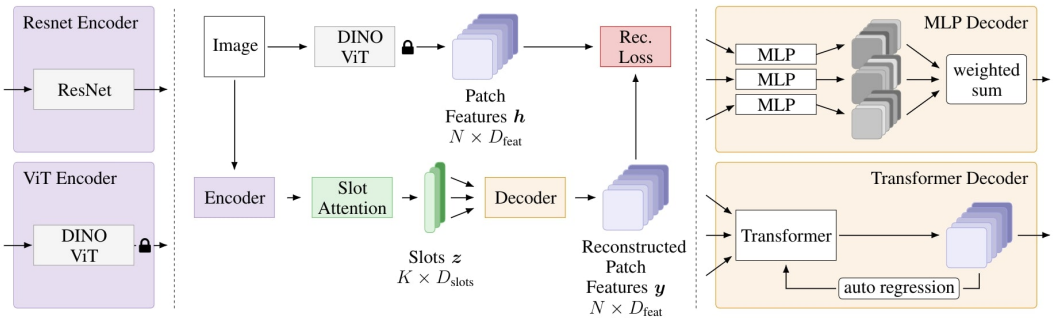


Рис. 3: Архитектура модели DINOSAur [27]. Модель поддерживает 2 возможных вида энкодеров: замороженный трансформер обученный по схеме DINO и ResNet. В качестве декодера патчей из слотов также может использоваться либо авторегресионный трансформер, либо многослойный перцептрон, который восстанавливает патчи из каждого слота независимо.

Трансформер предсказывает логарифмические вероятности авторегрессивно $\hat{o}_t^i$ для патча i из векторов $\hat{u}_t^{<i}$, сгенерированных для предыдущих патчей, совмещенных с объектно-центрическими представлениями $z_t^{1:N}$. Вектор $\hat{u}_t^l$, $l < i \in [1:P]$, формируется встроенным словарным эмбеддингом из ранее сгенерированного токена $\hat{j}_t^l$ для патча l с добавленным позиционным эмбеддингом p_φ^l. Токен $\hat{j}_t^i$ отображается на класс $c \in C$ с наивысшей логарифмической вероятностью $\hat{o}_{t,c}^i$. Полученный токен можно использовать для восстановления изображения $\hat{s}_t$ путем объединения восстановленных патчей $\hat{s}_t^i$.

$$\begin{cases} \hat{u}_t^{<i} = \text{Dictionary}_\varphi(\hat{j}_t^{<i}) + p_\varphi^i; \\ \hat{o}_t^i = \text{Transformer}_\theta(\hat{u}_t^{<i}; z_t^{1:N}); \\ \hat{j}_t^i = \arg\max_{c \in [1,C]} \hat{o}_{t,c}^i; \\ \hat{s}_t^i = g_\theta(\hat{j}_t^i)\ . \end{cases} \tag{3.5}$$

Задача оптимизации для трансформера-декодера – минимизировать кросс-энтропию между распределением токенов $\hat{j}_t$, сгенерированных трансформером, и токенами j_t, извлеченными dVAE.

$$L_{\text{T}} = \sum_{i=1}^{P} \text{CrossEntropy}(z_t^i, \hat{z}_t^i) \tag{3.6}$$

Комбинируя (3.3) и (3.6) мы получаем функцию потерь модели SLATE:

$$L_{SLATE} = L_{dVAE} + L_T \tag{3.7}$$

Ограничением метода SLATE является плохое качество выделения объектов на реалистичных изображений. Метод DINOSAur [27] пытается решить эту проблему при помощи использования предобученных по схеме DINO [15] трансформеров для изображений. Схема метода представлена на Рисунке 3.

Метод позволяет использовать любую модель в качестве бэкбона для извлечения признаков из изображения которые будут подаваться в Slot Attention. Однако, замороженный DINO трансформер имеет более высокие метрики на реальных данных.

Метод отличается от классической постановки с автоэнкодером (3.1) тем, что декодер восстанавливает патчи полученные из DINO, вместо восстановления целого изображения. В качестве декодера обычно используется многослойный перцептрон, который восстанавливает патчи с каждого слота независимо и объединяет их через взвешенное среднее. Пусть $z^{1:P}$ – признаки полученные из DINO для каждого патча, P – общее количество патчей, тогда задача оптимизации для DINOSAur формулируется следующим образом:

$$||z^{1:P} - MLP(SlotAttn(z^{1:P})||_2 \to \texttt{min} \tag{3.8}$$

Одной из важных проблем в рассмотренных ранее методах является отсутствие согласованности между слотами и объектами после обработки одного и того же изображения.

Для решения этой проблемы, slot attention обучают рекуррентным способом, подавая слоты с предыдущего шага. Для постановки классического обнаружения объектов по средствам автоенкодера (3.1) используют модели SAVI [20] и SAVI++ [23]. В постановке для SLATE (3.7) используется модель STEVE [25]. Для увеличения стабильности обучения используют так называемый предиктор для слотов с предыдущего шага перед использованием алгоритма slot attention. Схема моделей семейства SAVI изображена на Рисунке 4. Схема модели STEVE изображена на Рисунке 5.

3.2 Объектно-центричные модели мира

Объектно-центричные модели мира являются активной темой исследований. Однако большинство из них предсказывают следующме объектно-центрические состояния без учета совершенных действий, что ограничивает их применение в обучении с подкреплением.

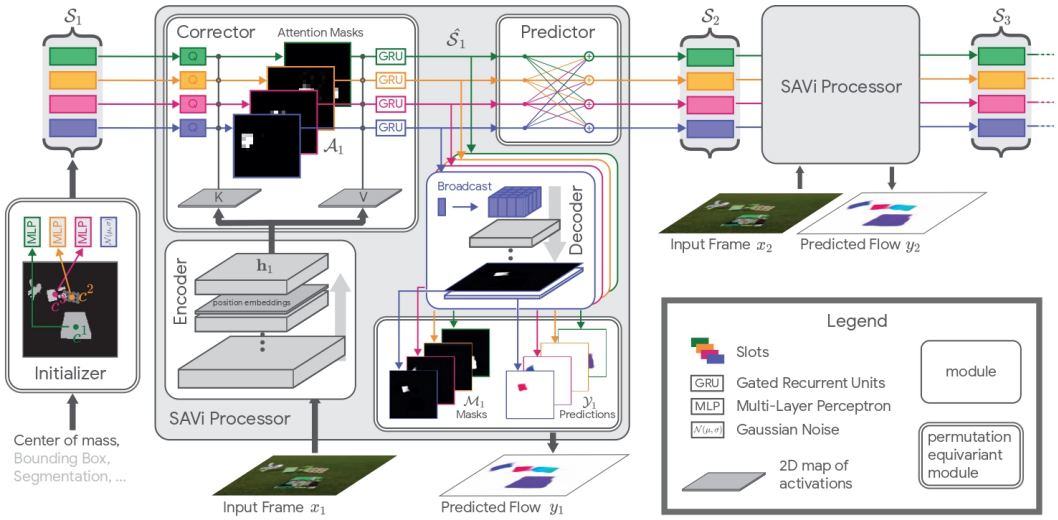


Рис. 4: Схема метода SAVI. SAVI блок представляет собой блок обычного модуля Slot Attention [12], с добавлением нового модуля - предиктора, который делает преобразования слотов, которые будут использоваться в качестве начальных на следующем временном шаге данной траектории. Также слоты на начальном шаге могут помимо Гауссовского распределения инициализироваться также из центров масс, блоков детекции, масок сегментации и т.п.

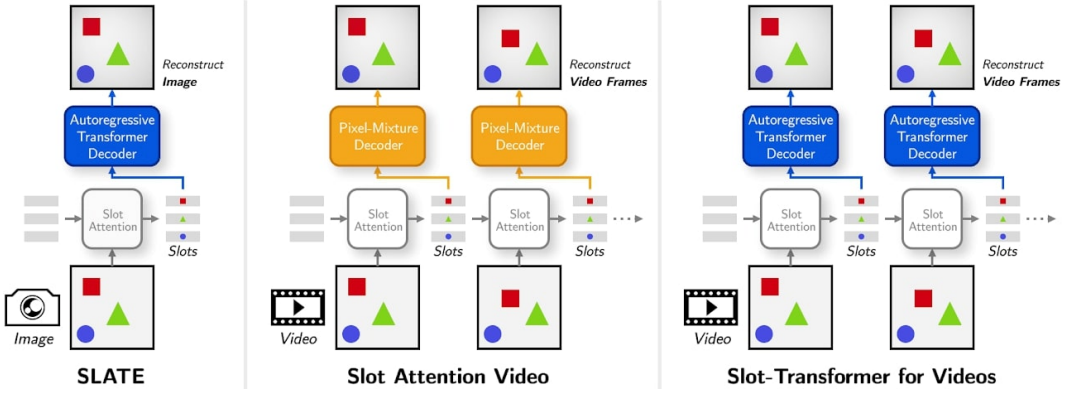


Рис. 5: Сравнение архитектуры метода STEVE с SAVI и SLATE

Одной из интересных работ является модель G-SWM [10]. Авторы работы, помимо слотов, предлагают использовать контекст-векторы, которые моделируют все элементы сцены, не являющиеся объектами. Такое нововведение добавляет механизм “понимание ситуации” объектам. Например, агент в среде PacMan должен взаимодействовать опираясь не только на информацию о противниках, но и иметь информацию о стенах, чтобы успешно передвигаться в лабиринте. Пример данной среды изображен на Рисунке **??**

Другое важное нововведение – механизм мултимодальной неопределенности. Он позволяет моделировать неопределенности в будущих предсказанных состояниях, например, развилки с 2 или 3 возможными путями в лабиринте среды PacMan.

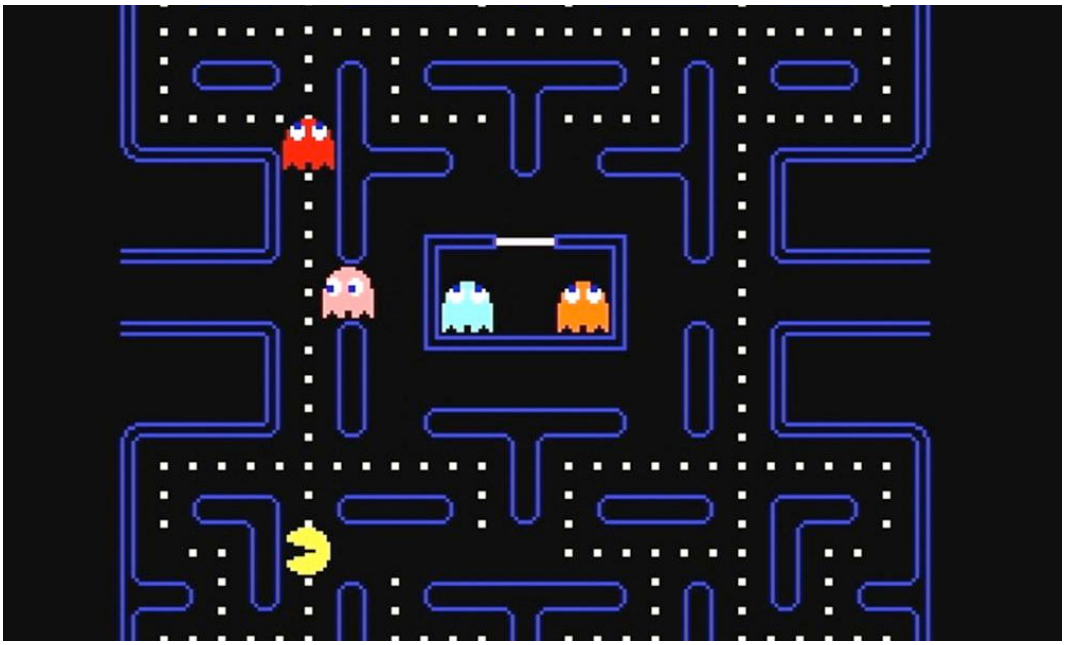


Рис. 6: Пример наблюдения в среде PacMan. Агент, управляя желтым кругом, должен собрать все белые точки в лабиринте и избежать столкновений с противниками, которые изображены в виде приведений.

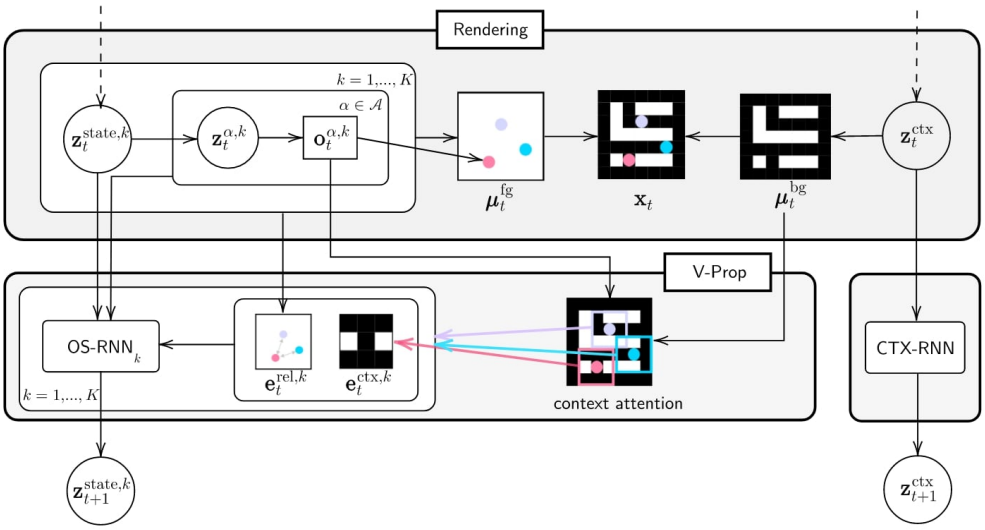


Рис. 7: Архитектура G-SWM. Модель состоит из 3 основных блоков. Блок Rendering предназначен для выделения объектов и заднего фона из изображений. В блоке V-Prop, производится моделирование мультимодальной неопределенности через механизм контекстного внимания, которое затем используется для предсказания следующего состояния объектов при помощи RNN. Для получения следующего состояния вектора контекста используется отдельный RNN

Архитектура G-SWM изображена на Рисунке 7. Для каждого шага t, сгенерированное изображение представляет собой комбинацию из изображений переднего μ^{fg} и заднего μ^{bg} планов. Передний план получается из латентного пространства обьек-

тов $z_t^{state,k}$. Задний план – из вектора-контекста z_t^{ctx}. Предсказания для латентного пространства объектов следующего шага объектов используется модуль V-prop состоящий из внимания контекста и RNN. Для предсказания контекста для следующего шага используется отдельная RNN. G-SWM, помимо 2-х мерных сред, позволяет моделировать столкновения и окклюзии в 3-х мерных средах.

Модель SlotFormer [31] позволяет авторегресивно моделировать динамику среды и предсказывать поведение объектов на произвольное количество шагов. Архитектура модели изображена на Рисунке 8. В качестве энкодера и декодера слотов используется предобученная модель, например SAVI или SLATE. На вход трансформера подаются объектно-центричные представления объектов. Затем к ним добавляются позиционное синусоидальное кодирование ко всем слотам на определенном шаге. Обработанные слоты затем подаются на вход трансфорера-энкодера [5].

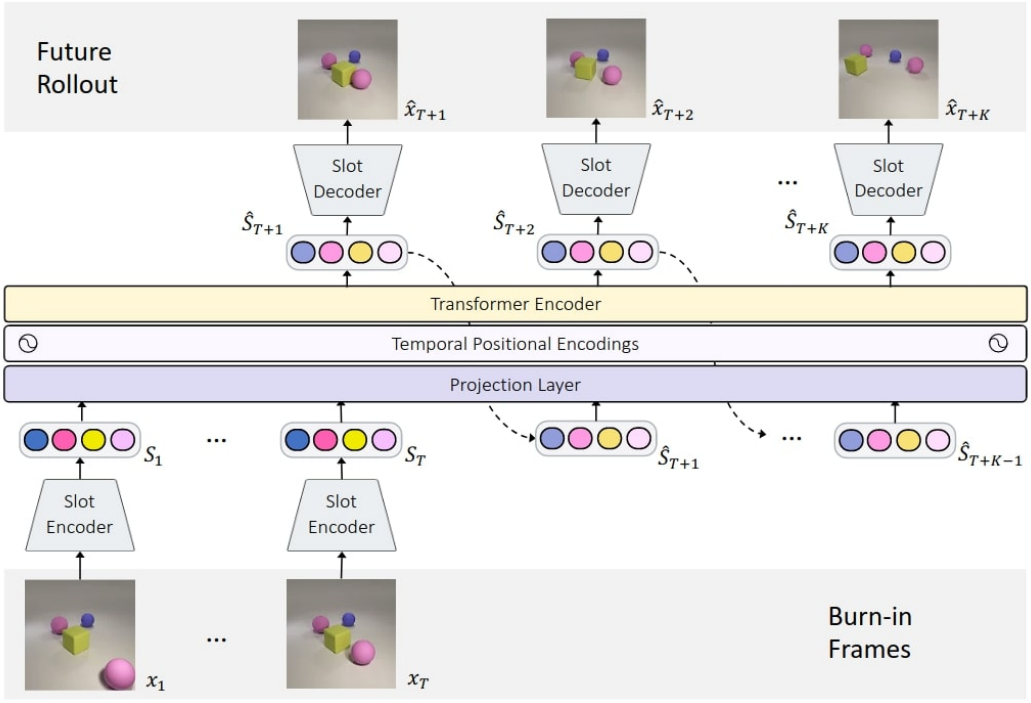


Рис. 8: Архитектура модели Slotformer. Модель представляет собой трансформер, в который подаются слоты S_t в определенном временном окне размера T. Временное позиционное кодирование позволяет трансформеру различать слоты на различных временных шагах. Предсказанием является выход трансформера, соответствующий слотам из правого конца временного окна S_T

. Модель поддерживает авторегрессию относительно переданных слотов, смещая временное окно вправо, добавляя предсказанные слоты как последний элемент и удаляя самые ранние слоты из временного окна.

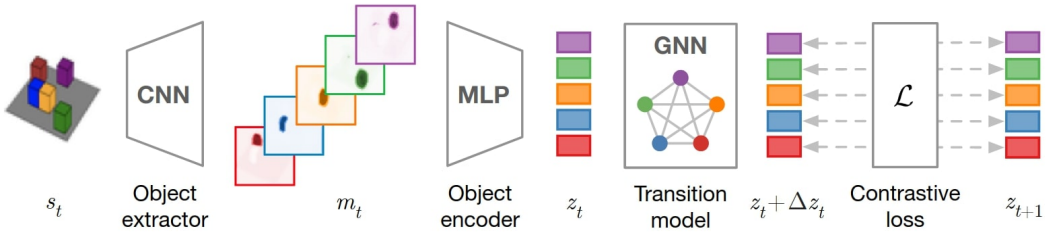


Рис. 9: Архитектура модели CSWM. Модель состоит из 3 основных блоков. Экстрактор объектов преобразует изображение в некоторое латентное пространство.Из этого латентного пространства, энкодер объектов, представляющий собой MLP, кодирует го в слоты. Дальше используется GNN которая предсказывает новые слоты

Наиболее близкой к тематике данной работы является модель C-SWM [11], которая учитывает действия при предсказаниях динамики. Модель предсказывает следующие состояние слотовых представлений в зависимости от совершенного действия при помощи графовой нейронной сети [2]. Она использует факторизированное пространство описаний объектов $\mathcal{Z} = \mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_K$ и действий $\mathcal{A} = \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K$ всех K объектов. Архитектура C-SWM, изображенная на Рисунке 9. Сначала CNN преобразует наблюдение s_t, на шаге t, в множество признаков m_t^k для k-го объекта. Каждый объект на Рисунке 9 обозначен определенным цветом. Затем, MLP преобразует признаки m_t в латентное пространство описания объектов $z_t \in \mathcal{Z}$. Далее, GNN, задающая функцию переходов $T : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$, предсказывает будущее состояние $z_t + \delta z_t$, в зависимости от совершенного действия $a_t \in \mathcal{A}$. Контрастная функция потерь в которой положительной частью является функция потерь H, а негативной $\tilde{H}$:

$$\mathcal{L} = H + \max(\gamma - \hat{H}) \tag{3.9}$$

$$H = \frac{1}{K} \sum_{k=1}^{K} d\left(z_k^t T^k\left(z_t, a_t\right), z_{t+1}^k\right)$$

$$\tilde{H} = \frac{1}{K} \sum_{k=1}^{K} d\left(\tilde{z}_{t'}^k, z_{t+1}^k\right)$$

где $\tilde{z}_{t'}$ – описание объектов, полученное из случайного наблюдения $\tilde{s}_t$ внутри эпизода, $d : S \times S \to \mathbb{R}$ – мера расстояния между слотами, в классической постановке используется евклидова мера.

Одна из более эффективных модификаций C-SWM [13] улучшает качество модели за счет более эффективного выбора негативного примера $\tilde{z}_{t'}$ в (3.9). В данной

работе предлагают выбирать негативный пример из другого эпизода и, при этом, он должен быть совмещен по шагам с позитивным $t = t'$.

Другая модификация [19] добавляет два механизма внимания для действий, который представляет собой 2 MLP с softmax операцией:

$$\alpha = \text{softmax}(k_1^T q, k_2^T q, ..., k_K^T q). \tag{3.10}$$

где j_k, j_q – MLP, для латентных состояний $\langle z_1, z_2, \ldots, z_k \rangle$ и действия a, $k = \langle j_k(z_1), j_k(z_2), ..., j_k(z_K) \rangle$, $q = j_q(a)$.

Первый предлагаемый механизм внимания называется Hard attention. В нем латентное состояние $\tilde{z}_{t+1}^k$ негативного наблюдения определяется следующим образом:

$$M \sim \text{Categorical}(\alpha_1, \alpha_2, ..., \alpha_K) \tag{3.11}$$

$$\tilde{z}_k^{t+1} = z_k^t + T_\theta(z^t, \text{pad}(a^t, m))_k \quad | \quad M = m, \tag{3.12}$$

где pad – оператор назначающий действие a^t на шаге t в m-тый слот, а во все остальные назначаются нули.

Второй – Soft Attention. В нем сначала формируется вектор $v = j_v(a)$, с которого потом формируется новое действие:

$$a' = \langle \alpha_1 v, \alpha_2 v, ..., \alpha_K v \rangle \tag{3.13}$$

Дальше это измененное действие используется вместо изначального a.

В рассмотренных методах все еще остается ряд проблем, затрудняющих их внедрение в алгоритмы обучения с подкреплением. Основной проблемой G-SWM [10] является отсутствие механизма учета действий которые совершил агент. Несмотря на то, что Slotformer [31] имеет более высокое качество предсказаний, к проблеме отсутствия действий добавляется высокое потребление вычислительных ресурсов. По сравнению со Slotformer, C-SWM хоть и учитывает действия, но модель переходов не может эффективно представлять среды с частично наблюдаемой динамикой, т.к. учитывает информацию только с предыдущего шага.

4 COMPAS

Основным результатом работы является модель мира объектно-центрическая модель мира COMPAS (**COMP**ose **A**ctions and **S**lots).- Модель состоит из 2 блоков: экстрактор объектов, который группирует наблюдения в слотовые представления и модель переходов, которая предсказывает слоты на следующем временном шаге. Архитектура модели представлена на Рисунке 10.

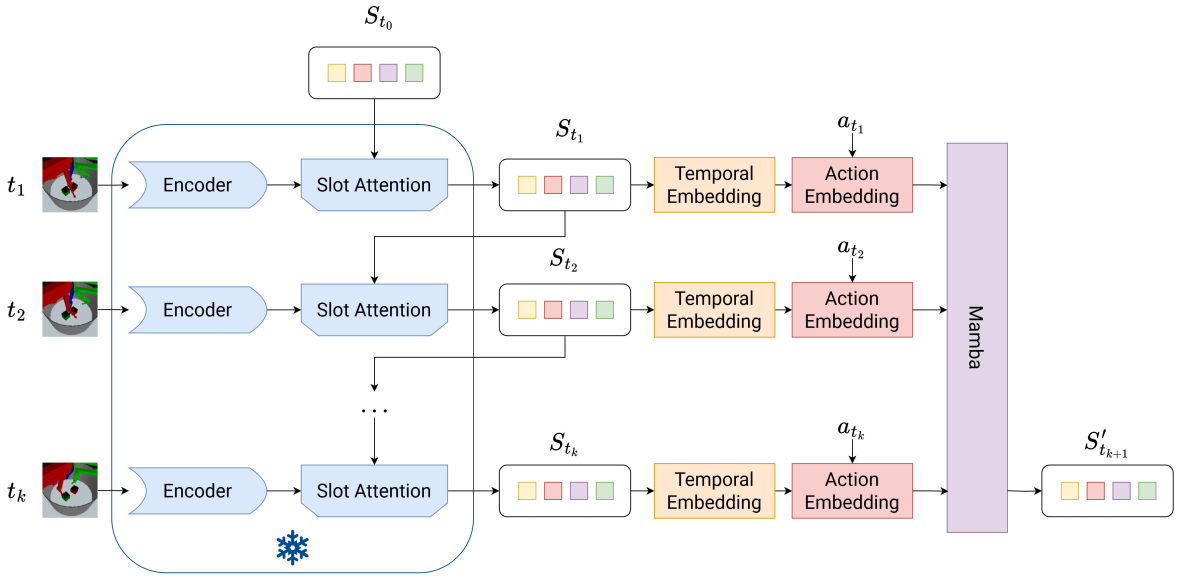


Рис. 10: Архитектура модели COMPAS. Для получения слотовых представлений, в замороженный энкодер (DINOSaur или SLATE) подается траектория наблюдений размера k. Для сохранения эпизодальной согласованности между слотами и объектами, Slot Attention обучается и используется в рекуррентным способом, подавая на обработку следующих наблюдений, слоты с предыдущего шага. На нулевом шаге слоты инициализируются из случайного Гауссовского распределения. Далее ко всем слотам на определенном временном шаге добавляется временное кодирование, чтобы встроить информацию о положении слотов по временной шкале. Затем ко всем слотам добавляется кодирование действий, которое добавляет информацию о совершенном действии. В конце, обработанные слоты передаются в модель Mamba и на выходе последний шаг из карты признаков модели переходов является следующими предсказанными слотами.

4.1 Экстрактор объектов

Модель COMPAS не зависит от экстрактора объектов. Для экспериментов с моделью переходов использовался Slot attention с рекурентным способом обучения, как в работах [23, 20, 25], без использования предиктора. Такой подход позволяет сохра-

нить консистентность слотов внутри эпизода и уменьшить затраты вычислительных ресурсов на выделение слотов и обучение экстрактора.

$$\text{slots}_0 \sim \mathcal{N}(\mu, \Sigma), \quad M = \frac{1}{\sqrt{D}} k(\mathbf{z_t}) q(\text{slots}_{t-1})^T, \tag{4.1}$$

$$\text{attn}_{i,j} = \frac{e^{M_{i,j}}}{\sum_{j=1}^{K} e^{M_{i,j}}}, \quad W_{i,j} = \frac{\text{attn}_{i,j}}{\sum_{i=1}^{N} \text{attn}_{i,j}}, \tag{4.2}$$

$$\text{updates} = W^T v(\mathbf{z_t}) \in R^{K \times D}, \tag{4.3}$$

$$\text{slots}_t = \text{GRU}(\text{input}{=}\text{updates},\ \text{hidden}{=}\text{slots}_{t-1}), \tag{4.4}$$

где $\mu,\ \Sigma$ – обучаемые параметры; K – количество слотов; D – размерность слотов; $k,\ q,\ v$ – линейные операторы, с обучаемой матрицей.

Также проводились эксперименты на более совершенном методе – Slot-mixture Module (SMM) [34]. Данный модуль представляет собой нейросетевую аппроксимацию EM-алгоритма. Псевдокод представлен в Алгоритме 4.1

Slot Attention и SMM могут использоваться в постановке DINO, так и в постановке SLATE, в зависимости от природы данных. Независимость от экстрактора объектов позволяет модели масштабироваться и для более новых и качественных экстракторов. Т.к. слоты оказывают сильное влияние на стабильность обучение всех последующих блоков (модель динамики, награды и т.д.) после предобучения, экстрактор объектов замораживается.

4.2 Модель переходов

Модель перходов была вдохновлена моделю Slotformer [31]. В основе модели перехода COMPAS лежит Mamba [29], что позволяет предсказывать более длинные траектории с уменьшенным использованием памяти. Для стабильного обучения модели переходов, экстрактор объектов замораживается.

В начале, слоты $S_t \in \mathbb{R}^{N \times D}, t = 0, \ldots, T$ во временном окне размера T проецируются в размерность K модели перехода (4.5)

$$\hat{S}_t \in \mathbb{R}^{N \times K} = Linear(S_t) \tag{4.5}$$

Алгоритм 4.1 Псевдокод Slot Mixture модуля. π инициализируется из равномерного категориального распределения, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_{diag}$ инициализируются из обучаемого гаусовского распределения. В рекуррентной постановке π, $\boldsymbol{\mu}$ и $\boldsymbol{\Sigma}_{diag}$ берутся из предыдущего шага

Вход: $\boldsymbol{x} \in \mathbb{R}^{N \times D}$ – Вход, по которому проводится группирование; $\boldsymbol{\mu}, \boldsymbol{\Sigma}_{diag} \in \mathbb{R}^{K \times D}, \pi \in \mathbb{R}^{K}$ – SMM параметры.

Выход: slots $\in \mathbb{R}^{N \times 2D}$

$\boldsymbol{x} = \text{MLP}(\text{LayerNorm}(\boldsymbol{x}))$

для $t = 0...T$

$\text{logits} = f_\theta(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}_{diag})$

$\gamma = \text{SoftMax}(\text{logits} + \log \pi, \text{dim}{=}K)$

$\pi = \gamma.\text{mean}(\text{dim}{=}N)$

$\boldsymbol{\mu}^* = \text{WeightedMean}(\text{weights}{=}\gamma, \text{values}{=}\boldsymbol{x})$

$\boldsymbol{\mu} = \text{GRU}(\text{input}{=}\boldsymbol{\mu}^*, \text{hidden}{=}\boldsymbol{\mu})$

$\boldsymbol{\mu} \mathrel{+}= \text{MLP}(\text{LayerNorm}(\boldsymbol{\mu}))$

$\boldsymbol{\Sigma}_{diag} = \text{WeightedMean}(\text{weights}{=}\gamma, \text{values}{=}(\boldsymbol{x} - \boldsymbol{\mu})^2))$

$\text{slots} = \text{concat}([\boldsymbol{\mu}, \boldsymbol{\Sigma}_{diag}], \text{dim}{=}D)$

return slots

Чтобы учесть информацию о времени последовательности, к каждому слоту $\hat{s}_t^i, i = 1, \ldots, N$ добавляется синусоидальные временное кодирование p_t.

$$\tilde{s}_t^i = \hat{s}_t^i + p_t \tag{4.6}$$

Модель переходов поддерживает дискретные и непрерывные действия. Непрерывные действия $a_t^{cont} \in \mathbb{R}^B$ приводится к той же размерности, с которой работает Mamba (4.7).

$$\hat{a}_t \in \mathbb{R}^K = Linear(a_t^{cont}) \tag{4.7}$$

В случае дискретных действий используются обучаемые эмбединги для каждого возможного действия $a_t^i \in A_t^{desc} = \left\{a_t^1, a_t^2, \ldots, a_t^J\right\}$.

$$\hat{a}_t = EmbeddingsDictionary(a_t^i) \tag{4.8}$$

Для добавления информации о действиях, действие $\hat{a}_t$ суммируется с каждым обработанным слотом $\tilde{s}_t^i$

$$\bar{s}_t^i = \tilde{s}_t^i + \hat{a}_t \tag{4.9}$$

Для осуществления прогнозирования слотов на последующий шаг S_{T+1}, используется модель Mamba. Перед этим в неё подаётся набор обработанных слотов, сформированный на основе фиксированного временного окна $\left\{\bar{S}_t\right\}_{t=1}^{T}$. Такой подход позволяет учитывать сложные взаимодействия, даже в средах, где это невозможно сделать всего за один шаг. Далее карта признаков полученная из последнего шага выхода модели Mamba, проецируется в размерность слотов. Таким образом, в результате получатся предсказания слотов на следующем шаге S'_{T+1}.

$$S'_{T+1} = Mamba(\left\{\bar{S}_t\right\}_{t=1}^{T}) \tag{4.10}$$

Изменяя временное окно, добавляя S'_{T+1} в качестве последнего элемента и получая $\hat{a}_{T+1}$, можно авторегрессивно предсказывать следующие состояния для любого горизонта L.

Функция обучения для модели COMPAS, обозначаемая L_2, определяется как квадратичная ошибка восстановления между истинными слотами $s_{t,i} \in S_t$ и предсказанными моделью $s'_{t,i} \in S'_t$

$$L_s = \frac{1}{L \cdot N} \sum_{t=T+1}^{L} \sum_{i=1}^{N} ||s'_{t,i} - s_{t,i}||^2 \tag{4.11}$$

4.3 Объектно-центричный TD-MPC2

Имплементация COMPAS для задачи обучения представляет собой расширение метода TD-MPC2 [32] для объектно центричных состояний.

В таком случае архитектурные составляющие принимают следующий вид:

$$\begin{cases} S_t = h(o_t) \\ S'_{t+1} = \texttt{COMPAS}(S_t, a_t, S_{t-1}, a_{t-1}, \ldots, S_{t-K}, a_{t-K}) \\ \hat{r}_t = R(S_t, a_t) \\ \hat{q}_t = Q(S_t, a_t) \\ \hat{a}_t = p(S_t) \end{cases} \tag{4.12}$$

где h – экстрактор слотов, S_t – слотовые представления на шаге t, o_t – наблюдение на шаге t, S'_{t+1} – предсказанное следующее состояние, COMPAS – модель переходов

COMPAS, K – количество поддерживаемой памяти в COMPAS, $\hat{r}_t$ – предсказанная награда перехода, R – MLP для предсказания награды, a_t – действие совершенное на шаге t, Q – MLP для предсказания дисконтированной отдачи по траектории. $\hat{q}_t$ – дисконтированная отдача по траектории до шага t, $\hat{a}_t$ – действие, которое максимизирует Q-значение, p – MLP для предсказания $\hat{a}_t$.

Функция обучения модели мира в объектно-центричной постановке принимает следующий вид

$$\mathcal{L} = \mathop{\mathbb{E}}_{\left(\mathbf{S}_t, \mathbf{a}_t, r_t, \mathbf{S}'_{t+1}\right)_{0:H} \sim \mathcal{B}} \left[\sum_{t=0}^{H} \lambda^t \left(\mathcal{L}_{\texttt{COMPAS}} + \mathcal{L}_{\texttt{reward}} + \mathcal{L}_{\texttt{value}} \right) \right] \tag{4.13}$$

$$\mathcal{L}_{\texttt{COMPAS}} = \| \mathbf{S}'_{t+1} - \mathrm{sg}(h(\mathbf{o}_{t+1})) \|_2^2 \tag{4.14}$$

$$\mathcal{L}_{\texttt{reward}} = \mathrm{CE}(\hat{r}_t, r_t) \tag{4.15}$$

$$\mathcal{L}_{\texttt{value}} = \mathrm{CE}(\hat{q}_t, q_t) \tag{4.16}$$

где $\texttt{sg}$ – оператор остановки градиента, $\mathcal{L}_{\texttt{COMPAS}}$ – функция потерь для модели перехода (COMPAS), $\mathcal{L}_{\texttt{reward}}$ – функция потерь для модели наград, $\mathcal{L}_{\texttt{value}}$ – функция потерь для модели, предсказывающей value-функцию.

Функция потерь для стратегии принимает вид:

$$\mathcal{L}_p = \mathop{\mathbb{E}}_{(\mathbf{s}, \mathbf{a})_{0:H} \sim \mathcal{B}} \left[\sum_{t=0}^{H} \lambda^t \left[\alpha Q(\mathbf{S}_t, p(\mathbf{S}_t)) - \beta \mathcal{H}(p(\cdot | \mathbf{S}_t)) \right] \right], \tag{4.17}$$

$$\mathbf{S}_{t+1} = \texttt{COMPAS}(\mathbf{S}_t, \mathbf{a}_t, \ldots, \mathbf{S}_{t-K}, \mathbf{a}_{t-K}),\ \mathbf{S}_0 = h(\mathbf{o}_0)\,, \tag{4.18}$$

5 Эксперименты

В данном разделе приведены эксперименты описывающие качество модели переходов COMPAS и ее применимость в контексте обучения с подкреплением. Для того чтобы проверить масштабируемость модели, была выбрана простая среда с дискретным пространством действий Atari Pong, и более реалистичные среды симулирующие физические взаимодействия между объектами с непрерывным пространством действий.

5.1 Среды

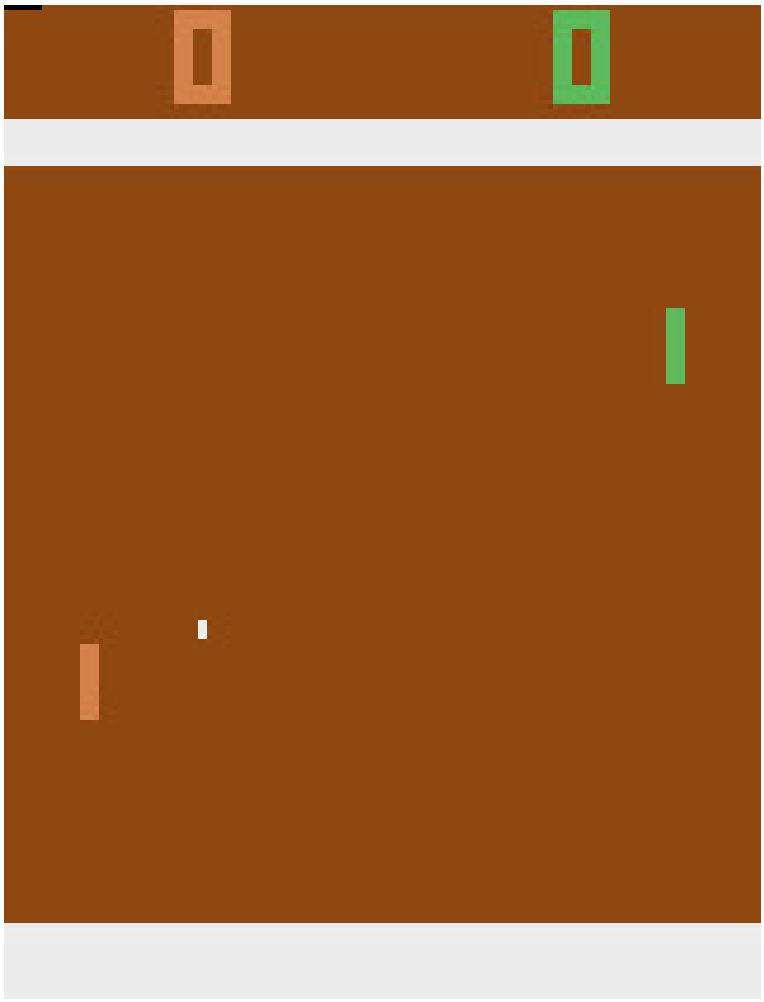


Рис. 11: Пример наблюдения в среде Atari Pong

Atari Pong представляет собой классическую среду видеоигры, предлагающую упрощенное окружение для проведения экспериментов. Эта двумерная среда обладает дискретным пространством действий. В рамках данной игры агенту необходимо

научиться управлять ракеткой с целью отражения мяча и предотвращения его достижения края экрана. Несмотря на кажущуюся простоту, Pong ставит перед агентом сложную задачу в виде прогнозирования динамики мяча и ракетки на основе текущего состояния и совершаемых действий. Пространство действий в Pong является дискретным и включает шесть возможных вариантов.

Для сбора данных была применена равномерная случайная стратегия. Набор данных был сформирован из 3000 эпизодов, каждый из которых включал 50 временных шагов, предназначенных для обучения, и еще 1000 эпизодов с аналогичной длиной траектории, предназначенных для валидации. Исходные изображения с разрешением 160x192 были обрезаны до размера 160x160, что позволило сохранить лишь основную часть игрового поля с ракетками и мячом. В процессе обучения использовались уменьшенные изображения размером 64x64. Пример наблюдения среды понг изображен на Рисунке 11

Для тестирования метода на более реалистичных наблюдений используется среда Causal World [8]. Causal World — это физически симулированная среда с тремя роботизированными манипуляторами, взаимодействующими с объектами. Она предоставляет непрерывное пространство действий и требует от агента понимания сложных физических взаимодействий и влияния действий на состояние мира.

В среде Causal World рассматриваются две задачи, обе из которых имеют непрерывное пространство действий $\mathbb{R}^9$:

- В задаче Push цель агента — толкать один блок в направлении цели с заданной ориентацией.
- В задаче Stack агенту необходимо ставить два блока друг на друга в заданном целевом положении и ориентации.

Для задач Stack и Push используется ту же схему сбора данных, что и для среды Atari Pong.

Наблюдения в Causal World формируются в виде изображений размером 128x128. Пример наблюдений из Causal World изображены на Рисунке 12

Дополнительно проводятся исследования в более сложной роботической среде RoboSuite [22], где используется роботическая рука Franka Panda для манипуляции. Для экспериментов используется подъема блока, в которой роботическая рука должна поднять красный куб, расположенный на столе перед ней, выше определенной

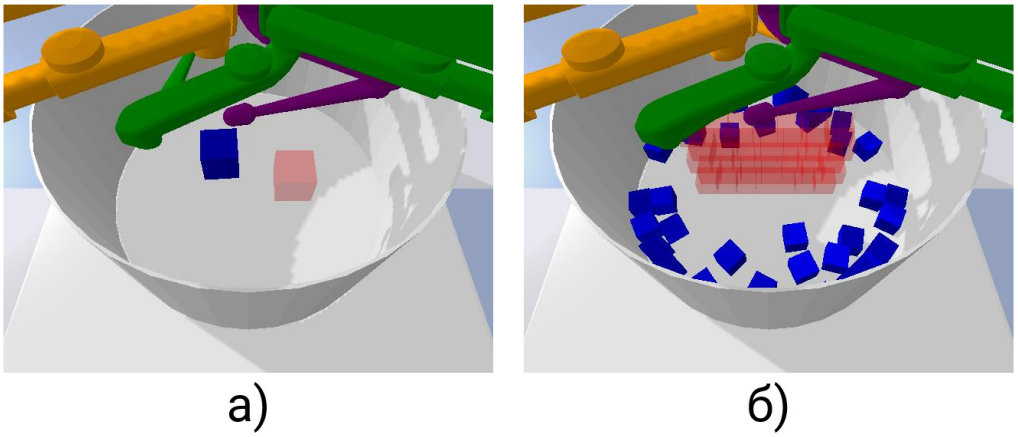


Рис. 12: Визуализация задач Causal World. а) задача Pushing, б) задача Stacking,

высоты. Местоположение куба и начальное состояние робота случайным образом назначаются в начале каждого и каждого эпизода. Размер непрерывного пространства действий для этой задачи – $\mathbb{R}^4$. Поскольку агент, управляемый равномерно распределенной случайной стратегией, не способен схватить куб, для более качественного обнаружения объектов, сначала должен обучится агент при помощи обучения с подкреплением DreamerV3 [33]. После предобучения, агентом собираются 6000 траекторий, содержащих 50 состояний перехода. При сборе данных отбрасываются траектории, в которых куб удерживается роботической рукой более 25 тактов времени. Этот процесс выбора позволяет сбалансировать распределение состояний, когда роботическая рука движется с кубом и без него. В 70% траекторий робот успешно поднимает и перемещает куб. В оставшихся 30% траекторий рука взаимодействует с кубом, но не может его схватить. Таким образом, этот набор данных содержит смесь экспертных и субоптимальных траекторий и напоминает наборы данных из обычных оффлайн-бенчмарков [14].

Набор данных был разделен на обучающий набор, содержащий 5000 траекторий, и проверочный набор, содержащий 1000 траекторий. В среде RoboSuite формируются изображения размером 256x256 с фронтальной камеры. Эти изображения затем обрезаются, в результате чего получается разрешение 224x224.

Тестирование модели мира в задаче с подкреплением производилась в среде Metaworld[17]. Среда представляет собой набор различных робототехнических задач. Для экспериментов используется друг похожий на Robosuite на задаче Push. Основная цель этой задачи передвинуть манипулятором кубик в выбранную случайным образом позицию на столе, отмеченную зеленым шаром. Наблюдения представляют

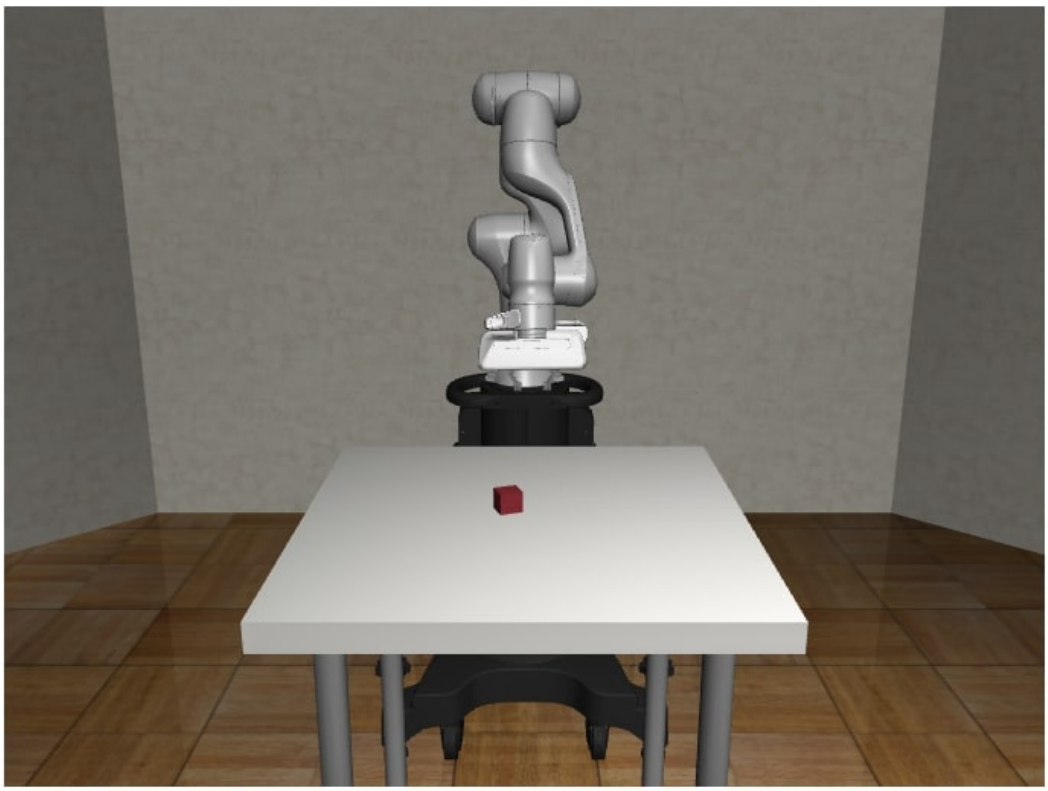


Рис. 13: Пример наблюдения в среде robosuite на задаче Lift

собой реалистичные изображения, которые формируются симулятором с разрешением 224x224. Пример случайного наблюдения из этой среды изображен на Рисунке 14

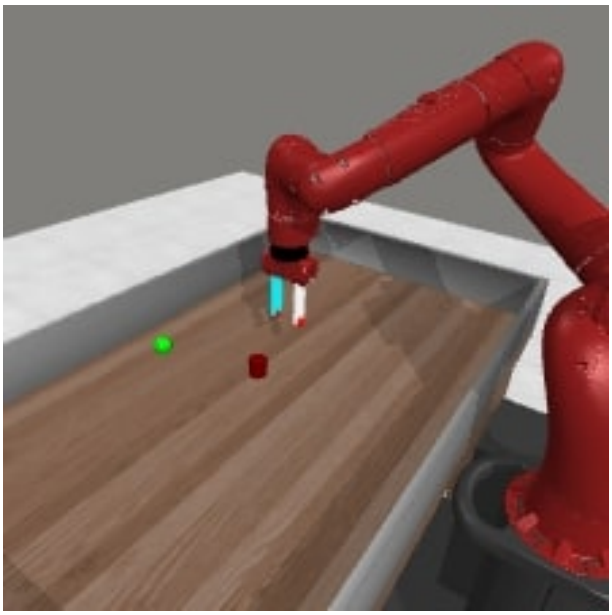


Рис. 14: Пример наблюдения в среде MetaWorld на задаче Pushing

Таблица 1: Параметры dVAE CNN энкодера.

Слой	Количество каналов	Активация	Гипперпараметры
Conv2D 4×4	64	ReLU	stride: 4
Conv2D 1×1	64	ReLU	stride: 1
Conv2D 1×1	64	ReLU	stride: 1
Conv2D 1×1	64	ReLU	stride: 1
Conv2D 1×1	64	ReLU	stride: 1
Conv2D 1×1	vocab size	ReLU	stride: 1
Position Embedding	-	-	absolute

5.2 Модель переходов

Вследствие того, что на момент написания работы у экстракторов объектов есть проблема используемой модальности, заключающаяся в том, что часть моделей работают качественно в синтетических средах – SLATE [24], STEVE [25], SAVI [20], но плохо в реалистичных. И обратная проблема у модели DINOSAur [27], которая наоборот может делать относительно качественные репрезентации объектов в реалистичных средах, и плохие в синтетических.

С учетом этих ограничений, в качестве экстрактора объектов для среды понг используется модель STEVE без предиктора слотов на следующий шаг. параметры для модели приведены в таблицах 1, 2 и 3

Для реалистичных сред Robosuite, MetaWorld и Causal World использовался рекуррентный DINOSAur. Параметры обучения изображены в таблице 4.

Для стабильного обучения модели переходов, экстрактор объектов замораживается. Гипперпараметры COMPAS представлены в таблице 5

Следуя методике оценки, описанной в [19], предсказываются будущие слоты на 1, 5 и 10 шагов вперёд и сравниваются с реальными представлениями слотов на соответствующих временных шагах. Метрика HITS@1 вычисляет долю случаев, когда предсказанные слоты оказались ближе к реальным слотам, чем к любым другим предсказаниям:

$$\texttt{HITS@1} = \frac{|U|}{|D|}, \tag{5.1}$$

где D представляет собой оценочный набор данных, а $U \subseteq D$ – множество ближай-

Таблица 2: Параметры dVAE CNN декодера.

Слой	Количество каналов	Активация	Гиперпараметры
Conv2D 3×3	64	ReLU	stride: 1
Conv2D 1×1	64	ReLU	stride: 1
Conv2D 1×1	64	ReLU	stride: 1
Conv2D 1×1	64 * 2 * 2	ReLU	stride: 1
PixelShuffle	upscale factor = 2	-	-
Conv2D 3×3	64	ReLU	stride: 1
Conv2D 1×1	64	ReLU	stride: 1
Conv2D 1×1	64	ReLU	stride: 1
Conv2D 1×1	64 * 2 * 2	ReLU	stride: 1
PixelShuffle	upscale factor = 2	-	-
Conv2D 1×1	3	-	stride: 1

ших истинных образцов к предсказанным.

Метрика среднего обратного ранга (Mean Reciprocal Rank, MRR) вычисляется следующим образом:

$$\texttt{MRR} = \frac{1}{|D|} \sum_{n=1}^{|D|} \frac{1}{\text{rank}_n}, \tag{5.2}$$

rank_n обозначает позицию реальных экземпляров в списке расстояний, который включает как реальные, так и предсказанные примеры, отсортированные по расстоянию между предсказанными и реальными латентными состояниями, от наименьшего к наибольшему значению.

Поскольку slot attention имеет тенденцию к перестановке слотов на разных временных шагах, для сопоставления предсказанных и реальных слотов при вычислении матрицы расстояний использовался Венгерский алгоритм.

Чем выше значения HITS@1 и MRR, тем лучше модель предсказывает будущие состояния.

Как показано в таблицах 6, 7, 8 и 9, COMPAS продемонстрировал более высокую точность предсказаний по сравнению с C-SWM в задаче Push and Stack. Этот результат свидетельствует о надежности данного подхода в работе со сложными физически моделируемыми средами. Визуализация карт внимания слотов для задач

Таблица 3: Гиперпараметры используемые при обучении на среде Pong.

Модуль	Параметр	Значение
	Размер изображения	64x64
	Количество собранных эпизодов	3000
	Количество шагов на эпизод	50
	Количество эпох обучения	150
	Размер батча	128
dVAE	Размер словаря	128
dVAE	Диапазон затухания температуры	1.0 to 0.1
dVAE	Кол. шагов затухания температуры	30000
dVAE	Скорость обучения	0.0003
Transformer	Кол. слоев	6
Transformer	Кол. голов	4
Transformer	Разм. трансформера	128
Slot Attention	Количество илотов	4
Slot Attention	Количество итераций	3
Slot Attention	Разм. слотов	64
Slot Attention	Кол. рекуррентных проходов	5

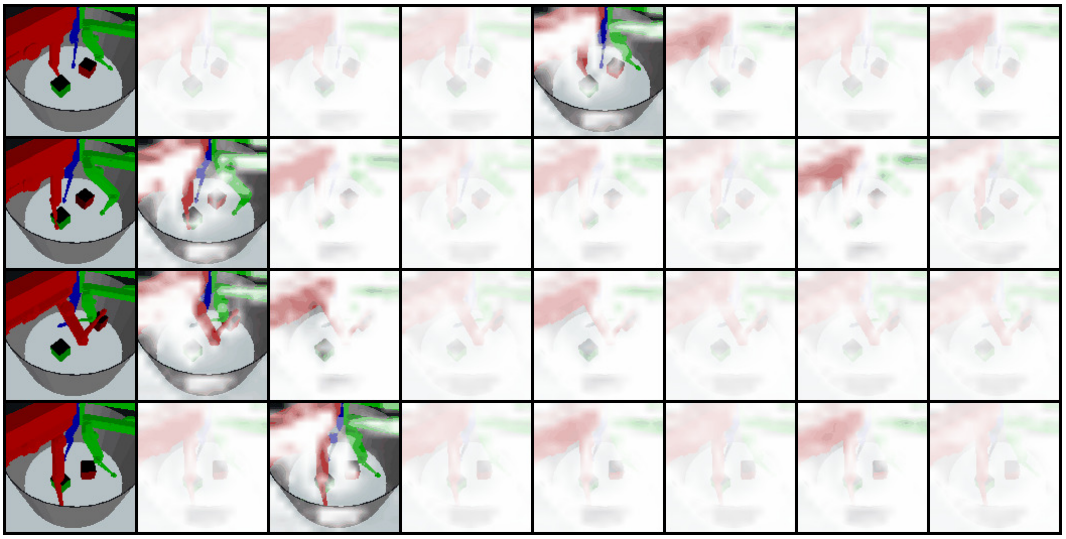


Рис. 15: Визуализация карт внимания слотов для среды Causal World (push). Первый столбец – исходное изображение, все остальные – визуализации масок внимания для каждого слота

Push и Stack представлены на Рисунках 15 и 16

Тем не менее модели C-SWM демонстрируют значительно лучшие результаты

Таблица 4: Гипперпараметры для DINO, используемые в средах casual world and robosuite.

Module	Parameter	Value
	Размер изображения (Casual World/Robosuite и MetaWolrd)	128/224
	Количество собранных эпизодов	3000
	Количество шагов на эпизод	50
	Количество эпох обучения	150
	Размер батча	128
DINO	Размер патча	8
DINO	Размер модели	S
MLP Decoder	Кол. слоев	3
MLP Decoder	Разм. MLP	2048
Slot Attention	Кол. слотов (MetaWorld/Casual World/Robosuite)	8/7/4
Slot Attention	Начальное кол. итераций.	3
Slot Attention	Посл. рекуррентное кол. итераций	1
Slot Attention	Разм. слотов (MetaWorld/Casual World/Robosuite)	32/64/128
Slot Attention	Кол. рекуррентных проходов	5

Таблица 5: COMPAS: гиппер параметры

Параметр	Pong	Casual World	Robosuite	MetaWorld
Разм. модели	128	128	128	64
Разм. состояния	32	32	32	32
Кол. слоев	6	6	6	6
Разм. фильтров CNN	4	4	4	5
Коэф. размерности MLP.	2	2	2	2

Таблица 6: Сравнения качества предсказаний на 1, 5 и 10 шагов вперед по метрике Hits@1 в задаче Causal World Push. Значения в таблице представлены в виде среднее ± стандартное отклонение по 3 различным инициализациям

Модель	1 шаг	5 шагов	10 шагов
CSWM	0.18 ± 0.05	0.15 ± 0.04	0.06 ± 0.02
C-SWM Hard Attention	0.04 ± 0.01	0.03 ± 0.02	0.01 ± 0.00
C-SWM Soft Attention	0.20 ± 0.06	0.15 ± 0.03	0.06 ± 0.01
COMPAS	**0.97 ± 0.01**	**0.97 ± 0.00**	**0.73 ± 0.01**

Таблица 7: Сравнения качества предсказаний на 1, 5 и 10 шагов вперед по метрике MRR в среде Causal World (Push). Значения в таблице представлены в виде среднее ± стандартное отклонение по 3 различным инициализациям

Модель	1 шаг	5 шагов	10 шагов
CSWM	0.33 ± 0.05	0.27 ± 0.06	0.14 ± 0.05
C-SWM Hard Attention	0.11 ± 0.03	0.07 ± 0.03	0.04 ± 0.01
C-SWM Soft Attention	0.35 ± 0.08	0.26 ± 0.06	0.11 ± 0.02
COMPAS	**0.97 ± 0.01**	**0.95 ± 0.01**	**0.89 ± 0.01**

Таблица 8: Сравнения качества предсказаний на 1, 5 и 10 шагов вперед по метрике Hits@1 в задаче Causal World (Stack). Значения в таблице представлены в виде среднее ± стандартное отклонение по 3 различным инициализациям

Модель	1 шаг	5 шагов	10 шагов
C-SWM	0.03 ± 0.01	0.02 ± 0.01	0.03 ± 0.01
C-SWM Hard Attention	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
C-SWM Soft Attention	0.04 ± 0.01	0.04 ± 0.02	0.03 ± 0.01
COMPAS	**0.94 ± 0.01**	**0.93 ± 0.01**	**0.74 ± 0.01**

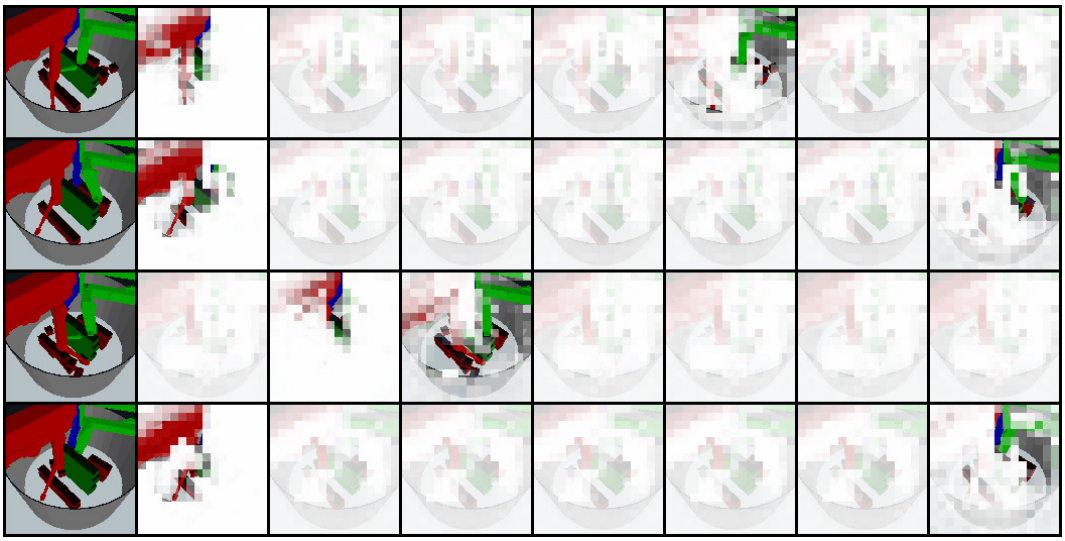


Рис. 16: Визуализация карт внимания слотов для среды Causal World (Stack). Первый столбец – исходное изображение, все остальные – визуализации масок внимания для каждого слота

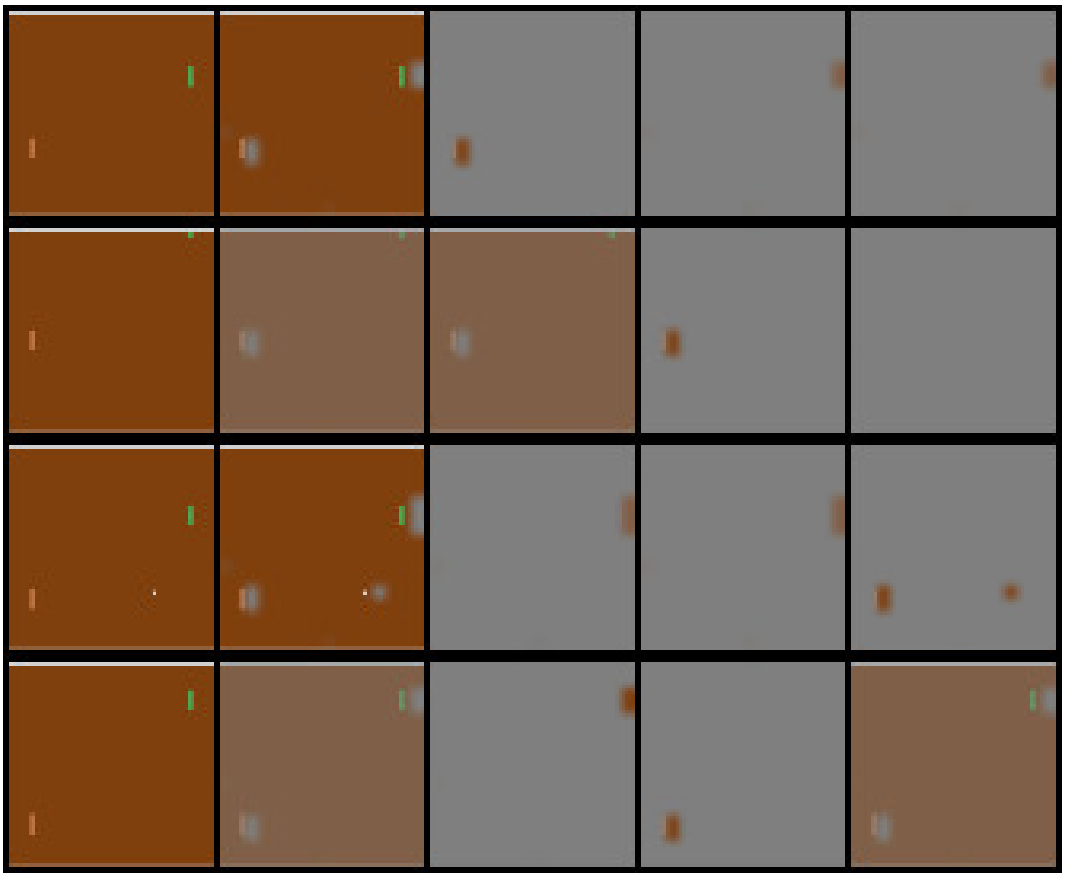


Рис. 17: Визуализация карт внимания слотов для среды Atari Pong. Первый столбец – исходное изображение, все остальные – визуализации масок внимания для каждого слота

на наборе данных RoboSuite, который был собран с использованием оптимальной стратегии. Визуализация карт внимания слотов представлена на Рисунке **??** Од-

Таблица 9: Сравнения качества предсказаний на 1, 5 и 10 шагов вперед по метрике MRR в среде Causal World (Stack). Значения в таблице представлены в виде среднее ± стандартное отклонение по 3 различным инициализациям

Модель	1 шаг	5 шагов	10 шагов
C-SWM	0.08 ± 0.02	0.08 ± 0.02	0.07 ± 0.01
C-SWM Hard Attention	0.02 ± 0.01	0.03 ± 0.00	0.04 ± 0.00
C-SWM Soft Attention	0.12 ± 0.02	0.12 ± 0.04	0.08 ± 0.03
COMPAS	**0.96 ± 0.01**	**0.95 ± 0.01**	**0.88 ± 0.01**

Таблица 10: Сравнения качества предсказаний на 1, 5 и 10 шагов вперед по метрике MRR в среде Robosuite. Значения в таблице представлены в виде среднее ± стандартное отклонение по 3 различным инициализациям

Модель	1 шаг	5 шагов	10 шагов
C-SWM	0.86 ± 0.02	0.34 ± 0.03	0.17 ± 0.02
C-SWM Hard Attention	0.85 ± 0.05	0.32 ± 0.02	0.16 ± 0.02
C-SWM Soft Attention	0.75 ± 0.15	0.30 ± 0.15	0.15 ± 0.07
COMPAS	**0.94 ± 0.01**	**0.88 ± 0.00**	**0.80 ± 0.01**

Таблица 11: Сравнения качества предсказаний на 1, 5 и 10 шагов вперед по метрике HITS@1 в среде Robosuite. Значения в таблице представлены в виде среднее ± стандартное отклонение по 3 различным инициализациям

Модель	1 шаг	5 шагов	10 шагов
C-SWM	0.78 ± 0.02	0.20 ± 0.03	0.08 ± 0.01
C-SWM Hard Attention	0.75 ± 0.07	0.21 ± 0.02	0.08 ± 0.01
C-SWM Soft Attention	0.64 ± 0.20	0.20 ± 0.15	0.14 ± 0.07
COMPAS	**0.92 ± 0.01**	**0.81 ± 0.01**	**0.69 ± 0.01**

нако, COMPAS также превосходит базовые модели на этом наборе данных, о чем свидетельствуют таблицы 10 и 11.

В контексте более простых окружений, таких как Atari Pong, наблюдается, что модели, основанные на C-SWM, достигают близких к оптимальным результатов при использовании предварительно обученных агентов во время сбора данных [13]. Однако производительность этих моделей значительно снижается при использовании случайной стратегии, что указывает на степень нестабильности в их работе. Ви-

Таблица 12: Сравнения качества предсказаний на 1, 5 и 10 шагов вперед по метрике MRR в среде Atari Pong. Значения в таблице представлены в виде среднее ± стандартное отклонение по 3 различным инициализациям

Модель	1 шаг	5 шагов	10 шагов
C-SWM	0.17 ± 0.01	0.01 ± 0.01	0.03 ± 0.03
C-SWM Hard Attention	0.18 ± 0.02	0.04 ± 0.06	0.02 ± 0.02
C-SWM Soft Attention	0.17 ± 0.00	0.04 ± 0.04	0.05 ± 0.05
COMPAS	**0.97 ± 0.01**	**0.96 ± 0.00**	**0.95 ± 0.01**

Таблица 13: Сравнения качества предсказаний на 1, 5 и 10 шагов вперед по метрике HITS@1 в среде Atari Pong. Значения в таблице представлены в виде среднее ± стандартное отклонение по 3 различным инициализациям

Модель	1 шаг	5 шагов	10 шагов
C-SWM	0.18 ± 0.02	0.01 ± 0.02	0.03 ± 0.03
C-SWM Hard Attention	0.16 ± 0.01	0.03 ± 0.04	0.01 ± 0.01
C-SWM Soft Attention	0.16 ± 0.02	0.03 ± 0.03	0.05 ± 0.05
COMPAS	**0.98 ± 0.00**	**0.98 ± 0.00**	**0.98 ± 0.00**

зуализация карт внимания слотов представлена на Рисунке 17. В отличие от этого, предлагаемая модель COMPAS демонстрирует устойчивую производительность независимо от используемой стратегии для сбора данных. Как показано в среде robosuite и продемонстрировано на таблицах 12 и 13, COMPAS во всех экспериментах превосходит модели, основанные на C-SWM, даже при использовании случайной стратегии для сбора данных.

Результаты проведенных выше экспериментов, демонстрируют универсальность и маштабируемость модели. Compas — от более простых, дискретных сред действий, таких как Pong, до более сложных, непрерывных сред действий, таких как Causal World и RoboSuite.

5.3 Обучение с подкреплением

Для тестирования в разработанного метода COMPAS используется среда MetaWorld, задача Push.

Для улучшения стабильности обучения агента, COMPAS предобучался на дата-

сете, собранном случайной стратегией. Датасет для обучения состоял из 3000 эпизодов по 50 шагов каждый. Для валидации использовалось 1000 эпизодов, где в каждом собрано по 50 шагов. Во время обучения агента, экстрактор объектов замораживается, а модель переходов продолжает обучаться

В качестве экстрактора слотов использовалась модель DINOSAUR [27] с SMM [34]. Размер слотов: 32, количество слотов: 8. На Рисунке 19 изображена визуализация карт внимания слотов.



Рис. 18: Сравнение объектно-центричных методов TD-MPC2 с классическим. Object Centric Compas – в объектно-центричной постановке TD-MPC в качестве модели перехода использовался COMPAS. Object Centic MLP – в качестве модели перехода в объектно-центричном TD-MPC использовался MLP. Classical TD-MPC – классическая постановка TD-MPC. По оси x – количество шагов обучения, по y – частота успехов за 20 валидационных эпизодов.

Классический TD-MPC агент обучался со стандартными настройками. Размер наблюдений для агента составлял 84x84.

В качестве мертрики сравнения используется частота успеха во время валидации. График изменения этой метрики во время обучения изображен на рисунке 18

В результате сравнения, COMPAS работает лучше, чем MLP для объектно-центричной постановки, но проигрывает классическому TD-MPC. Если судить по Рисунку 19, то более плохое качество решения задачи обусловлено не качественным выделением слотов. Конкретно, основные объекты которые должны быть разделены, а именно цель, кубик и манипулятор попадают в один слот, что мешает агенту выделить динамические взаимодействия между объектами. Дальнейшее улучшение

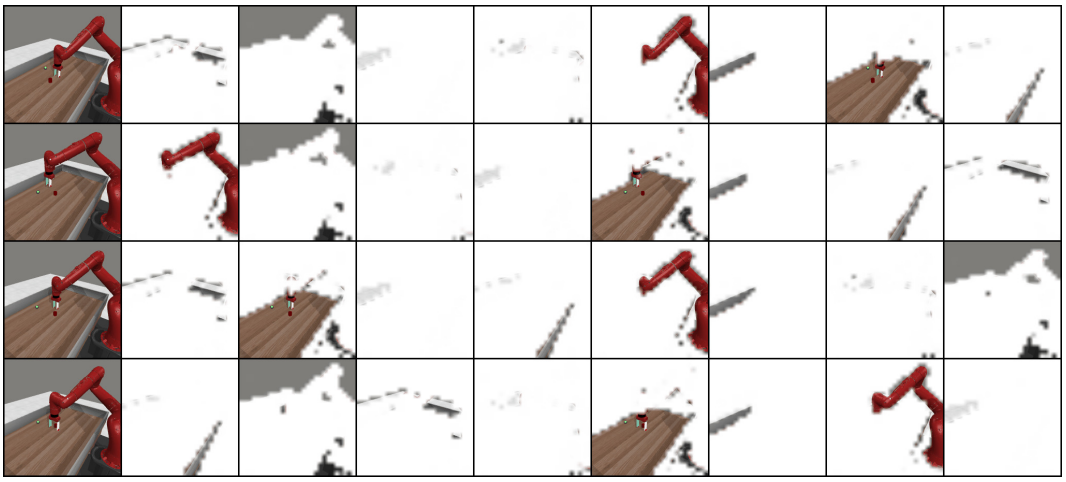


Рис. 19: Визуализация карт внимания слотов для среды Causal World (push). Первый столбец – исходное изображение, все остальные – визуализации масок внимания для каждого слота

методов выделению слотов, могут поспособствовать улучшению качества решения задач обучения с подкреплением.

6 Заключение

Одной из центральных проблем в объектно-центричных моделях мира является сложность обучения динамики в средах со сложными взаимодействиями объектов в зависимости от совершенного действия. Данная работа посвящена решению этой проблемы путем создания модели мира COMPAS. Основными особенностью данной модели являются

Разработанная модель демонстрирует превосходство как и в простых средах, с синтезированными наблюдениями, например Atari Pong, так и в сложных реалистичных симуляторах со сложным физическим взаимодействием объектов, например Causal World, Robosuite.

Также был проведен эксперимент в контексте задачи обучения с подкреплением, где модель перехода COMPAS продемонстрировала превосходство над классическим решением в виде MLP в постановке объектно-центрического обучения.

Тот факт, что модель справляется хуже классического метода говорит о том, что COMPAS зависит от качества объектно-центрических репрезентации, которые формируются в экстракторе объектов из визуальных наблюдений. Повышение качества формирования этих репрезентации является открытой проблемой в данной тематике.

Улучшение алгоритмов выделения объектов и адаптация методов обучения с подкреплением в объектно-центрированную парадигму представляют собой перспективные направления для дальнейших исследований. В этом контексте возможен значительный прирост эффективности модели COMPAS.

Результаты работы были предоставленны:

- COMPAS: Compose Actions and Slots in Object-Centric World Models / D. Kirilenko, V. Vorobyov, A. K. Kovalev, A. I. Panov. IJCAI 2023 NSA: Neuro-Symbolic Agents Workshop. — 2023.

- Object-Centric Learning with Slot Mixture Module / D. Kirilenko, V. Vorobyov, A. K. Kovalev, A. I. Panov // The Twelfth International Conference on Learning Representations. — 2024

- COMPAS: Compose Actions and Slots in Object-Centric World Models / V. Vorobyov, Leonid Anatolievich Ugadiarov, Vladimir Frolov, A. K. Kovalev, A. I.

Panov. Twenty Seventh European Conference On Artificial Intelligence. — 2024 (Находится на ревью)

Список литературы

1. *Diuk C.*, *Cohen A.*, *Littman M. L.* An Object-Oriented Representation for Efficient Reinforcement Learning // Proceedings of the 25th International Conference on Machine Learning. — Helsinki, Finland : Association for Computing Machinery, 2008. — С. 240—247. — (ICML '08). — ISBN 9781605582054. — DOI: 10.1145/1390156.1390187. — URL: https://doi.org/10.1145/1390156.1390187.
2. The Graph Neural Network Model / F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini // IEEE Transactions on Neural Networks. — 2009. — Т. 20, № 1. — С. 61—80. — DOI: 10.1109/TNN.2008.2005605.
3. Empirical evaluation of gated recurrent neural networks on sequence modeling / J. Chung, C. Gulcehre, K. Cho, Y. Bengio // arXiv:1412.3555. — 2014.
4. *Burgard W.*, *Hebert M.*, *Bennewitz M.* World modeling // Springer handbook of robotics. — 2016. — С. 1135—1152.
5. Attention is All you Need / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin // Advances in Neural Information Processing Systems. Т. 30 / под ред. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett. — Curran Associates, Inc., 2017. — URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
6. *Ha D.*, *Schmidhuber J.* World models // arXiv:1803.10122. — 2018.
7. *Oord A. van den*, *Vinyals O.*, *Kavukcuoglu K.* Neural Discrete Representation Learning. — 2018. — arXiv: 1711.00937 [cs.LG].
8. Causalworld: A robotic manipulation benchmark for causal structure and transfer learning / O. Ahmed, F. Träuble, A. Goyal, A. Neitz, Y. Bengio, B. Schölkopf, M. Wüthrich, S. Bauer // arXiv:2010.04296. — 2020.
9. *Greff K.*, *Van Steenkiste S.*, *Schmidhuber J.* On the binding problem in artificial neural networks // arXiv:2012.05208. — 2020.
10. Improving generative imagination in object-centric world models / Z. Lin, Y.-F. Wu, S. Peri, B. Fu, J. Jiang, S. Ahn // ICML. — PMLR. 2020. — С. 6140—6149.
11. *Kipf T. N.*, *Pol E. van der*, *Welling M.* Contrastive Learning of Structured World Models // ICLR. — 2020.

12. Object-Centric Learning with Slot Attention / F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, T. Kipf // arXiv preprint arXiv:2006.15055. — 2020.

13. *Biza O.*, *Pol E. van der*, *Kipf T.* The Impact of Negative Sampling on Contrastive Structured World Models // ICML Workshop: Self-Supervised Learning for Reasoning and Perception. — 2021.

14. D4RL: Datasets for Deep Data-Driven Reinforcement Learning / J. Fu, A. Kumar, O. Nachum, G. Tucker, S. Levine. — 2021. — arXiv: 2004.07219 [cs.LG].

15. Emerging Properties in Self-Supervised Vision Transformers / M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin // Proceedings of the International Conference on Computer Vision (ICCV). — 2021.

16. Learn collision-free self-driving skills at urban intersections with model-based reinforcement learning / Y. Guan, Y. Ren, H. Ma, S. E. Li, Q. Sun, Y. Dai, B. Cheng // 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). — 2021. — C. 3462—3469. — DOI: 10.1109/ITSC48978.2021.9564880.

17. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning / T. Yu, D. Quillen, Z. He, R. Julian, A. Narayan, H. Shively, A. Bellathur, K. Hausman, C. Finn, S. Levine. — 2021. — arXiv: 1910.10897.

18. Zero-Shot Text-to-Image Generation / A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever. — 2021. — arXiv: 2102.12092 [cs.CV].

19. Binding Actions to Objects in World Models / O. Biza, R. Platt, J.-W. van de Meent, L. L. S. Wong, T. Kipf // ICLR 2022 workshop on Objects, Structure and Causality. — 2022.

20. Conditional Object-Centric Learning from Video / T. Kipf, G. F. Elsayed, A. Mahendran, A. Stone, S. Sabour, G. Heigold, R. Jonschkowski, A. Dosovitskiy, K. Greff // International Conference on Learning Representations (ICLR). — 2022.

21. *Hansen N.*, *Wang X.*, *Su H.* Temporal Difference Learning for Model Predictive Control. — 2022. — arXiv: 2203.04955 [cs.LG].

22. robosuite: A Modular Simulation Framework and Benchmark for Robot Learning / Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, Y. Zhu. — 2022. — arXiv: 2009.12293 [cs.RO].

23. SAVi++: Towards end-to-end object-centric learning from real-world videos / G. F. Elsayed, A. Mahendran, S. van Steenkiste, K. Greff, M. C. Mozer, T. Kipf // Advances in Neural Information Processing Systems (NeurIPS). — 2022.

24. *Singh G.*, *Deng F.*, *Ahn S.* Illiterate DALL-E Learns to Compose // ICLR. — 2022.

25. *Singh G.*, *Wu Y.-F.*, *Ahn S.* Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos // Advances in Neural Information Processing Systems / под ред. A. H. Oh, A. Agarwal, D. Belgrave, K. Cho. — 2022. — URL: https://openreview.net/forum?id=eYfIM88MTUE.

26. *Yavas U.*, *Kumbasar T.*, *Ure N. K.* Model-Based Reinforcement Learning for Advanced Adaptive Cruise Control: A Hybrid Car Following Policy // 2022 IEEE Intelligent Vehicles Symposium (IV). — 2022. — С. 1466—1472. — DOI: 10.1109/IV51971.2022.9827279.

27. Bridging the Gap to Real-World Object-Centric Learning / M. Seitzer, M. Horn, A. Zadaianchuk, D. Zietlow, T. Xiao, C.-J. Simon-Gabriel, T. He, Z. Zhang, B. Schölkopf, T. Brox, F. Locatello // In ICLR 2023: The Eleventh International Conference on Learning Representations. — 2023.

28. Daydreamer: World models for physical robot learning / P. Wu, A. Escontrela, D. Hafner, P. Abbeel, K. Goldberg // Conference on Robot Learning. — PMLR. 2023. — С. 2226—2240.

29. *Gu A.*, *Dao T.* Mamba: Linear-Time Sequence Modeling with Selective State Spaces // arXiv preprint arXiv:2312.00752. — 2023.

30. *Lin B.*, *Bouneffouf D.*, *Rish I.* A Survey on Compositional Generalization in Applications // arXiv:2302.01067. — 2023.

31. SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models / Z. Wu, N. Dvornik, K. Greff, T. Kipf, A. Garg. — 2023. — arXiv: 2210.05861 [cs.CV].

32. *Hansen N.*, *Su H.*, *Wang X.* TD-MPC2: Scalable, Robust World Models for Continuous Control // International Conference on Learning Representations (ICLR). — 2024.

33. Mastering Diverse Domains through World Models / D. Hafner, J. Pasukonis, J. Ba, T. Lillicrap. — 2024. — arXiv: 2301.04104 [cs.AI].

34. Object-Centric Learning with Slot Mixture Module / D. Kirilenko, V. Vorobyov, A. K. Kovalev, A. I. Panov // The Twelfth International Conference on Learning Representations. — 2024. — URL: https://openreview.net/forum?id=aBUidW4Nkd.