


Обучение с подкреплением: теоретические основы и алгоритмические реализации

Александр Панов

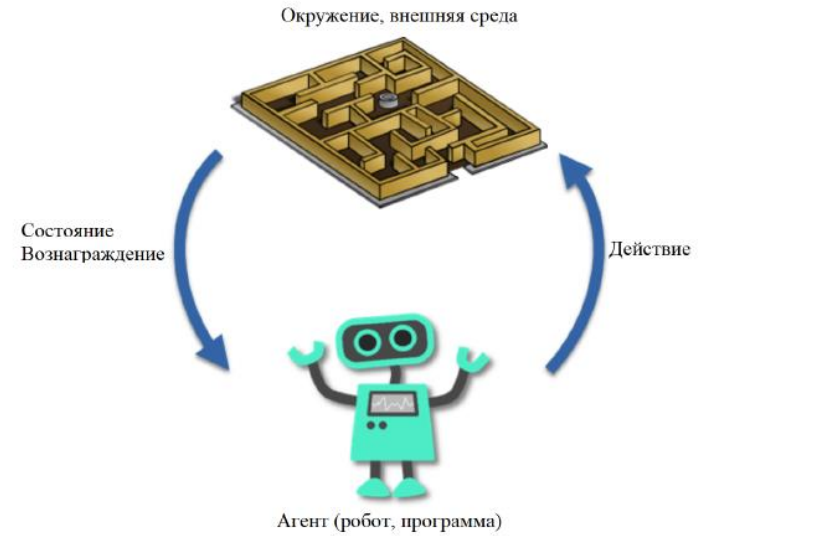
в.н.с. ФИЦ ИУ РАН & AIRI, директор ЦКМ МФТИ

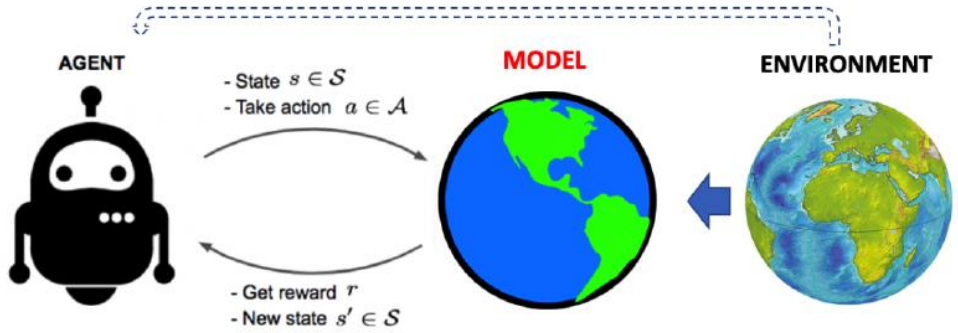
Содержание

01

Обучение с подкреплением: теоретические основы

Взаимодействие агента и среды

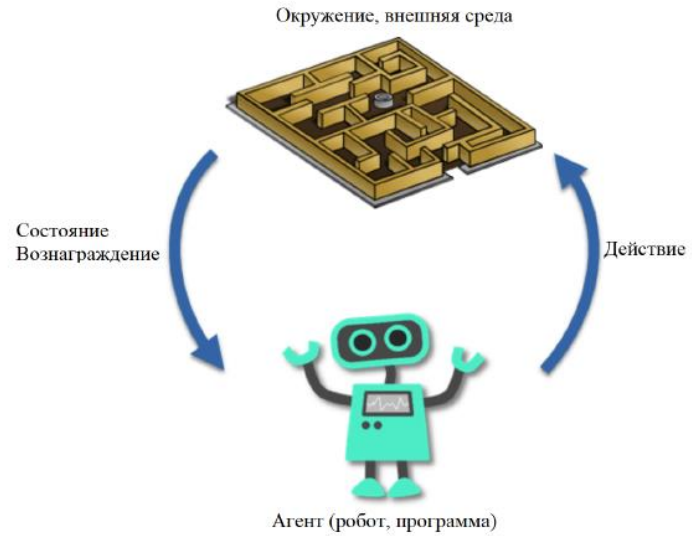


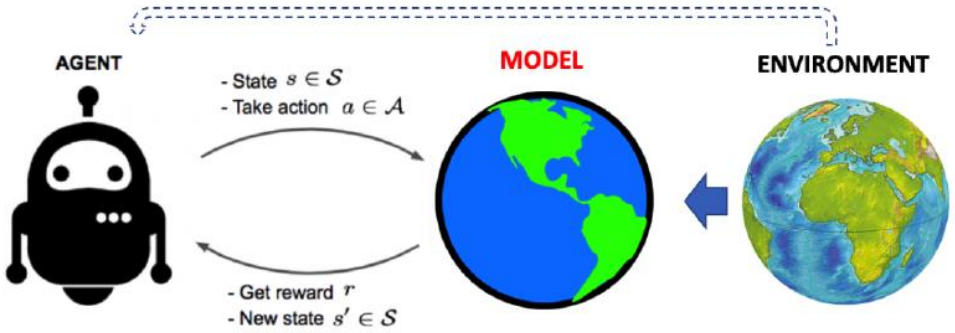


- Агент (принимающего решения на основе данных) отделен от среды (источника данных)
- Действия агенты влияют на поступающие данные в будущем
- Обратная связь от среды может приходить с задержкой
- Особую роль играет параметр времени – причинно-следственные связи в последовательности данных

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning. An Introduction.* 2018.
Laura Graesser and Wah Loon Keng. *Foundations of Deep Reinforcement Learning: Theory and Practice in Python.* 2020.

RL – особый вид машинного обучения





Ключевые особенности:

- Нестационарная целевая переменная (non-stationary target) -> память прецедентов (replay buffers)
- Скоррелированные данные (not i.i.d.) -> память прецедентов
- Частичная наблюдаемость -> рекуррентные модели и планирование
- Неустойчивость процессе улучшения стратегии агента -> алгоритмы оптимизации с ограничениями (PPO)
- Эффективность выборки в среде (sample inefficiency) -> перенос модели обучения (transfer learning)

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning. An Introduction*. 2018.
Laura Graesser and Wah Loon Keng. *Foundations of Deep Reinforcement Learning: Theory and Practice in Python*. 2020.

Формальная постановка задачи

Пусть $< S, A, T, R, \ \gamma >$ - марковский процесс принятия решений (МППР), где:

- S – пространство **состояний** (информационных),
- A – множество **действий** (дискретных, непрерывных),
- $T: S \times A \to S$ – функция переходов (не известна агенту),
- $R: S \times A \to \mathbb{R}$ - функция вознаграждений (не известна агенту),
- γ – дисконтирующий множитель

Агент выполняет действия в среде, используя функцию **стратегии** (стохастическую или детерминированную)

$$\pi: S \to A$$

Цель агента — максимизировать ожидаемую **отдачу** по стратегии π:

$$\mathbb{E}_{\pi} \sum_{t=0}^{\tau} \gamma^t R(s_t, a_t)$$

Необходимо **исследовать** среду, прежде чем формировать стратегию на накопленных данных

AIRI

Связь с задачей оптимального управления

Математическая модель многих систем управления задана обыкновенным дифференциальным уравнением вида

$$\dot{x}(t) = f[x(t), u(t), t], \qquad t \in [t_0, t_1], x(t_0) = a, u(t) \in \Omega$$

- $x(t) \in \mathbb{R}^n$ - состояние управляемой системы,
- $u(t) \in \mathbb{R}^m$ - управляющее воздействие из множества допустимых управлений Ω

Требуется минимизировать функционал качества:

$$J \coloneqq \int_{t_0}^{t_1} \varphi[x(t), u(t), t]dt + g[x(t_1)] \to \min$$

Особенности и различия от постановки задачи RL:

- Рассматривается непрерывное время
- Непрерывный функционал качества, функции f, φ, g – дифференцируемы,
- Функция f – задана (известна модель управляемой системы)

Эргодичность МППР

Динамика в МППР является эргодичной тогда и только тогда, когда какое-либо состояние достижимо из любого другого состояния с помощью некоторой стратегии:

$$\forall s, s' \exists \pi_r \colon \mathbb{E}_\beta \mathbb{E}^P_{s,\pi_r}[B_{s'}] = 1,$$

$B_{s'}$ - индикаторная случайная величина, определяющая, что система попадет в состояние s' хотя бы один раз:

$$B_{s'} = 1\{\exists t < \infty \colon s_t = s'\}$$

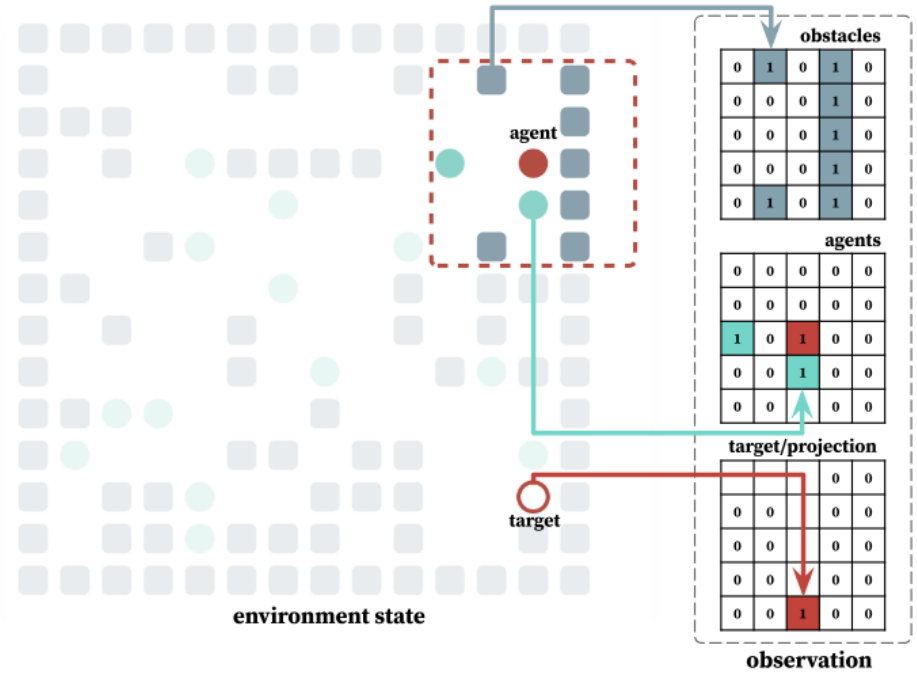
β – Байесовская вероятностная мера динамики в МППР, P – функция переходов МППР

При правильном исследовании среды – динамика близка к эргодической

Teodor Moldova et al. Safe Exploration in Markov Decision Processes. https://arxiv.org/abs/1205.4810

Наблюдение и аппроксимация состояния

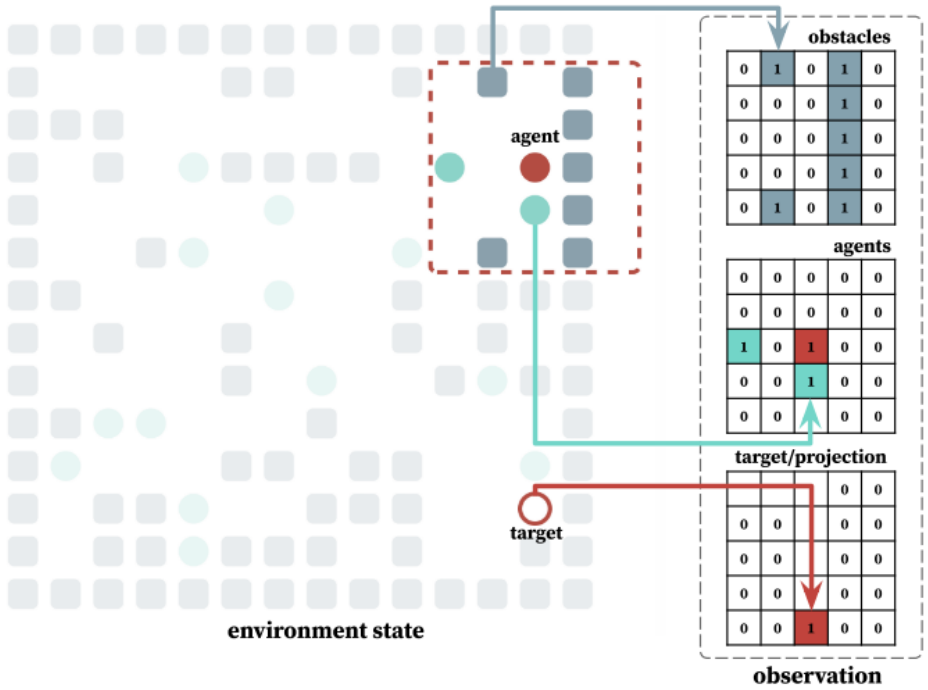
- Также необходимо отметить, что во многих средах условие полной наблюдаемости среды не выполняется
- Агент не имеет непосредственного доступа к информационному состоянию s_t в каждый момент времени, а получает от среды только так называемое наблюдение $o_t \in O$
- Последовательность прецедентов $o_1, a_1, r_2, o_2, a_2, r_3, \ldots$ уже не будет являться марковским процессом
- Формально такой процесс называется *частично наблюдаемым марковским процессом*
- Стандартной практикой в этом случае является введение некоторой функции $s_t \approx h(o_t, o_{t-1}, \ldots)$ от истории наблюдений (полной или с некоторым горизонтом)



Функция полезности

Состояния и наблюдения агента на примере клеточной среды:

- a_t — перемещения из одной клетки в соседнюю не занятую,
- $o_t \in \mathbb{R}^{(2d)^2}$ — **наблюдения** агента,
- $\hat{s}_t = h(o_1, \dots, o_t)$ — функция **аппроксимации** состояния (рекуррентная нейронная сеть)



Функция полезности Q – ожидаемая отдача для текущего состояния и действия:

$$Q(s_t, a_t) = \mathbb{E}_{\pi}\left[\sum_{i=t}^{\tau} \gamma^i R(s_i, a_i)\right]$$

Уравнение Беллмана для оптимальной функции полезности:

$$Q^*(s, a) = \mathbb{E}_{s_t \sim T}\left[r_t + \gamma \max_{a_t} Q^*(s_t, a_t) \,|s, a\right]$$

Аппроксимация функции полезности

- Уравнение Беллмана обычно решается итеративными методами и введением аппроксимации функции полезности: $\hat{Q}(s, a; \theta) \approx Q(s, a)$
- Для поиска оптимальных значений параметров θ вводится функция потерь:

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a\sim\mathcal{D}}[(y - Q(s, a; \theta))^2]$$

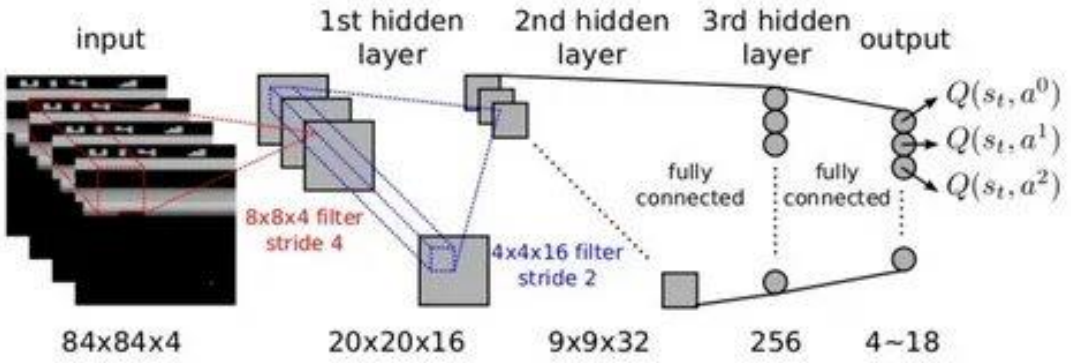
$$y = \mathbb{E}_{s_t\sim T}\left[r_t + \gamma \max_{a_t} Q\ (s_t, a_t; \theta)\ |s, a\right]$$

- Поиск минимума такой функции потерь можно проводить градиентными методами:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{s,a\sim\mathcal{D};s_t\sim T}\left[\left(r_t + \gamma \max_{a_t} Q\ (s_t, a_t; \theta) - Q(s, a; \theta)\right) \nabla_\theta \mathrm{Q}(\mathrm{s}, \mathrm{a}; \theta)\right]$$

AIRI

Алгоритм DQN



Для решения проблемы нестационарного целевого значения используется память прецедентов (replay buffer) и фиксация Q-показателя:

Наблюдаемые переходы сохраняются в память прецедентов

$$(s_t, a_t, r_{t+1}, s_{t+1}) \rightarrow \mathcal{D}$$

В процессе обучения выбирается случайный мини-пакет данных (s, a, r, s') из $\mathcal{D}$

Для вычисления Q-показателя используются «замороженные» веса с предыдущей итерации:

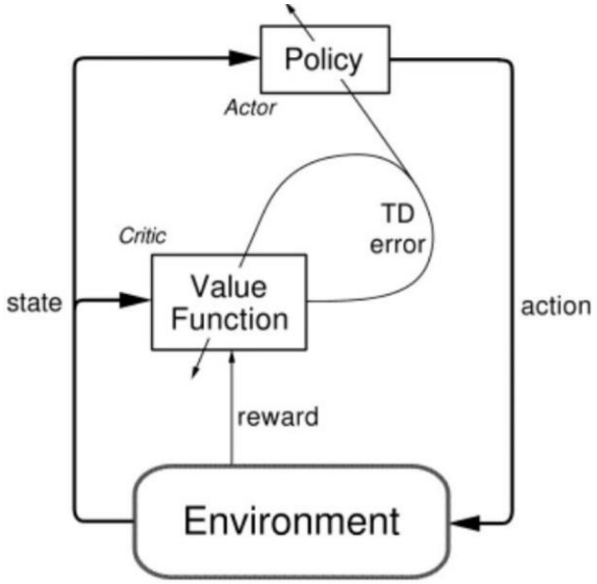
$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s' \sim \mathcal{D}} \left[\left(r_t + \gamma \max_{a'} Q\left(s', a'; \theta^-\right) - Q(s, a; \theta) \right)^2 \right]$$

Градиент стратегии и актор-критик

- Введем напрямую параметризацию стратегии: $\hat{\pi}(s; w) = \mathbb{P}[a|s, w]$
- Пусть задана дифференцируемая функция полезности стратегии J, тогда справедлива теорема о градиенте стратегии:

$$\nabla_w J(w) = \mathbb{E}_{\hat{\pi}(w)}\left[\nabla_w \log \hat{\pi}(s; w)\, Q^{\hat{\pi}(w)}(s, a)\right]$$

- Для оценки значения функции полезности Q используется критик – получается архитектура актора-критика:
  - веса критика обновляются с помощью функции потерь $\mathcal{L}(\theta)$, обычно чаще, чем веса стратегии,
  - веса актора обновляются в соответствии с максимизацией функции полезности $J(w)$

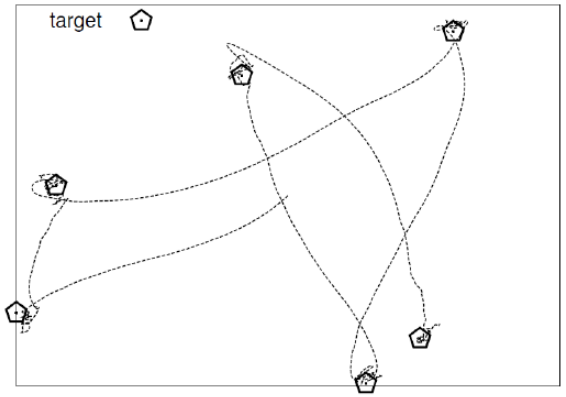


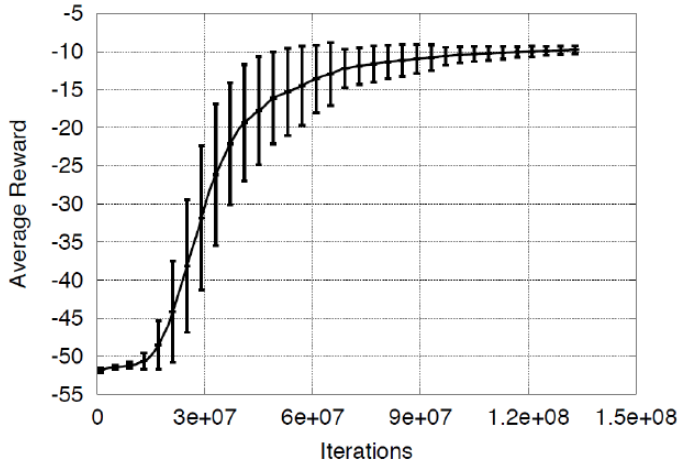
Алгоритм REINFORCE

Монте-Карло оценка функции полезности критика:

$$R_t \approx Q^{\hat{\pi}(w)}(s_t, a_t)$$

Несмещенная оценка с высокой дисперсией





Algorithm 1 REINFORCE

function REINFORCE
    инициализируем w;
    for каждого эпизода $\{s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T\} \approx \hat{\pi}(\mathrm{w})$ **do**
        for $t = 1$ и до $T - 1$ **do**
            $\mathrm{w} \leftarrow \mathrm{w} + \alpha \nabla_{\mathrm{w}} \log \hat{\pi}(s_t, a_t, \mathrm{w}) R_t$
    return w

Совместимые функции аппроксимации

Если удовлетворены следующие два условия:

- аппроксиматор функции полезности совместим со стратегией:

$$\nabla_\theta Q(s,a;\theta) = \nabla_\phi \log \pi(s;\phi),$$

- параметры функции полезности минимизируют среднеквадратичную ошибку:

$$\epsilon = \mathbb{E}_{\pi(\phi)}\left[\left(Q^{\pi(\phi)}(s,a) - Q(s,a;\phi)\right)^2\right],$$

- тогда градиент стратегии в точности равен

$$\nabla_\phi J(\phi) = \mathbb{E}_{\pi(\phi)}\left[\nabla_\phi \log \pi(s;\phi)\, Q(s,a;\theta)\right]$$

AIRI

Нижняя граница полезности стратегии

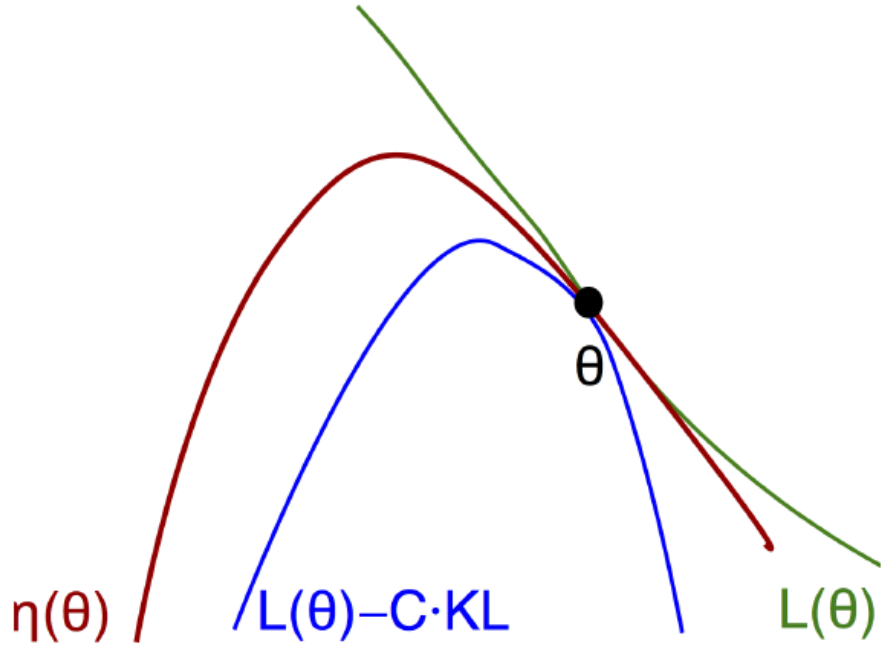
Пусть общее статистическое расстояние (total variance)

$$D_{TV}^{max}(\pi_1, \pi_2) = \max_s D_{TV}^{max}(\pi_1(\cdot\,|s), \pi_2\pi_1(\cdot\,|s))$$

тогда

$$J^{\hat{\pi}(\phi')} \geq L_{\hat{\pi}(\phi)}(\hat{\pi}(\phi')) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\,(D_{TV}^{max}\hat{\pi}(\phi), \hat{\pi}(\phi'))$$

где $\epsilon = \max_{s,a} \left| A^{\hat{\pi}(\phi)}(a, s) \right|$

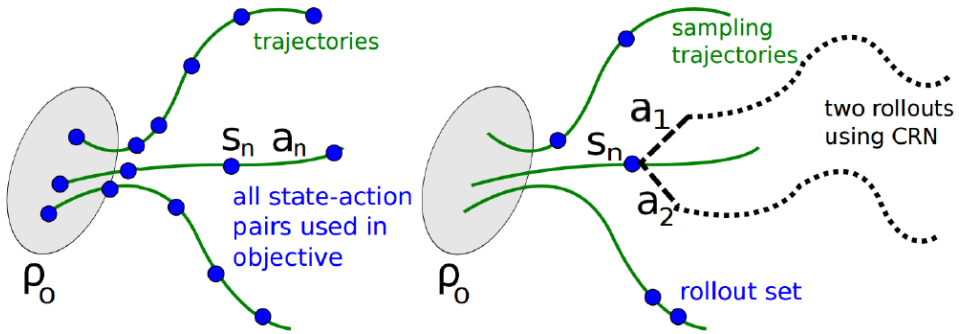


Упрощенная оптимизация градиента

- Хотим добиться монотонности улучшения стратегия за счет подбора шага градиентного спуска
- Оптимизационная задача с ограничением:

$$\max_{w} \mathbb{E}_{s\sim\mu_{w_{old}},a\sim\pi(w_{old})}\left[\frac{\pi(a|s,w)}{\pi(a|s,w_{old})}Q(s,a;w_{old})\right]$$

$$\text{при условии } D_{KL}^{s\sim\mu_{w_{old}}}\left(\pi(w_{old}),\pi(w)\right) \le \delta$$



По сути максимизируется суррогатная функция потерь:

$$L_{w_{old}}(w) = J(w_{old}) + \sum_{s}\mu_{w_{old}}(s)\sum_{a}\pi(a|s,w)A(s,a;w_{\text{old}})$$

Можно облегчить решение оптимизационной задачи введением расписания для константны δ либо усечением функции потерь L – алгоритм PPO

AIRI

Алгоритм мягкого актора-критика

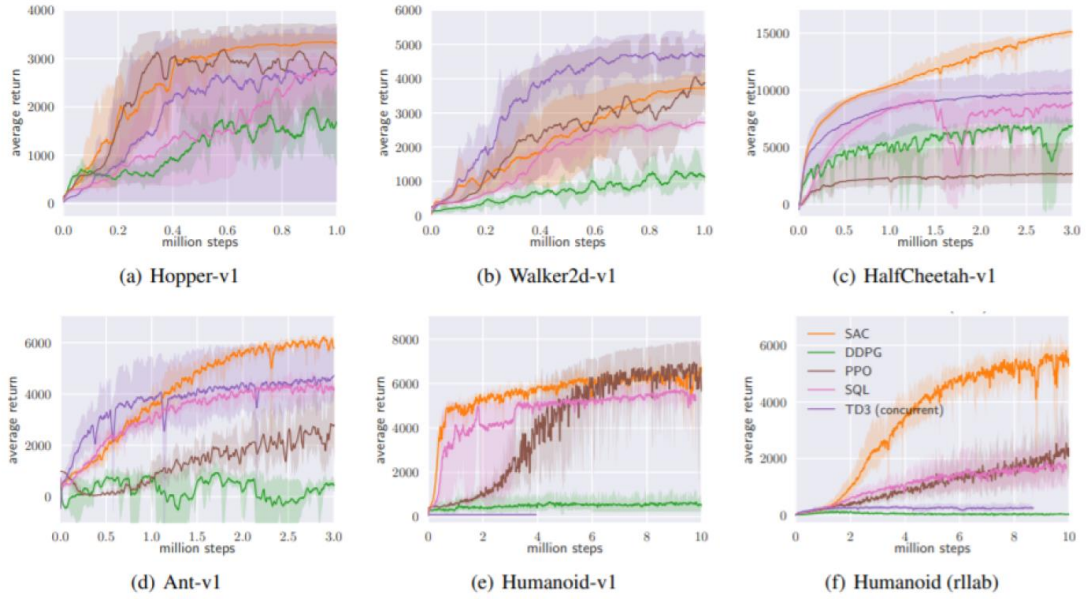
Энтропийная регуляризация – псевдовознаграждения за разнообразие стратегии:

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s' \sim \mathcal{D}}[(y(r,s') - Q(s,a;\theta_i^-))^2]$$

$$y(r,s') = r + \gamma \min_{i=1,2} Q\,(s',a';\theta_i^-) - \alpha \log \pi(a'|s',w)$$

Стратегия актора получается путем репараметризации с гауссовым шумом $\xi \sim \mathcal{N}(0,1)$:

$$\pi(a|s,w,\xi) = \tanh(\mu(s;w) - \sigma(s;w) \cdot \xi)$$

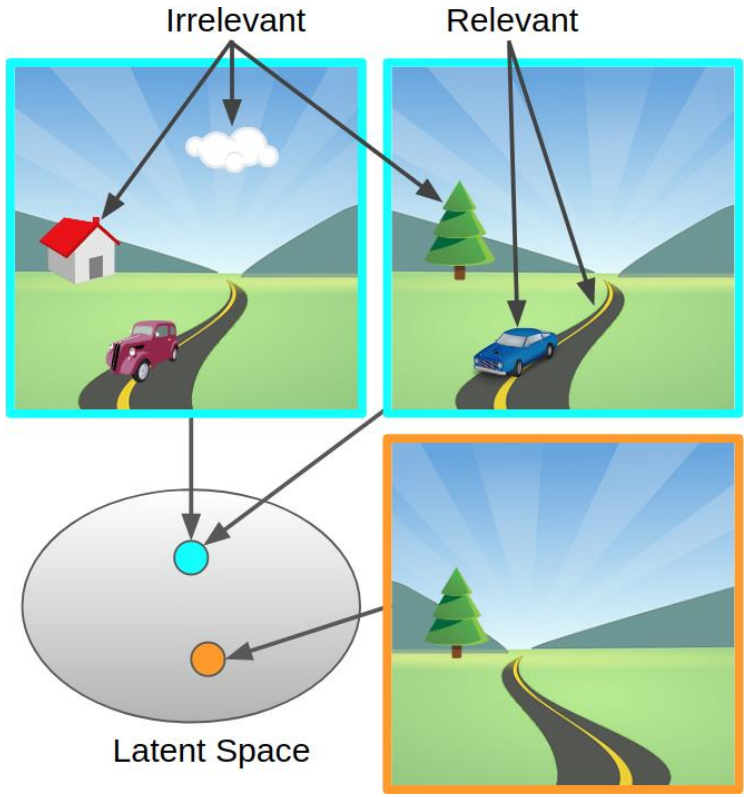


- Обновление актора – максимизация ожидаемой отдачи:
  $$J(w) = \mathbb{E}_{s \sim \mathcal{D}, \xi \sim \mathcal{N}}\left[\min_{i=1,2} Q(s, a(s;\xi,w);\theta_i) - \alpha \log \pi(a(s;\xi,w)|s)\right]$$

AIRI

Инвариантные представления в RL

- Робастные представления визуальных сцен должны быть не чувствительны к нерелевантным задаче объектам
- Необходимо кодировать два наблюдения эквивалентно, если их релевантные детали эквивалентны
- Последовательные автоэнкодеры предназначены для обучения представлений последовательности наблюдений без потерь
- Но такие методы агностичны к задаче



A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine, “Learning Invariant Representations for Reinforcement Learning without Reconstruction,” in ICLR, 2021. http://arxiv.org/abs/2006.10742.

Отношение и метрика бисимуляции

Бисимуляция – это форма абстракции состояния, при которой группируются состояния s_i и s_j, эквивалентные по поведению

Пусть дан МППР M, отношение эквивалентности B между состояния называется отношением бисимуляции, если для всех состояний $s_i, s_j \in S$, которые эквивалентны по B выполняются условия:

- $R(s_i, a) = R(s_j, a)\ \forall a \in A$,
- $\mathcal{P}(G|s_i, a) = \mathcal{P}(G|s_j, a)\ \forall a \in A, \forall G \in S_B$

где S_B - это подмножество S по отношению B и $\mathcal{P}(G|s_i, a) = \sum_{s' \in S} \mathcal{P}(s'|s, a)$.

Метрика бисимуляции определяется как

$$d(s_i, s_j) = \max_{a \in A}(1 - c) \cdot \left|R^a_{s_i} - R^a_{s_j}\right| + c \cdot W_1(\mathcal{P}^a_{s_i}, \mathcal{P}^a_{s_i}; d),$$

где $c \in [0,1)$, а p[th] метрика Вассерштейна (расстояние «земплекопа»):

$$W_p = (\mathcal{P}_i, \mathcal{P}_j; d) = \left(\inf_{\gamma' \in \Gamma(\mathcal{P}_i, \mathcal{P}_j)} \int_{S \times S} d(s_i, s_j)^p d\gamma'(s_i, s_j)\right)^{1/p}$$

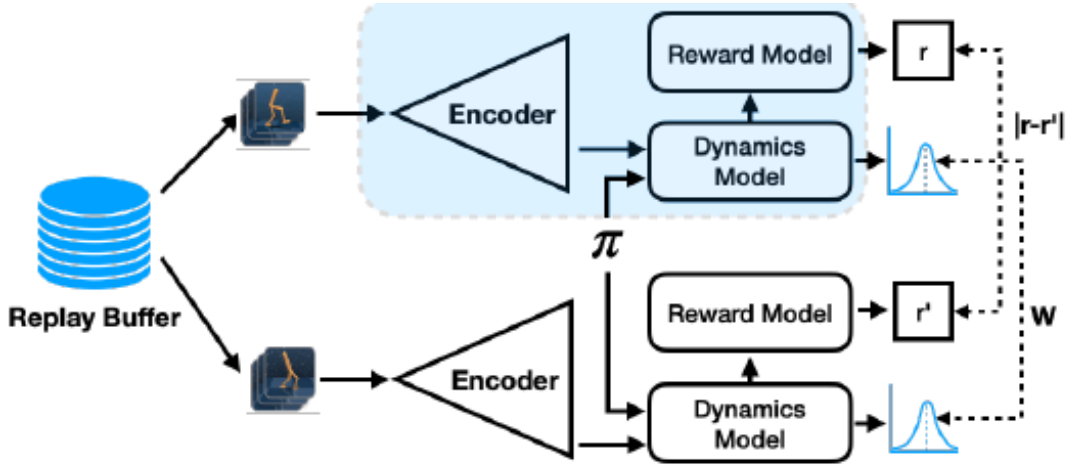
Обучение инвариантных представлений

Algorithm 1 Deep Bisimulation for Control (DBC)

1: **for** Time $t = 0$ to ∞ **do**
2: Encode state $\mathbf{z}_t = \phi(\mathbf{s}_t)$
3: Execute action $\mathbf{a}_t \sim \pi(\mathbf{z}_t)$
4: Record data: $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_{t+1}\}$
5: Sample batch $B_i \sim \mathcal{D}$
6: Permute batch: $B_j = \text{permute}(B_i)$
7: Train policy: $\mathbb{E}_{B_i}[J(\pi)]$ ▷ Algorithm 2
8: Train encoder: $\mathbb{E}_{B_i,B_j}[J(\phi)]$ ▷ Equation (4)
9: Train dynamics: $J(\hat{\mathcal{P}},\phi) = (\hat{\mathcal{P}}(\phi(\mathbf{s}_t), \mathbf{a}_t) - \bar{\mathbf{z}}_{t+1})^2$

Algorithm 2 Train Policy (changes to SAC in blue)

1: Get value: $V = \min_{i=1,2} \hat{Q}_i(\hat{\phi}(\mathbf{s})) - \alpha \log \pi(\mathbf{a}|\phi(\mathbf{s}))$
2: Train critics: $J(Q_i, \phi) = (Q_i(\phi(\mathbf{s})) - r - \gamma V)^2$
3: Train actor: $J(\pi) = \alpha \log p(\mathbf{a}|\phi(\mathbf{s})) - \min_{i=1,2} Q_i(\phi(\mathbf{s}))$
4: Train alpha: $J(\alpha) = -\alpha \log p(\mathbf{a}|\phi(\mathbf{s}))$
5: Update target critics: $\hat{Q}_i \leftarrow \tau_Q Q_i + (1 - \tau_Q)\hat{Q}_i$
6: Update target encoder: $\hat{\phi} \leftarrow \tau_\phi \phi + (1 - \tau_\phi)\hat{\phi}$



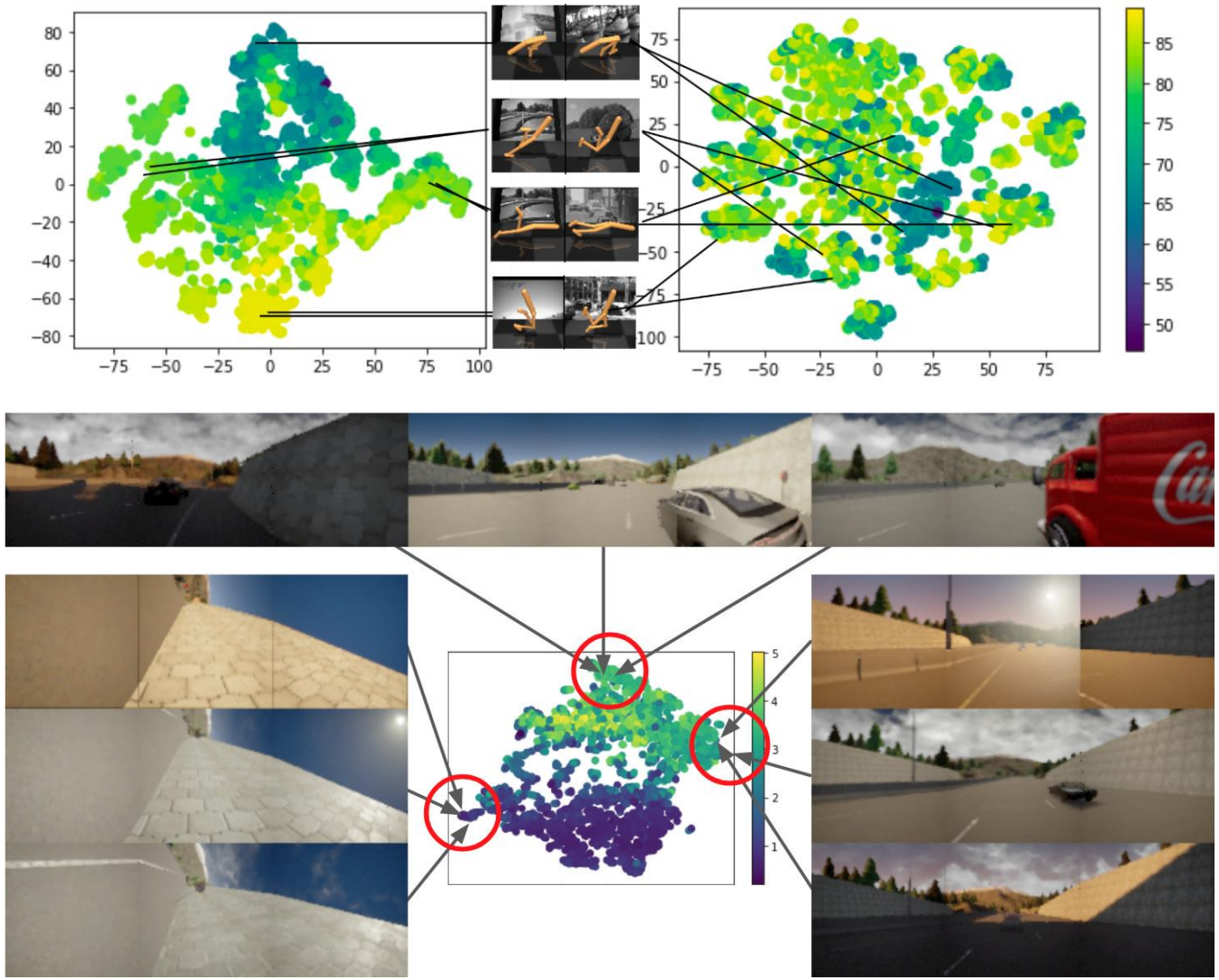
Минимизация MSE между метрикой бисимуляции по актуальному опыту и ℓ_1 расстоянием:

$$J(\phi) = \left(\left|\left| z_i - z_j \right|\right|_1 - |r_i - r_j| - \gamma W_2(\hat{\mathcal{P}}(\cdot\,|\bar{z}_i, a_i), \hat{\mathcal{P}}(\cdot\,|\bar{z}_j, a_j)) \right)^2,$$

где $\hat{\mathcal{P}}$ - гауссово распределение и

$$W_2\left(\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\mu_j, \Sigma_j)\right)^2 = \left|\left|\mu_i - \mu_j\right|\right|_2^2 + \left|\left|\Sigma_i^{1/2} - \Sigma_j^{1/2}\right|\right|_{\mathcal{F}}^2$$

Латентное пространство и метрика бисимуляции



- Более структурирвоанное латентное пространство (по сравнению с обычным VAE)
- Легкая интерпретация наблюдений по картам полезности и определение нерелевантных частей наблюдения
- Теоретические гарантии на оптимальные значения функции полезности

Обучение с подкреплением без модели

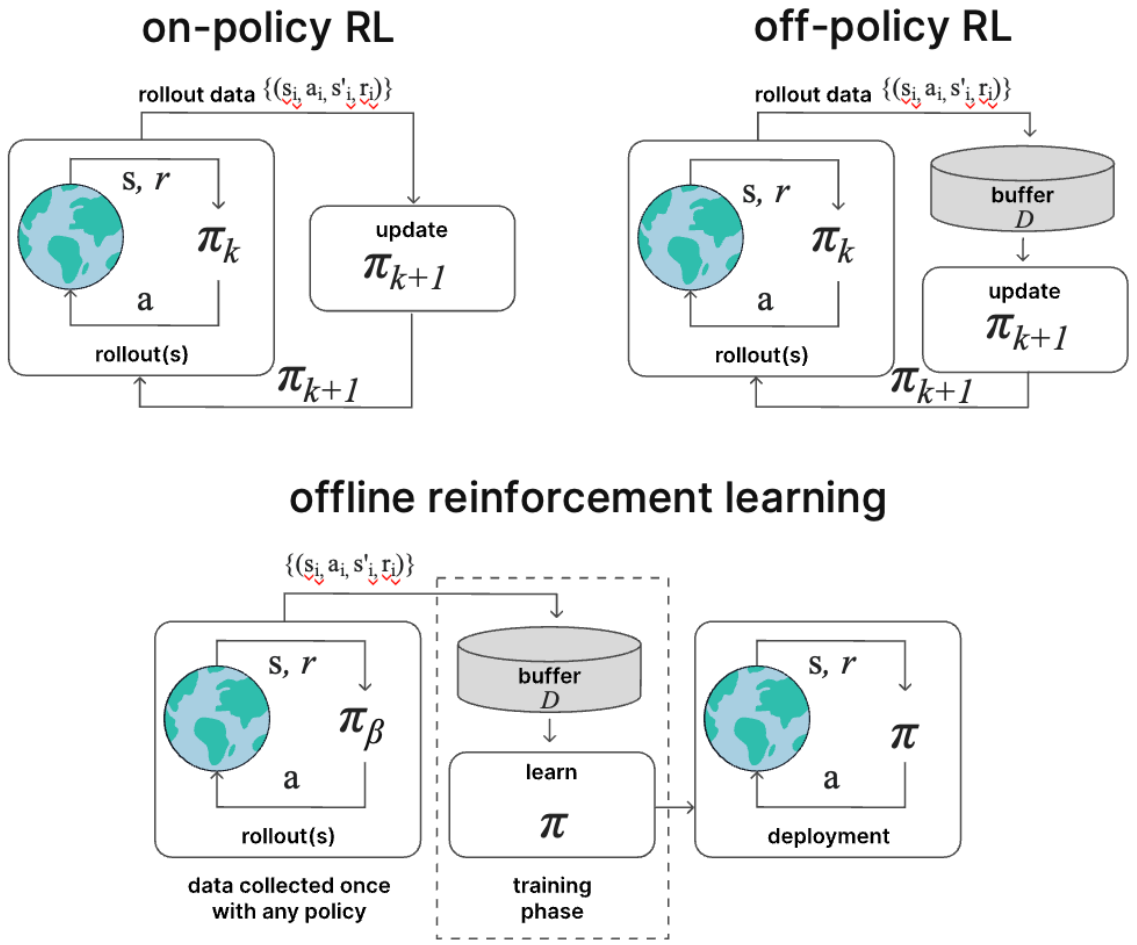
- Успешные примеры безмодельных подходов: Atari, OpenAI Five, AlphaStar, Gato
- Низкая эффективность выборки:
  - Rainbow работал с играми ATARI 2600 требовал от 44 до 200 млн фреймов (от 200 до 900 часов игры) для достижения уровня человека
  - OpenAI Five требовал 11,000+ лет игры в Dota 2
  - AlphaZero использовало 4.9 млн игр в Go
  - Обучение AlphaStar занимало 200 лет игры в Starcraft II
  - Обучение Gato занимало 4 дня на 16x16 TPU v3 кластере
- Перспективные подходы: демонстрации, иерархия, использование модели

Silver D. et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science. 2018.
Berner C. et al. Dota 2 with Large Scale Deep Reinforcement Learning. 2019.
Vinyals O. et al. Alphastar: Mastering the real-time strategy game Starcraft II. DeepMind blog. 2019.

AIRI

Интерактивное и автономное обучение

- **Идея:** использовать большие набора данных истории взаимодействия со средой
- **Задача:** найти хорошие примеры в наборе данных, включающих примеры и плохого поведения
- **Обобщение:** хорошее поведение в одних случаях может приводить к хорошему поведению и в других
- **«Сшивка» (stitching):** части хорошего траекторий можно скомбинировать



S. Levine et al. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. 2020. https://arxiv.org/abs/2005.01643

Постановка задачи обучения с моделью мира

Пусть $< S, A, T, R, G, \gamma >$ - марковский процесс принятия решений, где:

- S – пространство состояний (информационных),
- A – множество действий (дискретных, непрерывных),
- $T: S \times A \to S$ – функция переходов (не известна агенту),
- $R: S \times A \to \mathbb{R}$ - функция вознаграждений (не известна агенту),
- $\boldsymbol{G: S \to \{0, 1\}}$ **– целевая функция, определяющие терминальные состояния,**
- γ – дисконтирующий множитель

Агент строит аппроксимации функции **переходов и функции вознаграждения** $< \hat{T}, \hat{R} >$ и может генерировать предсказания – планы по достижению цели $G\ (s_{n+1}) = 1$:

$$Plan: < s_i, r_i, a_i, \hat{s}_{i+1}, \hat{r}_{i+1}, a_{i+1}, \dots, a_n, \hat{s}_{n+1} >$$

Цель агента – максимизировать ожидаемую отдачу по стратегии π:

$$\mathbb{E}_\pi \sum_{t=0}^{\tau} \gamma^t R(s_t, a_t)$$

Алгоритм MBPO в обучении с моделью

Initialize policy π_ϕ, predictive model p_θ, environment dataset $\mathcal{D}_{\text{env}}$, model dataset $\mathcal{D}_{\text{model}}$
for N epochs **do**
  Train model p_θ on $\mathcal{D}_{\text{env}}$ via maximum likelihood
  for E steps **do**
    Take action in environment according to π_ϕ; add to $\mathcal{D}_{\text{env}}$
    for M model rollouts **do**
      Sample s_t uniformly from $\mathcal{D}_{\text{env}}$
      Perform k-step model rollout starting from s_t using policy π_ϕ; add to $\mathcal{D}_{\text{model}}$
    for G gradient updates **do**
      Update policy parameters on model data: $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi, \mathcal{D}_{\text{model}})$

Обучение модели среды

Обучение стратегии на синтетических данных из модели

Janner, Michael, et al. "When to trust your model: Model-based policy optimization." Advances in neural information processing systems 32 (2019).

AIRI

Гарантии качества для MBPO

Theorem 4.1. *Let the expected TV-distance between two transition distributions be bounded at each timestep by ϵ_m and the policy divergence be bounded by ϵ_π. Then the true returns and model returns of the policy are bounded as:*

$$\eta[\pi] \geq \hat{\eta}[\pi] - \underbrace{\left[\frac{2\gamma r_{\max}(\epsilon_m + 2\epsilon_\pi)}{(1-\gamma)^2} + \frac{4 r_{\max}\epsilon_\pi}{(1-\gamma)}\right]}_{C(\epsilon_m, \epsilon_\pi)} \tag{1}$$

$\eta[\pi]$ - стратегия для истинного МППР

$\hat{\eta}[\pi]$ - стратегия для МППР, индуцированного моделью

$r_{\max}$ - максимальное значение получаемого вознаграждения

$$\max_t \mathbb{E}_{s\sim\pi_{D,t}} [D_{TV}(P(s'|s,a)||\hat{P}(s'|s,a))] \leq \epsilon_m$$

$$\max_s D_{TV}(\pi_D(a|s)||\pi_D(a|s)) \leq \epsilon_\pi$$

02

Мультиагентные задачи в обучении с подкреплением

Задача мультиагентного обучения

Децентрализованный МППР (Dec-POMDP): $G = \langle S, A, U, P, r, Z, O, n, \gamma \rangle$,

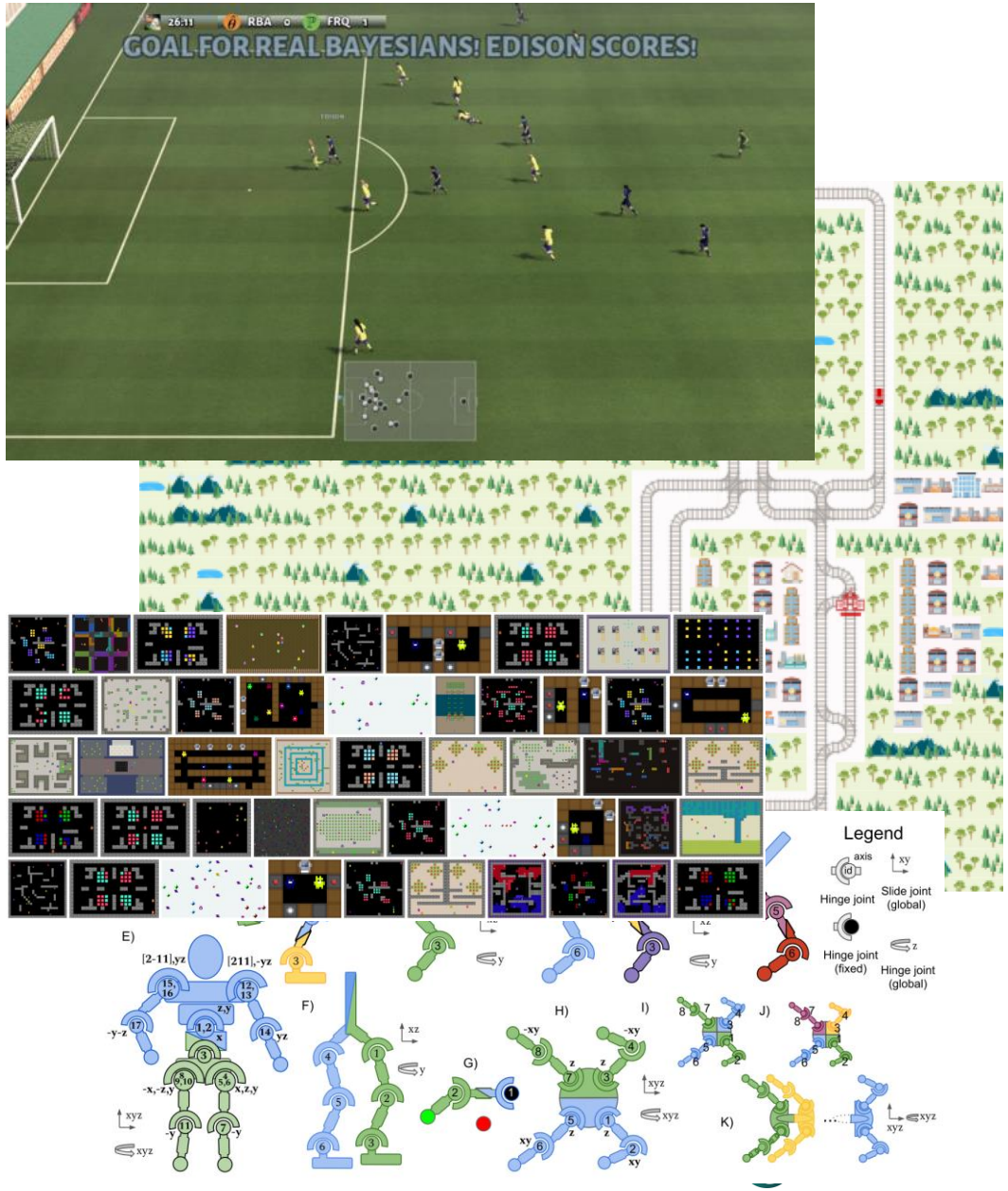
- В каждый момент времени агент $a \in A \equiv 1, \ldots, n$ выбирает действие $u^a \in U$
- Выбранные каждым агентом действия формируют объединенное действие $\mathbf{u} \in \mathbf{U} \equiv U^n$
- Эти действия ведут к переходу окружения в новое состояние в соответствии с функцией переходов $P(s'|s, u): S \times U \times S \to [0, 1]$, в каждый момент времени t
- Вознаграждения генерируются в соответствии с функцией $r(s, \mathbf{u})$: $S \times S \to \mathbf{R}$, которая разделяется всеми агентами, а $\gamma \in [0, 1)$ – это дисконтирующий множитель

Задача мультиагентного обучения

- В каждый момент времени каждый агент получает индивидуальное наблюдение $z^a \in Z$ в соответствие с функцией наблюдения O(s, a): $S \times A \rightarrow Z$
- Каждый агент поддерживает свою историю действие-наблюдение $\tau^a \in T \equiv (Z \times U)^*$, которую обуславливается стратегия агента $\pi^a(u^a|\tau^a)$: $T \times U \rightarrow [0,1]$
- Общая стратегия π ассоциирована с функцией полезности общего действия: $Q^{\pi}(s_t, \mathbf{u}_t) = \mathbb{E}_{s_{t+1:\infty}, \boldsymbol{u}_{t+1:\infty}}[R_t | s_t, \boldsymbol{u}_t]$, где $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ обозначает дисконтированную отдачу
- Цель обучения – найти оптимальную функцию полезности действия

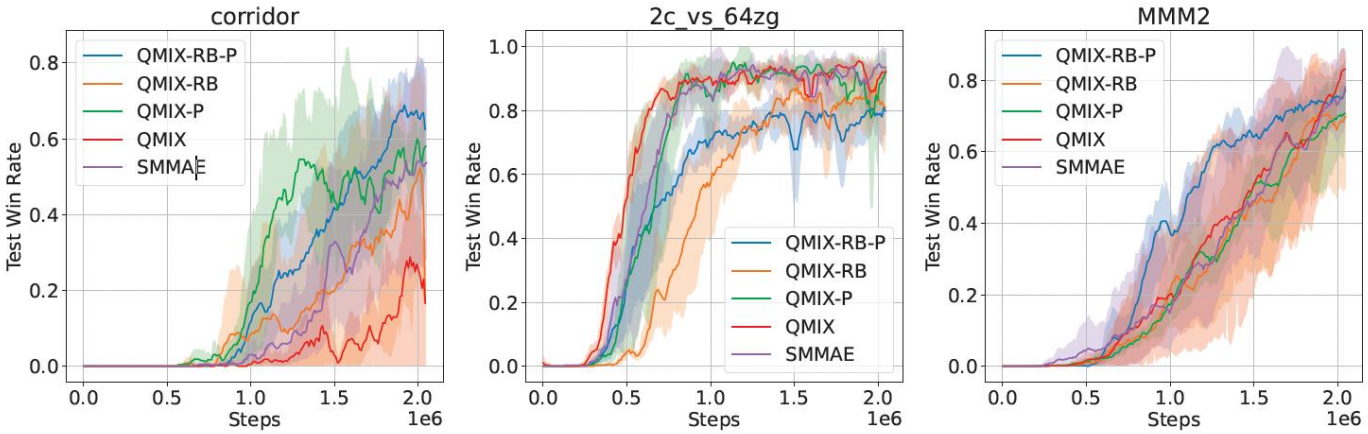
MARL окружения

Environment	Repository	Navigation	Partially observable	Python based	Hardware-agnostic	Performance >10K Steps/s	Procedural generation	Requires generalization	Evaluation protocols	Tests & CI	PyPi Listed	Scalability >1000 Agents	Induced behavior
Flatland [48]	link	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓	✓	Coop
GoBigger [52]	link	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✗	Mixed/Coop
Google Research Football [20]	link	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	Mixed
Griddly [53]	link	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓	✓	Mixed
Hide-and-Seek [43]	link	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	Comp
IMP-MARL [47]	link	✗	✓	✓	✓	✗	✗	✗	✓	✗	✗	✓	Coop
Jumanji (XLA) [42]	link	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓	✗	Mixed
LBF [45]	link	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗	Coop
MAMuJoCo [50]	link	✗	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗	Coop
MATE [46]	link	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	✗	Coop
MeltingPot [44]	link	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	Mixed/Coop
Minecraft MALMO [41]	link	✓	✓	✗	✗	✗	✓	✓	✓	✓	✗	✓	Mixed
MPE [54]	link	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗	Mixed
MPE (XLA) [38]	link	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓	✗	Mixed
Multi-agent Brax (XLA) [38]	link	✗	✓	✓	✗	✓	✗	✗	✗	✓	✓	✗	Coop
Multi-Car Racing [55]	link	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	Comp
Neural MMO [40]	link	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	Comp
Nocturne [49]	link	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✓	Mixed
Overcooked [39]	link	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗	Coop
Overcooked (XLA) [38]	link	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✓	Coop
RWARE [45]	link	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	Coop
SISL [51]	link	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	Coop
SMAC [37]	link	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	Mixed/Coop
SMAC v2 [16]	link	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	Mixed/Coop
SMAX (XLA) [38]	link	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓	✓	Mixed/Coop
POGEMA (ours)	link	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Mixed



SMAC пример: алгоритм ReMIX

- Улучшение современных алгоритмов MARL может быть достигнуто простой модификацией є-жадной стратегии
- Исследование зависит от соотношения доступных совместных действий и количества агентов
- Улучшение памяти прецедцентов, чтобы декоррелировать опыт на основе повторяющихся траеткорий, а не эпизодов



Algorithm 1: Inserting transitions in replay buffer

Input: List of transitions $\mathcal{D}$, buffer size D
Output: List of transitions $\mathcal{D}$

Sample transition tuples $\rho \leftarrow \left\{\left(s_t, r_t\left\{\left(z_t^a, u_t^a, z_{t+1}^a\right)|a=1,...,n\right\}\right)|t=0,...,T-1\right\}$
for *each step* $t=0,...,T-1$ **do**
  if $size(\mathcal{D}) = D$ **then**
    $\mathcal{D} \leftarrow \mathcal{D}[1:]$ `// Pop oldest index`
  end
  $\mathcal{D} \leftarrow \text{concat}(\mathcal{D}, \rho_t)$
end

Algorithm 2: Sample transitions from replay buffer

Input: List of transitions $\mathcal{D}$, sequence size m, batch size B
Output: Batch of transitions $\mathcal{B}$

$\mathcal{B} \leftarrow ()$ `// Initialize batch as an empty list`
while $size(\mathcal{B}) < B$ **do**
  $i \sim \mathcal{U}(0, \text{size}(\mathcal{D}) - 1)$ `// Randomly sample starting index of a sequence`
  if $i + m < size(\mathcal{D})$ **then**
    $b \leftarrow \mathcal{D}[i : i+m]$
  else
    $b \leftarrow \text{concat}(\mathcal{D}[i:], \mathcal{D}[: \text{size}(\mathcal{D}) - i])$
  end
  $\mathcal{B} \leftarrow \text{concat}(\mathcal{B}, b)$
end

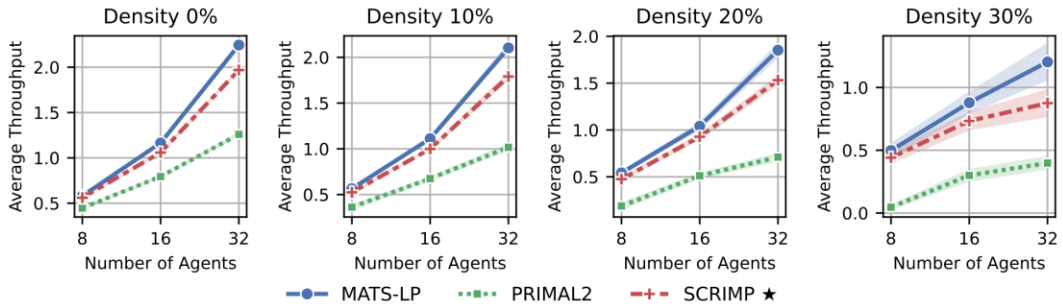
Обучаемые подходы в задаче поиска пути

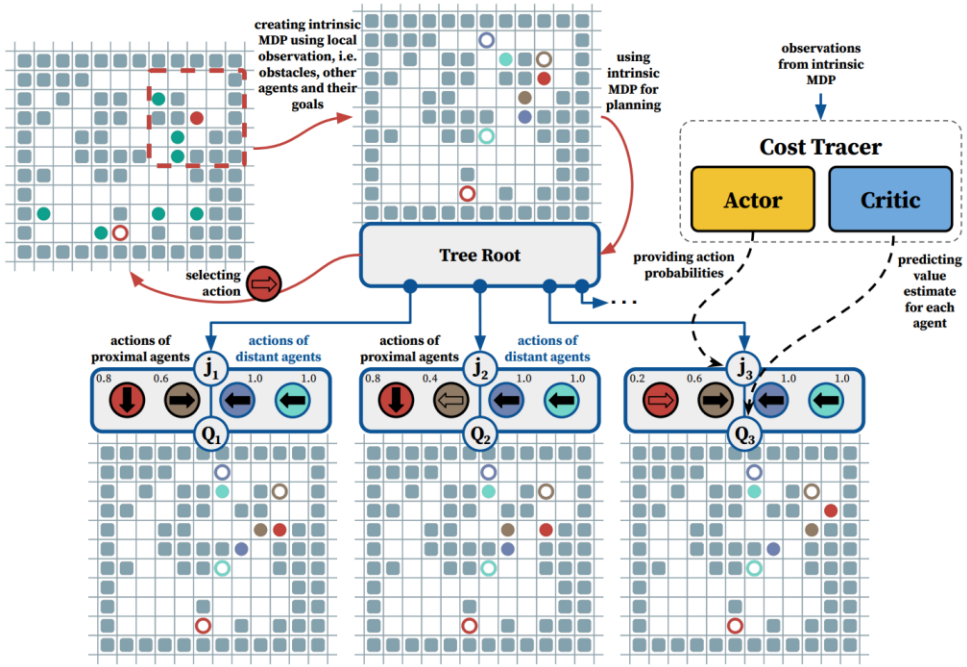
Новые SOTA методы на POGEMA окружении:

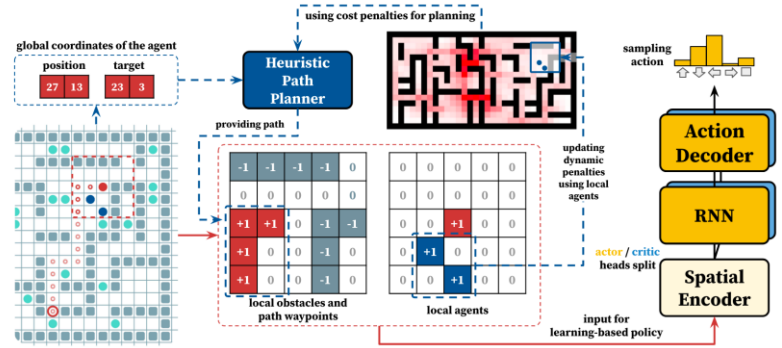
- Switcher – гибридная стратегия с переключением между эвристическим планировщиком и обучаемой распределенной RL-стратегией с памятью

- Follower – эвристический планировщик верхнего уровня для постановки подцелей и локальная RL-стратегия для разрешения конфликтов при достижении подцелей

- MATS-LP – многоагентный поиск по дереву Монте-Карло с предварительно обученными RL-стратегиями для выбора действий в узлах дерева

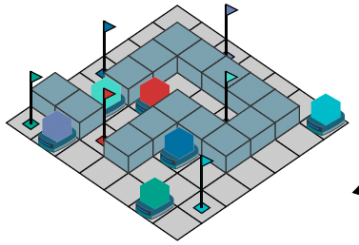




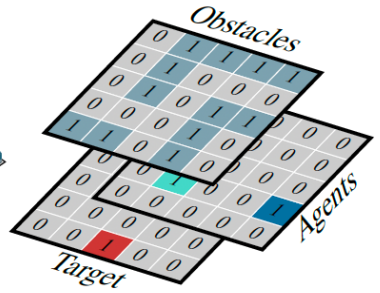


A. Skrynnik et al. Learn to Follow: Decentralized Lifelong Multi-Agent Pathfinding via Planning and Learning. AIRI & MIPT. AAAI 2024.
A. Andreychuk et al. Decentralized Monte Carlo Tree Search for Partially Observable Multi-agent Pathfinding. AIRI. AAAI 2023.

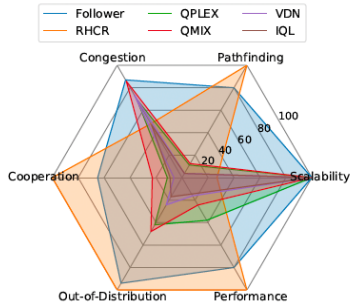
PogemaBench: оценка MAPF алгоритмов



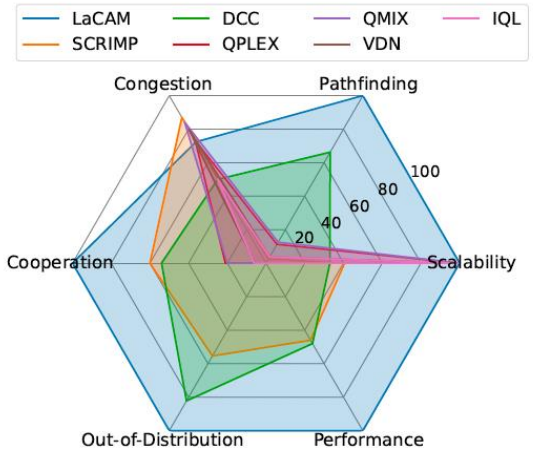
(a) `task`

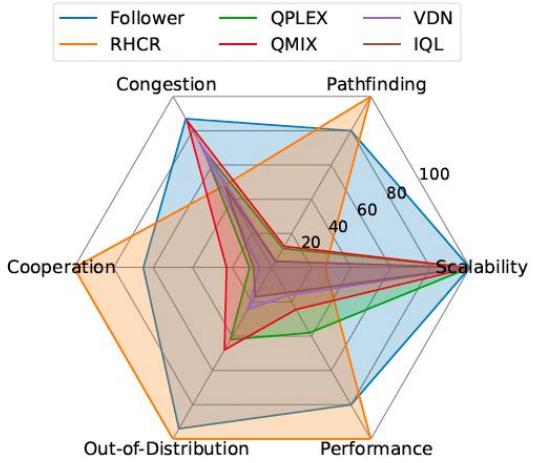


(b) `observation`



(c) `results`



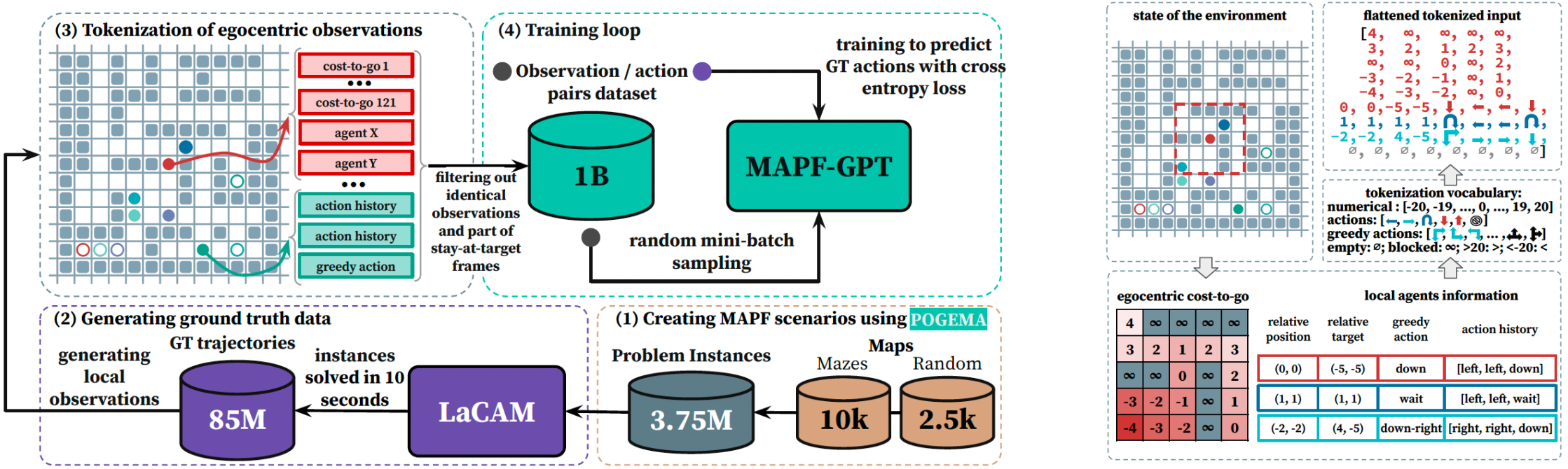


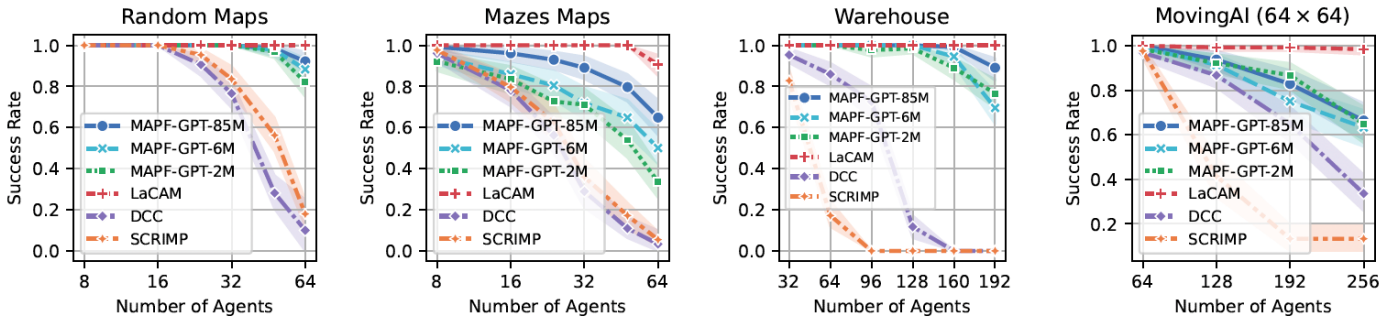
Environment	Repository	Navigation	Partially observable	Python based	Hardware-agnostic	Performance >10K Steps/s	Procedural generation	Requires generalization	Evaluation protocols	Tests & CI	PyPi Listed	Scalability >1000 Agents	Induced behavior
Flatland [48]	link	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓	✓	Coop
GoBigger [52]	link	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✗	Mixed/Coop
Google Research Football [20]	link	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	Mixed
Griddly [53]	link	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓	✓	Mixed
Hide-and-Seek [43]	link	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	Comp
IMP-MARL [47]	link	✗	✓	✓	✓	✗	✗	✗	✓	✗	✗	✓	Coop
Jumanji (XLA) [42]	link	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓	✗	Mixed
LBF [45]	link	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗	Coop
MAMuJoCo [50]	link	✗	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗	Coop
MATE [46]	link	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	✗	Coop
MeltingPot [44]	link	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	Mixed/Coop
Minecraft MALMO [41]	link	✓	✓	✗	✗	✗	✓	✓	✓	✓	✗	✓	Mixed
MPE [54]	link	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗	Mixed
MPE (XLA) [38]	link	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓	✗	Mixed
Multi-agent Brax (XLA) [38]	link	✗	✓	✓	✗	✓	✗	✗	✗	✓	✓	✗	Coop
Multi-Car Racing [55]	link	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	Comp
Neural MMO [40]	link	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	Comp
Nocturne [49]	link	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✓	Mixed
Overcooked [39]	link	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗	Coop
Overcooked (XLA) [38]	link	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✓	Coop
RWARE [45]	link	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	Coop
SISL [51]	link	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	Coop
SMAC [37]	link	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	Mixed/Coop
SMAC v2 [16]	link	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	Mixed/Coop
SMAX (XLA) [38]	link	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓	✓	Mixed/Coop
POGEMA (ours)	link	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Mixed

Skrynnik A. et al. POGEMA: A Benchmark Platform for Cooperative Multi-Agent Navigation. https://arxiv.org/abs/2407.14931
https://github.com/AIRI-Institute/pogema
https://github.com/AIRI-Institute/pogema-benchmark
https://github.com/AIRI-Institute/pogema-toolbox

AIRI

MAPF-GPT: базовая модель для задачи MAPF





Andreychuk A. et al. MAPF-GPT: Imitation Learning for Multi-Agent Pathfinding at Scale. 2024

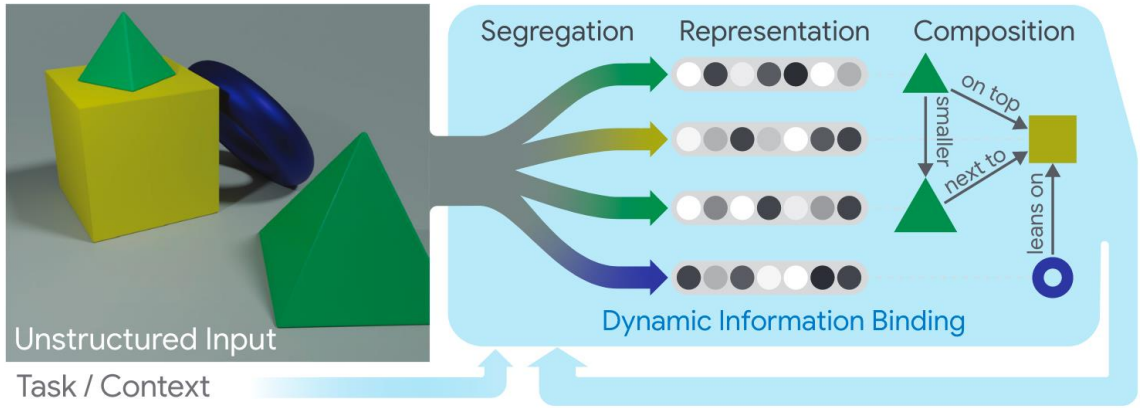
AIRI

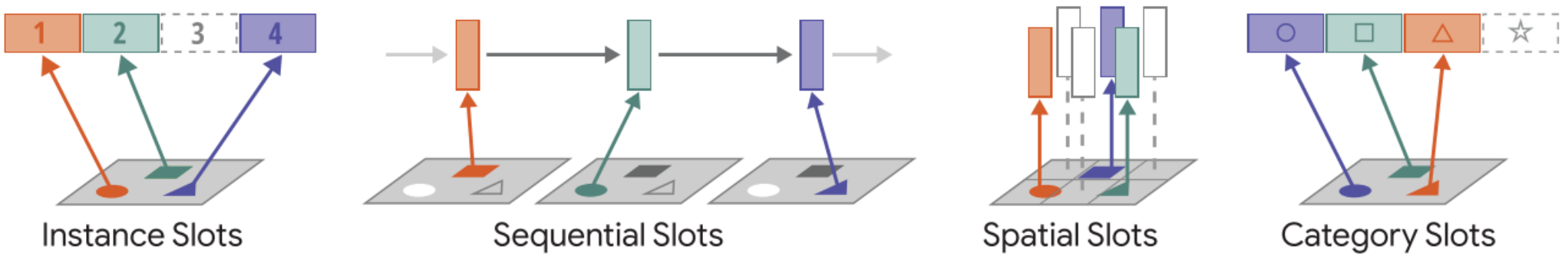
03

Объектная декомпозиция модели мира

Объектно-центричные представления

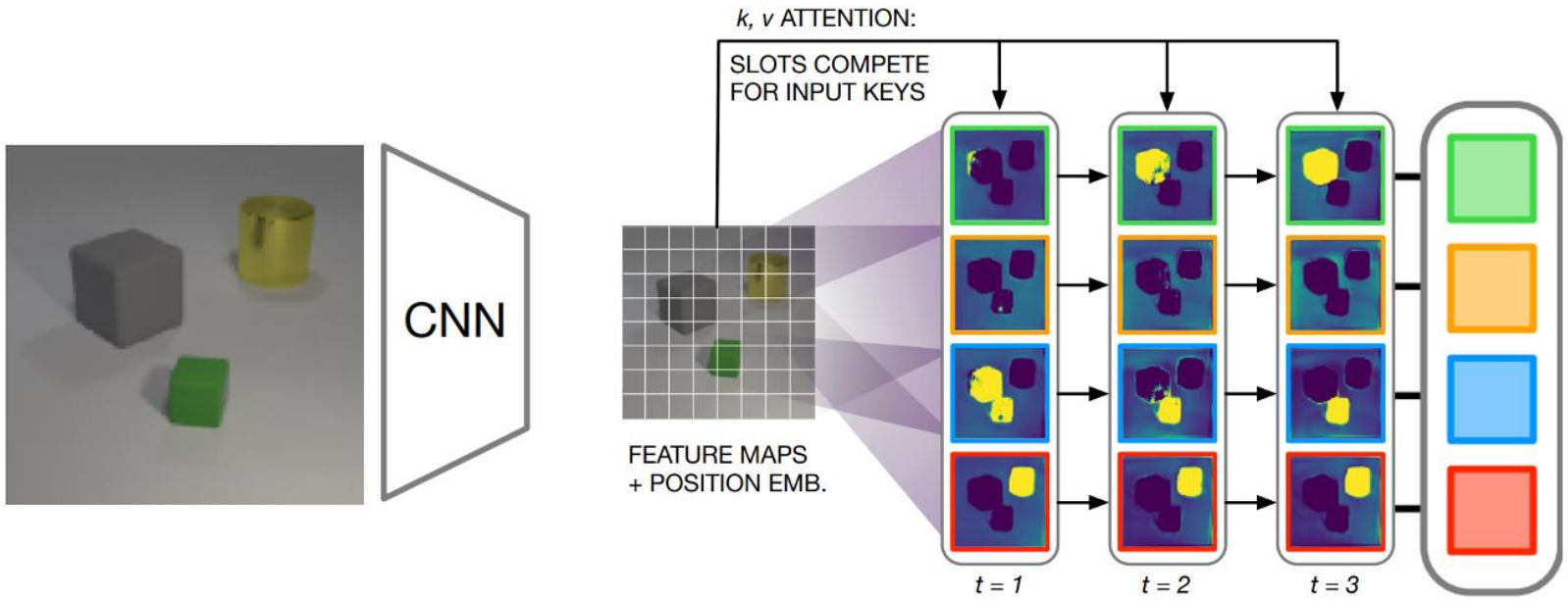
- Символьное описание сцены
- Логический вывод на основе объектно-центричных представлений
- Улучшение обобщающей способности алгоритмов
- Моделирование взаимодействие объектов между собой

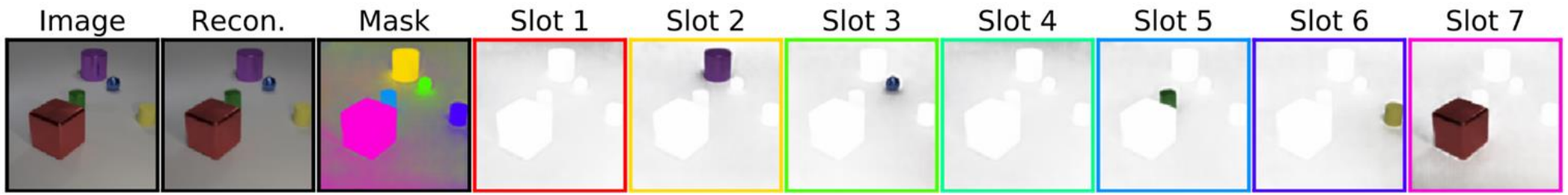




Adam Santoro el al. A simple neural network module for relational reasoning, 2017
Greff Klaus et al. On the Binding Problem in Artificial Neural Networks, 2020

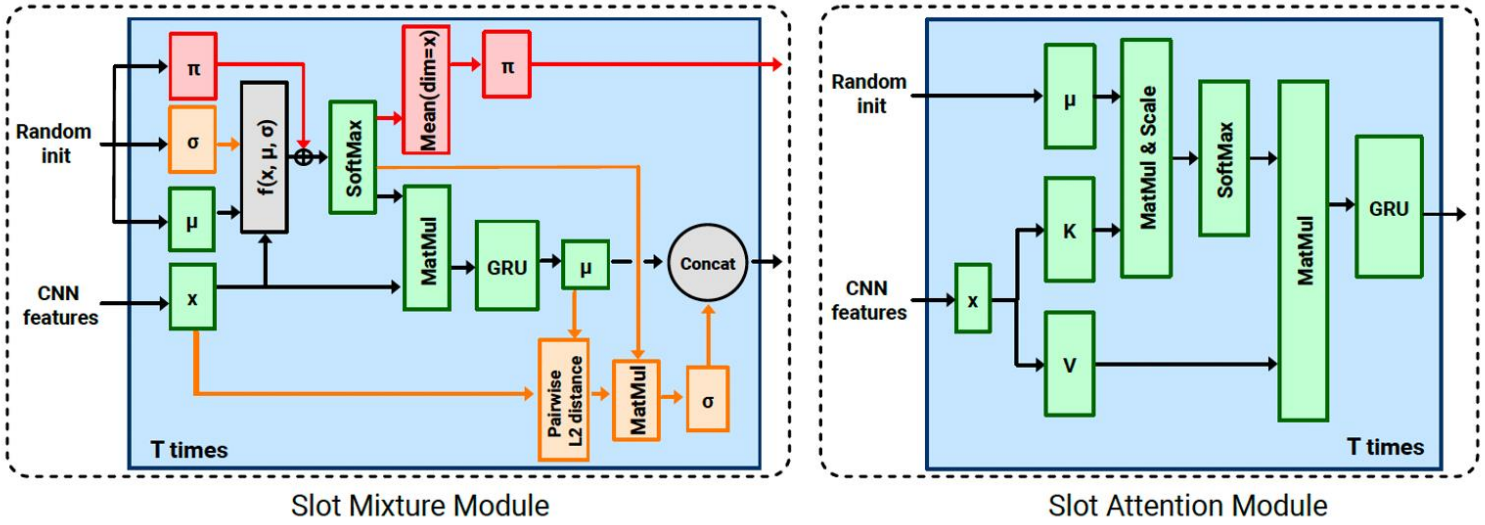
Slot Attention – итеративное выделение слотов





Francesco Locatello et al. Object-Centric Learning with Slot Attention, 2020

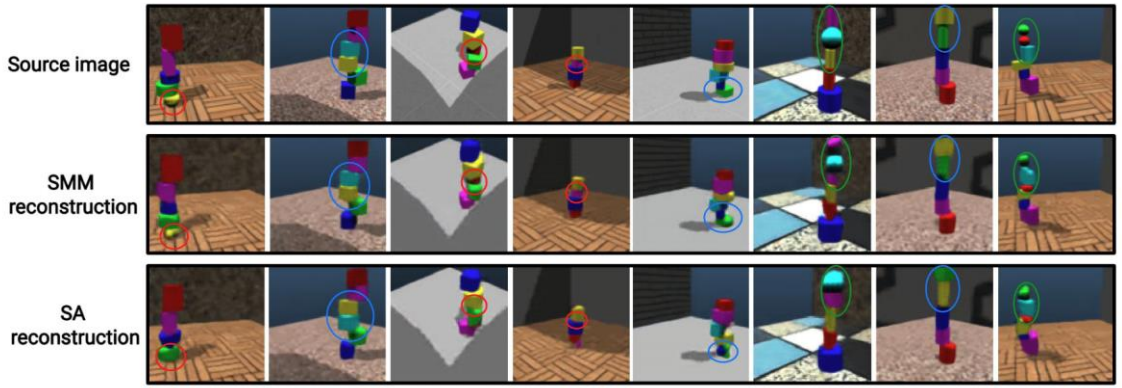
SMM – модель смеси слотов



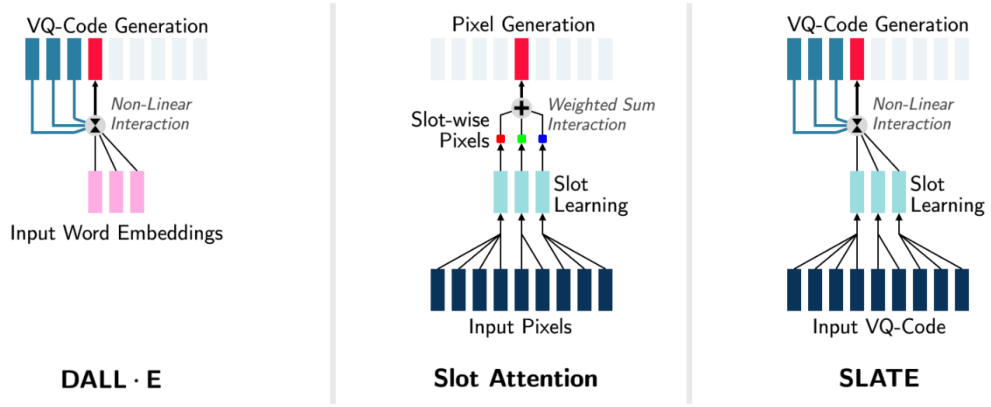
Улучшение модели Slot Attention:

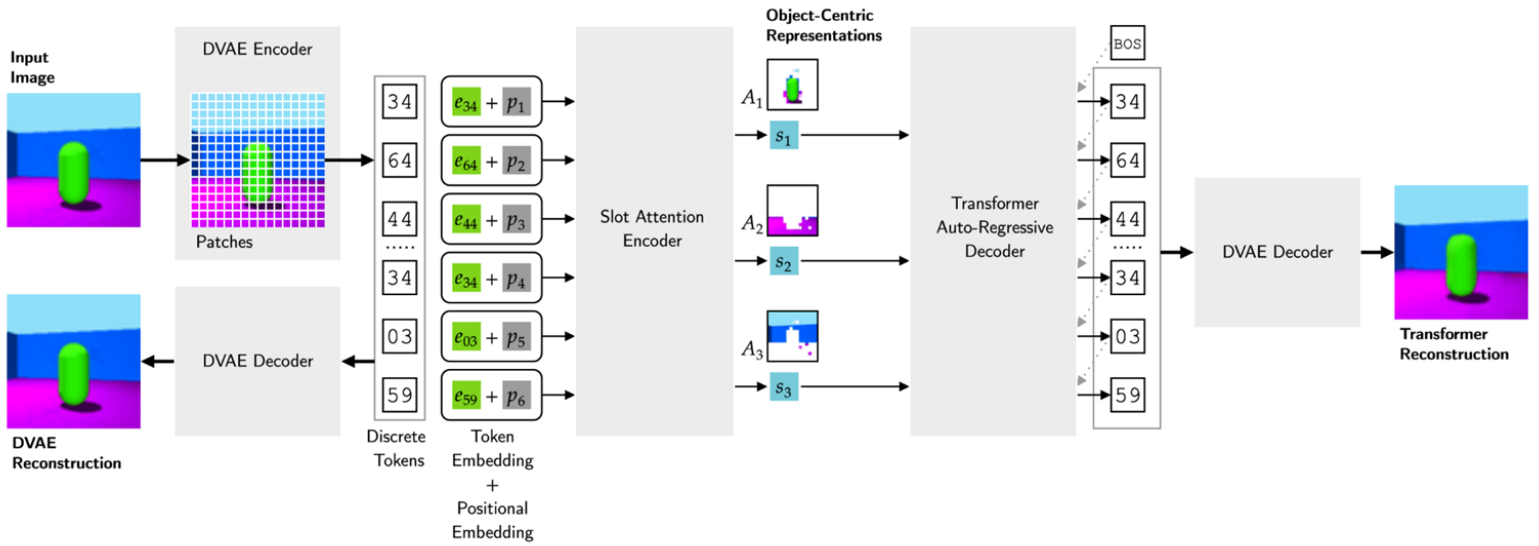
- → представление слота в виде гауссианы
- → итеративное разделение гауссиан EM-алгоритмом

Model	AP_{∞}	AP_1	$AP_{0.5}$	$AP_{0.25}$	$AP_{0.125}$	$AP_{0.0625}$
SA	94.3 ± 1.1	86.7 ± 1.4	$56.0 \pm 3.$	10.8 ± 1.7	0.9 ± 0.2	–
SA*	97.1 ± 0.7	94.5 ± 0.7	88.3 ± 3.2	62.5 ± 5.4	23.6 ± 1.4	4.6 ± 0.3
SA-MESH	99.4 ± 0.1	99.2 ± 0.2	98.9 ± 0.2	91.1 ± 1.1	47.6 ± 0.8	12.5 ± 0.4
IDSPN	98.8 ± 0.5	98.5 ± 0.6	98.2 ± 0.6	95.8 ± 0.7	76.9 ± 2.5	32.3 ± 3.9
SMM (ours)	99.4 ± 0.2	99.3 ± 0.2	98.8 ± 0.4	$\mathbf{98.4 \pm 0.7}$	$\mathbf{92.1 \pm 1.2}$	$\mathbf{47.3 \pm 2.5}$



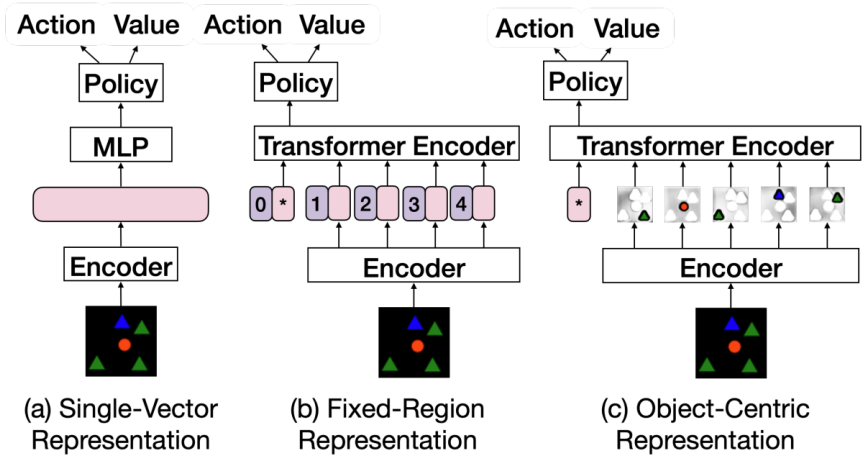
SLATE (SLot Attention TransformEr)

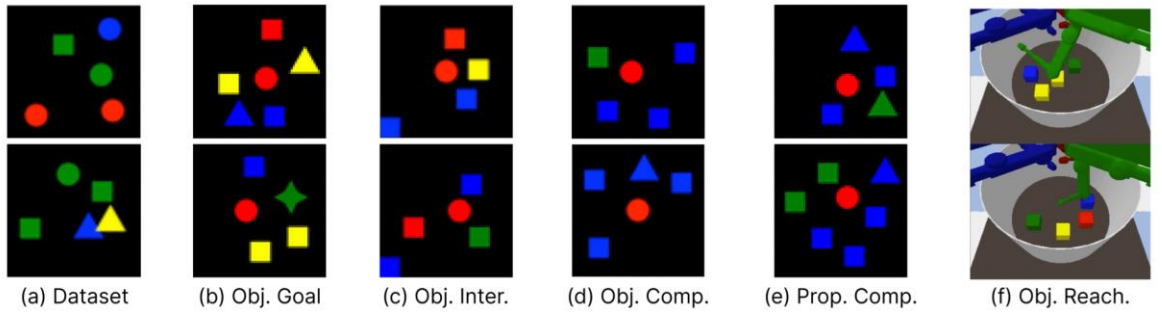


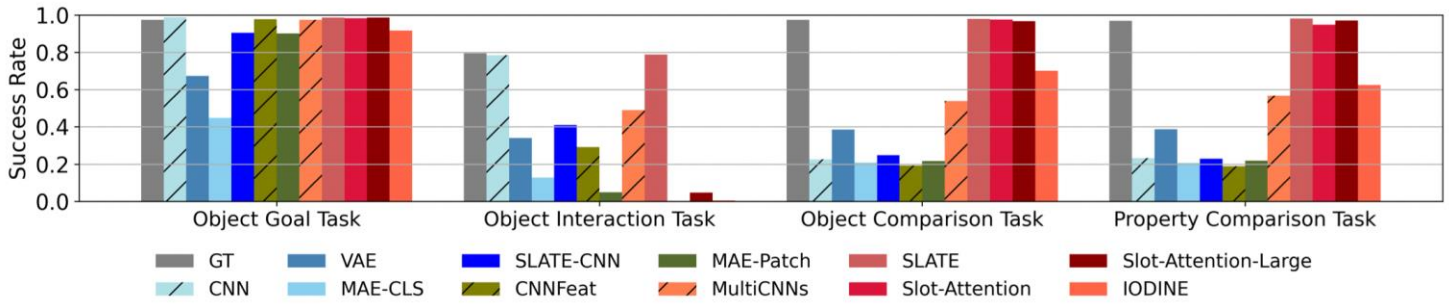


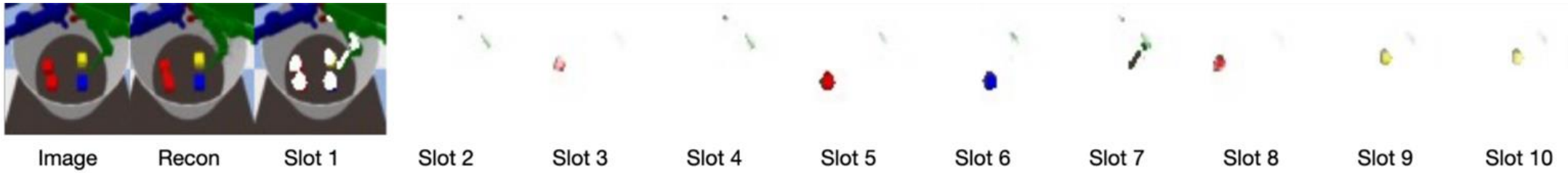
Сравнение объектно-центрированных кодировщиков

- Алгоритм PPO
- Задачи в 2D и 3D средах



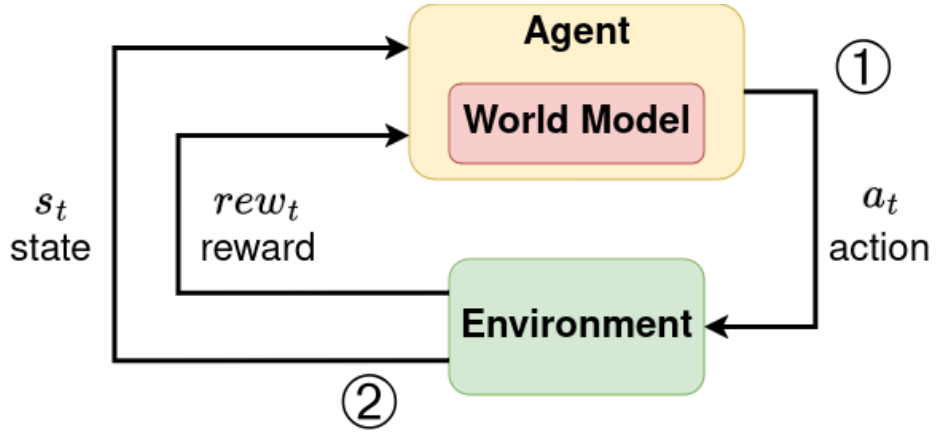


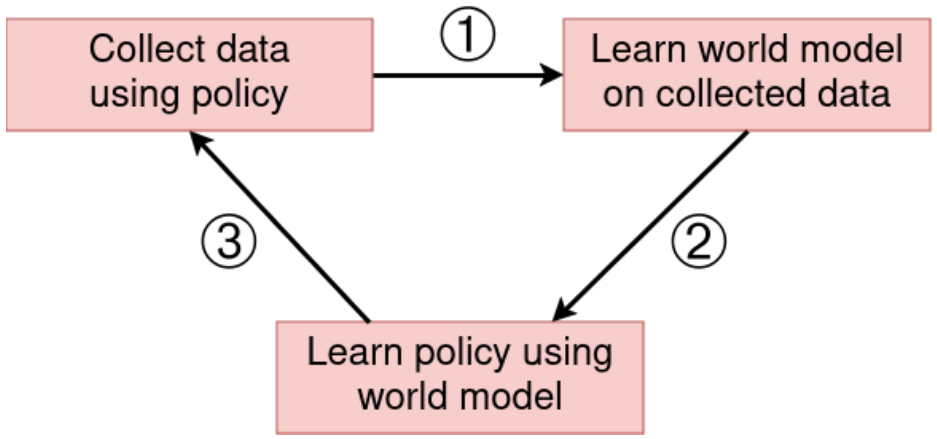




Jaeski Yoon et al. An Investigation into Pre-Training Object-Centric Representations for Reinforcement Learning, 2023

Цикл сбора данных в обучении с моделью





МППР: $(\mathcal{S}, \mathcal{A}, P, r, \rho, \gamma)$

Пространство состояний: $\mathcal{S}$

Пространство действий: $\mathcal{A}$

Начальное распределение: ρ

Дисконт фактор: γ

Функция переходов: $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$

Функция вознаграждений: $r : \mathcal{S} \to \mathbb{R}$

Стратегия: $\pi : \mathcal{S} \times \mathcal{A} \to [0; 1]$

Отдача: $G(\pi) = \mathbb{E}_{s_0, a_0, s_1, a_1, \ldots, s_T} \left[\sum_{t=0}^{T} \gamma^t r(s_t) \right],$

где $s_0 \sim \rho(\cdot), a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$

Цель: $\pi^* = \arg\max G$

AIRI

Обновление модели среды

Формально задачу обучения с подкреплением на основе модели можно сформулировать как следующую оптимизационную задачу

$$\begin{cases} \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^i R(s_t, a_t)\right], \\ \arg\min_{\tilde{R}} \mathcal{L}_R(R, \tilde{R}), \\ \arg\min_{\tilde{T}} \mathcal{L}_T(T, \tilde{T}), \end{cases}$$

В качестве функций потерь для определения компонентов модели $\mathcal{L}_R$ и $\mathcal{L}_T$ обычно выступают среднеквадратичная ошибка и оценка максимума правдоподобия соответственно:

$$\mathcal{L}_R\left(R, \tilde{R}\right) = E_{(s,a)\sim\mathcal{D}}\left[\left(R\left(s,a\right) - \tilde{R}\left(s,a\right)\right)^2\right],$$

$$\mathcal{L}_T\left(T, \tilde{T}\right) = E_{(s,a)\sim\mathcal{D}}\left[D_{\mathrm{KL}}\left(T\left(\cdot|s,a\right) || \tilde{T}(\cdot|s,a)\right)\right]$$

Эквивалентные по полезности модели

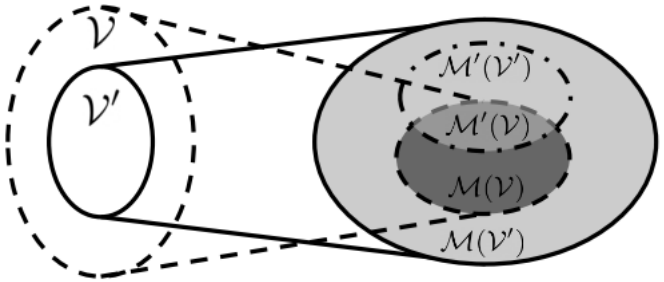
Каждая модель $M = < T, R >$ индуцирует оператор Беллмана:

$$\mathcal{B}_\pi[V](s) = E_{a\sim\pi, s'\sim T}[R(s,a) + \gamma V(s')]$$

Пусть $\mathcal{P}$ – множество стратегией, а $\mathcal{V}$ – множество некоторых функций (полезности)

Будем говорить, что модели $M = < T, R >$ и $\tilde{M} = < \tilde{T}, \tilde{R} >$ эквивалентны по полезности относительно $\mathcal{P}$ и $\mathcal{V}$, когда выполняется

$$\mathcal{B}_\pi[V] = \tilde{\mathcal{B}}_\pi[V], \ \forall \pi \in \mathcal{P}, V \in \mathcal{V}$$



Grimm, C., Barreto, A., Singh, S., Silver, D.: The value equivalence principle for model-based reinforcement learning. In: NeurIPS 2020

AIRI

Пространство моделей, эквивалентных по полезности

В обучении на основе модели целью агента является в том числе и поиск оптимальной модели $\mathrm{m}^* = < T, R >$

Агенту достаточно найти любую модель, которая эквивалентна по полезности оптимальной модели m^*, т. е. найти $m \in M_{m^*}(\mathcal{P}, \mathcal{V}) \equiv M(\mathcal{P}, \mathcal{V})$

Оптимизационная задача перепишется следующим образом:

$$\arg\min_{\tilde{m}} \sum_{\pi \in \mathcal{P}} \sum_{V \in \mathcal{V}} \left\| \mathcal{B}_\pi [V] - \tilde{\mathcal{B}}_\pi [V] \right\|$$

Соответствующая функция потерь уже явно зависит от выбора стратегии и соответствующей функции полезности

Основной трудностью в данном случае является определение пространств $\mathcal{P}$ и $\mathcal{V}$

Объектная ситуация

- Предполагаем, что в результате анализа ситуации на тактическом уровне, нам доступна функция выделения объектов из сенсорных данных $\Phi: S \to O$
- Пусть определено множество классов объектов $\mathcal{C} = \{c_1, \dots, c_n\}$
- Каждый класс имеет набор атрибутов $\{c.p_1, \dots, c.p_m\}$
- Каждое состояния s состоит из объектов $f(s) = \{o_1, \dots, o_k\}$, где каждый объект относится к какому-либо классу $c(o) \in \mathcal{C}$
- Состояние объекта определяется значениями атрибутов соответствующего класса
- Состояние s является объединением состояний всех объектов $\bigcup_{i=0}^{k} o_i$
- Для простейшего учета связывания объектов вводится индикаторная функция взаимодействия $I: O \times O \to \{0,1\}$, которая приводит к декомпозиции модели переходов: $T_c(o, a)$ или $T_{c_i,c_j}(o_i, o_j, a)$, если $I(o_i, o_j) = 1$

Diuk, Carlso, Andre Cohen, and Michael L. Littman. An object-oriented representation for efficient reinforcement learning. ICML. 2008.
Keramati, Ramtin, Jay Whang, Patrick Cho, and Emma Brunskill. Strategic Object Oriented Reinforcement Learning. CoRR. 2018.

AIRI

Иерархическая стратегия в объектном случае

Объектная версия оператора Беллмана:

$$\mathcal{B}_\pi [V](e) = E_{a(e)\sim\pi, s'\sim T}\left[R(e,a) + \gamma V\left(e'\right)\middle| e' \in s'\right].$$

Выполнение действий агента становится двухэтапным:

- вначале агент должен определить объект $e \in s$, с которым он будет взаимодействовать,
- а затем уже выбрать действие из множества доступных действий A_c, определенных для класса данного объекта $c(e)$

Получаем иерархический вариант обучения с подкреплением, для которого обычно используется формулировка абстрактных действий в виде умений

Пусть $\kappa(c) =< I_c, \pi_c, \beta_c >$ - длящееся во времени действие, или умение, для которого

- $I_c \subseteq S$ - инициирующее множество состояний, такое, что $\forall s \in S, e \in s, c(e) = c$,
- π_c - стратегия, реализующая данное умение, такая, что $\forall a \sim \pi_c, a \in A_c, c(e) = c$,
- $\beta_c : S \to \{0,1\}$ - это терминальная функция, завершающая данное умение

Иерархическая постановка задачи

Иерархический вариант уравнения Беллмана:

$$\mathcal{B}_{\pi_\kappa}[V](s) = E_{\kappa(c)\sim\pi_\kappa, s'\sim T}\left[R(s,\kappa(c)) + \gamma^\tau V\left(s'\right)\middle| s \in I_c(\kappa)\right]$$

Объектно-центричная оптимизационная задача:

$$\begin{cases} \operatorname{argmax}_{\pi_\kappa} E\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,\kappa_t)\right], \\ \operatorname{argmin}_{\tilde{m}} \sum_{\pi_\kappa\in\mathcal{P}_\kappa}\sum_{V\in\mathcal{V}_\kappa}\left\|\mathcal{B}_{\pi_\kappa}[V] - \tilde{\mathcal{B}}_{\pi_\kappa}[V]\right\|, \\ \operatorname{argmin}_{\tilde{m}_{c_1}} \sum_{\pi_{c_1}\in\mathcal{P}_{c_1}}\sum_{V\in\mathcal{V}_{c_1}}\left\|\mathcal{B}_{\pi_{c_1}}[V] - \tilde{\mathcal{B}}_{\pi_{c_1}}[V]\right\|, \\ \dots \\ \operatorname{argmin}_{\tilde{m}_{c_k}} \sum_{\pi_{c_k}\in\mathcal{P}_{c_k}}\sum_{V\in\mathcal{V}_{c_k}}\left\|\mathcal{B}_{\pi_{c_k}}[V] - \tilde{\mathcal{B}}_{\pi_{c_k}}[V]\right\| \end{cases}$$

$k+1$ подзадачи минимизации ошибки могут решаться параллельно с использованием одной и той же памяти прецедентов или траекторий $\mathcal{D}$, для каждого состояний которой применена функция выделения объектов Φ

AIRI

Применимость декомпозиции задачи

Полученная декомпозиция справедлива только в том случае, когда существует локальность эффектов каждого действия

- → Это означает, что изменение свойств одних объектов точно определяет изменение свойств других
- → Это достаточно частный, но важный случай сред

В качестве практической реализации подсчета близости модели к оптимальной эффективно использовать эмпирическую версию расстояния между предсказаниями полезности для всех состояний, которые встречаются для некоторой сгенерированной выборки:

$$\mathcal{L}\left(m^*, \tilde{m}\right) = \sum_{\pi \in \mathcal{P}} \sum_{V \in \mathcal{V}} \sum_{s \sim \mathcal{D}} \left[\frac{\sum_{\mathcal{D}, s_i = s} R(s_i) + \gamma V(s_i')}{N_{\mathcal{D}}(s_i)} - \tilde{\mathcal{B}}_\pi \left[V\right](s) \right],$$

где $N_D(s_i)$ - количество раз, сколько состояние s_i встретилось в выборке $\mathcal{D}$

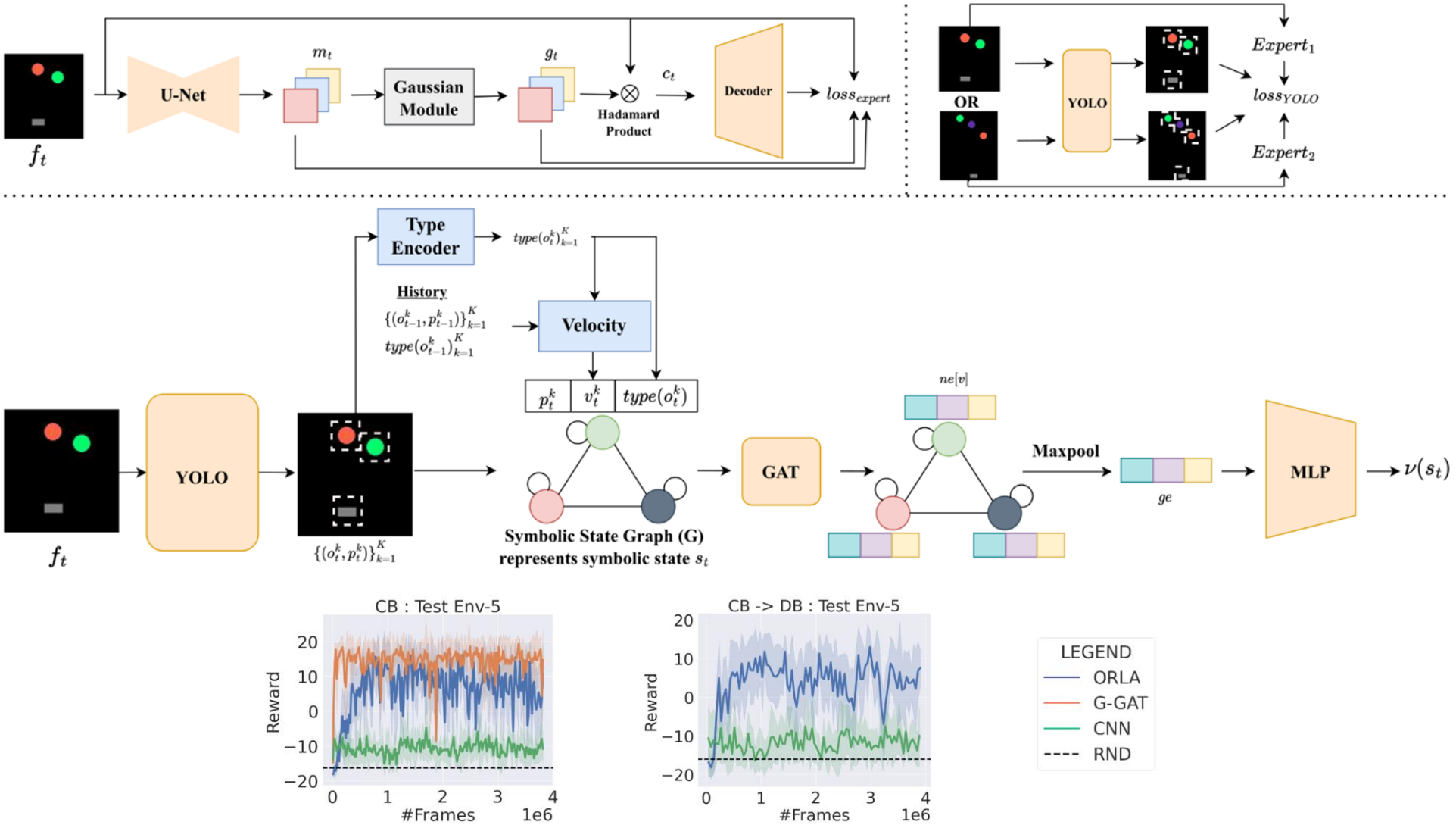
Алгоритм обучения когнитивного агента с объектной декомпозицией

1. Инициализация верхнеуровневой стратегии агента π, объектных умений π_{c_i}, $\tilde{m}$ и частных моделей $\tilde{m}_{c_i}$
2. Генерация дополнительного опыта агента $\mathcal{D}$, полученного в среде с использованием стратегий π и π_{c_i}
3. Обновление общей и частных моделей с использованием объектной оптимизационной задачи
4. Решение задачи планирования (определения субоптимальных функций полезности) для умений V^i и общей стратегии V^κ по общей и частными моделям
5. Использование жадных по полезностям V^i и V^κ стратегий в качестве новых высокоуровневой стратегии π и объектных умений π_{c_i}, переход к шагу 2

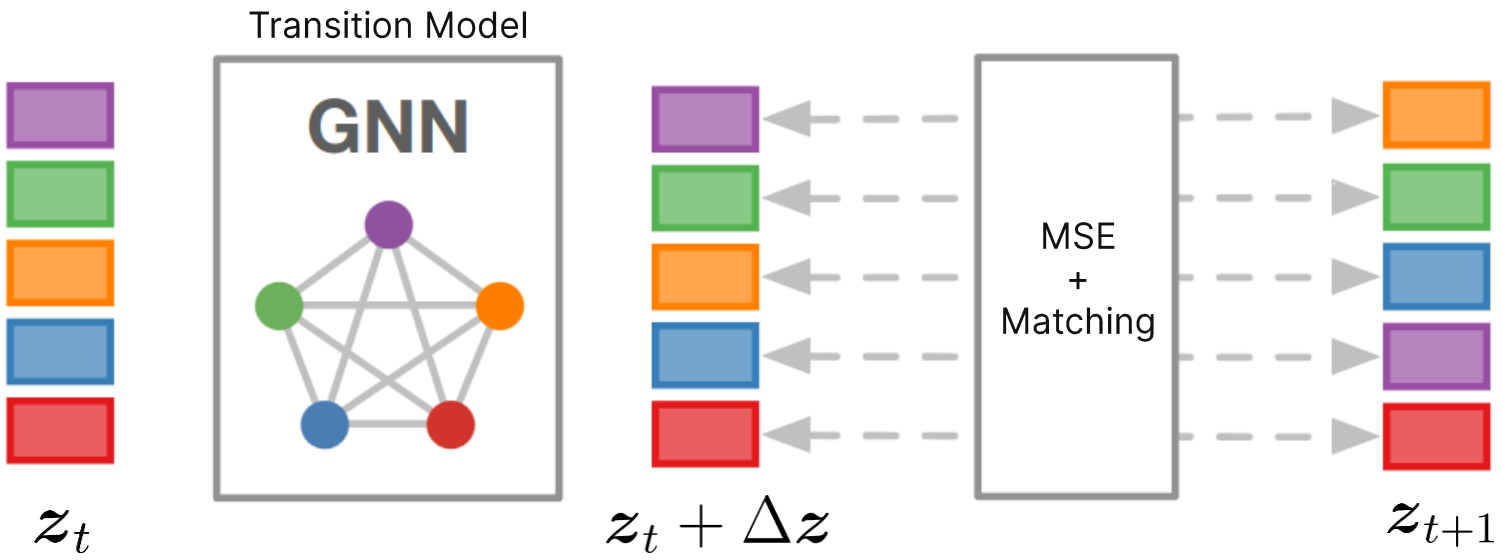
04

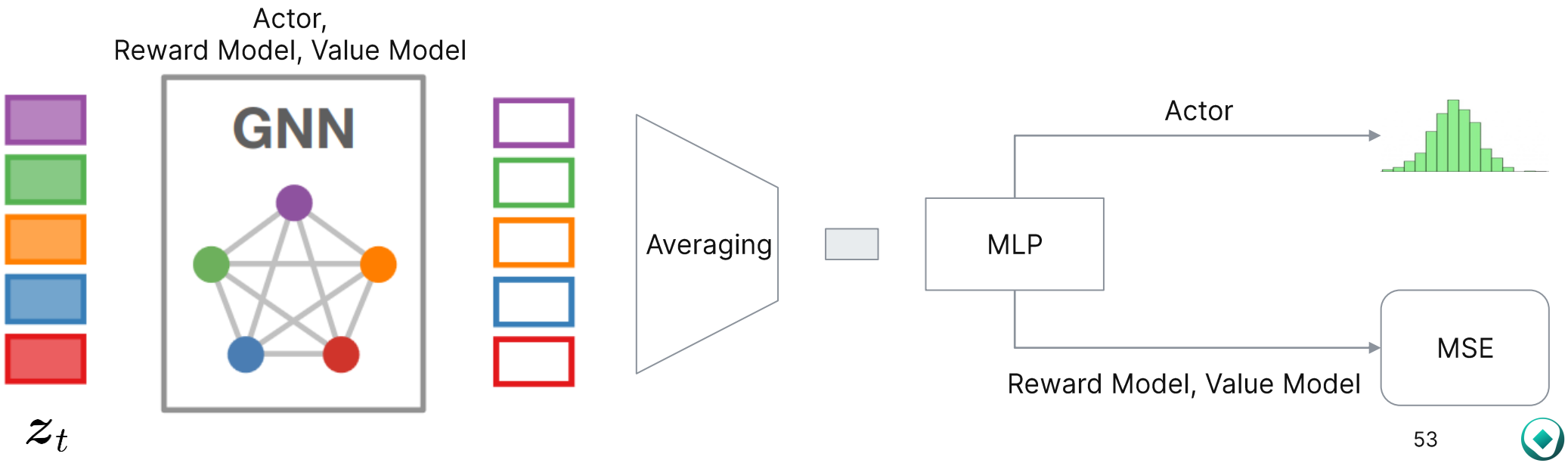
Объектно-центричное обучение с моделью среды

Подход на основе автокодировщика

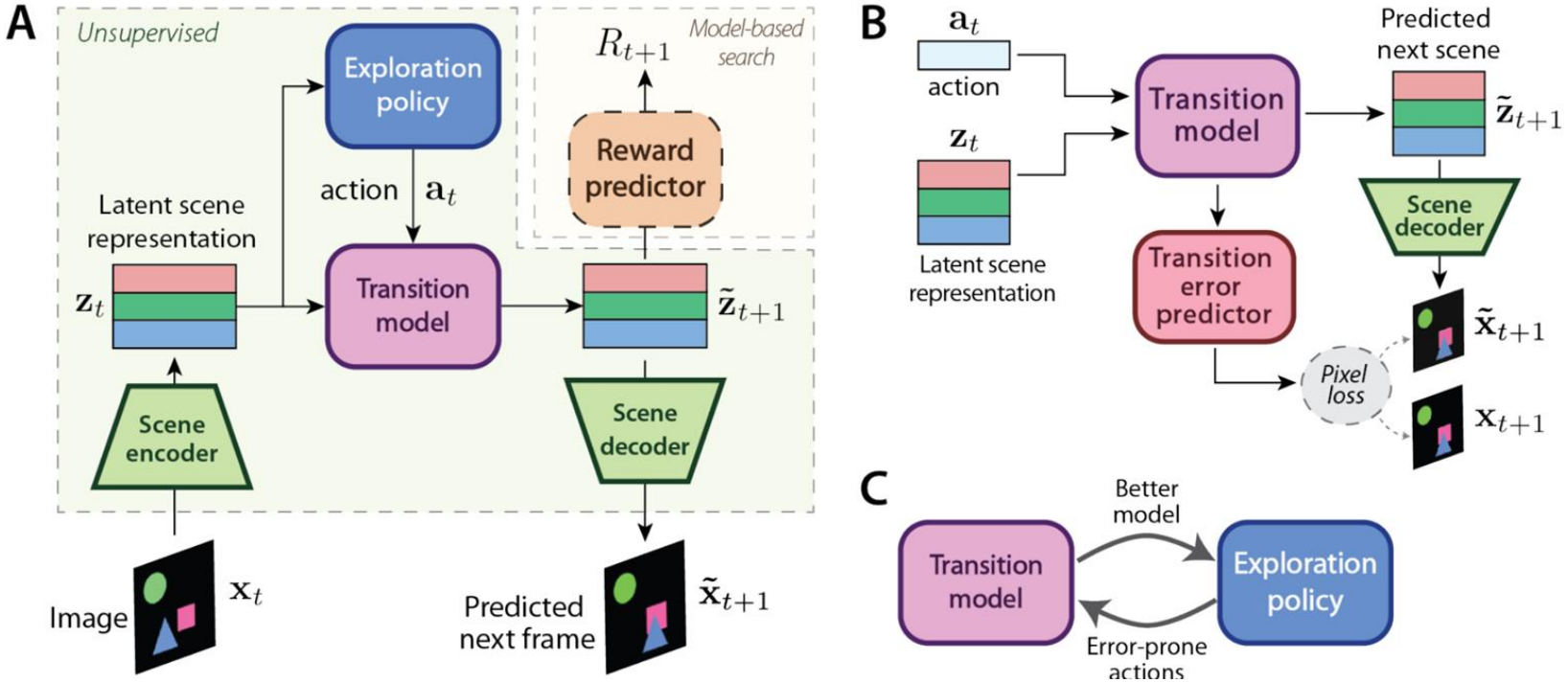


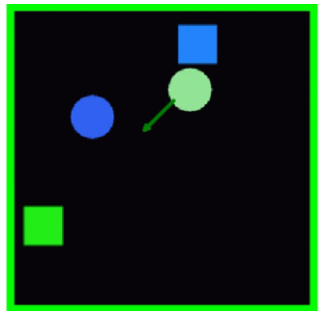
Графовые нейронные сети





COBRA





$$\tilde{z}_{t+1} = T(z_t, a_t) = [T_{net}(z_t^0, a_t), ..., T_{net}(z_t^K, a_t)]$$

$$Loss_T(z_t, a_t) = ||V_{dec}(T(z_t, a_t)) - x_{t+1}||^2$$

$$a_t^{expl} = D_\theta^{expl}(z_t, u), u \sim [0; 1]^4$$

$$Loss_{expl}(z_t) = |a_t^{expl}| - L_{error-pred}(z_t, a_t^{expl})$$

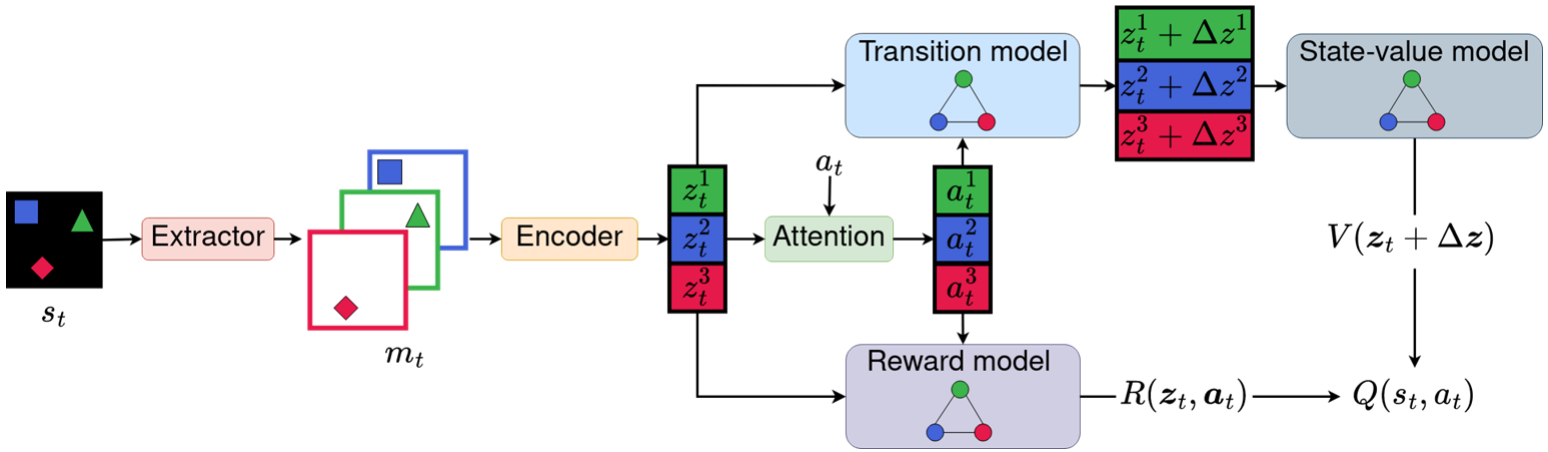
$$R(z_t) = GNN(z_t)$$

$$a_t = \underset{\{a_t^1, \cdots, a_t^N\} \sim D_\theta^{expl}}{\arg\max} R(T(z_t, a))$$

Nicholas Watters et al. COBRA: Data-Efficient Model-Based RL through Unsupervised Object Discovery and Curiosity-Driven Exploration, 2019

AIRI

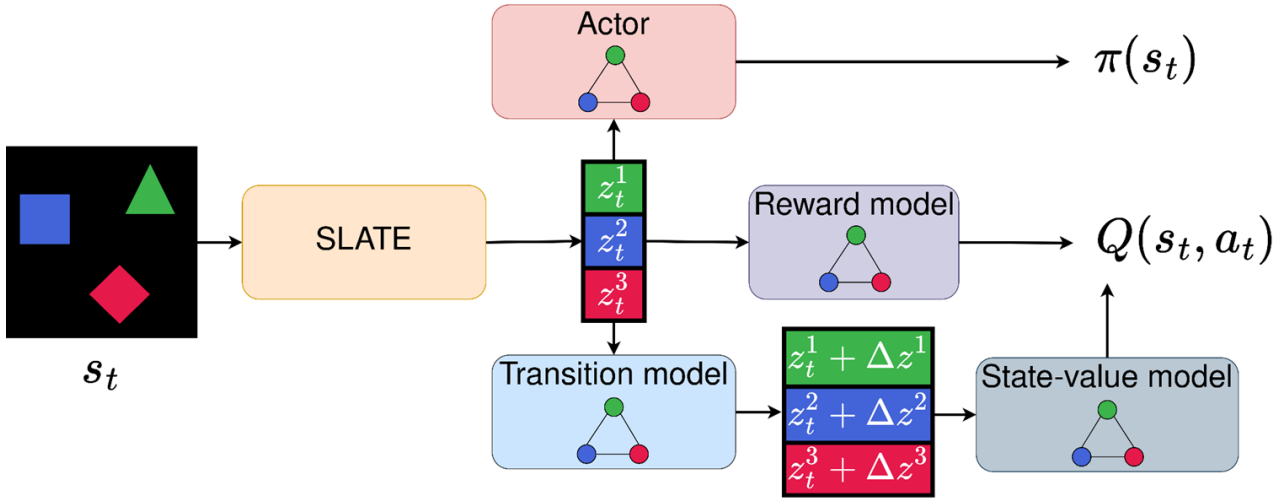
Объектно-центричная Q-сеть



- Основан на алгоритме TreeQN
- Использует CSWM-подобную графовую модель переходов
- Обучение с помощью контрастной функции потерь
- Падение качества экстрактора при наличии статических объектов
- Работа только с дискретным пространством действий
- Чередование фаз обучения моделей среды и полезности

Gregory Farquhar et al. TreeQN and ATreeC: Differentiable Tree-Structured Models for Deep Reinforcement Learning, 2019
Thomas Kipf et al. Contrastive Learning of Structured World Models, 2020

Графовый объектно-центрированный актор-критик



- GOCA основан на off-policy алгоритме Soft Actor-Critic
- Работает с дискретным и непрерывным пространством действий
- Извлечение объектных представлений с помощью SLATE
- Полезность действий предсказывается с помощью графовой модели среды

$$J_V(\theta) = \mathbb{E}_{(s_t,a_t)\sim D}\left[\left(R(\boldsymbol{z}_t, a_t) + \gamma V_\theta(\boldsymbol{z}_t + \Delta\boldsymbol{z}) - r_t - \gamma V_{\bar{\theta}}(\boldsymbol{z}_{t+1})\right)^2\right]$$

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim D}\left[\mathbb{E}_{a_t\sim\pi_\phi(\cdot|\boldsymbol{z}_t)}\left[\alpha\log(\pi_\phi(a_t|\boldsymbol{z}_t)) - (R(\boldsymbol{z}_t, a_t) + \gamma V_\theta(\boldsymbol{z}_t + \Delta\boldsymbol{z}))\right]\right]$$

$$J(\alpha) = \mathbb{E}_{a_t\sim\pi(\cdot|\boldsymbol{z}_t)}\left[-\alpha(\log\pi(a_t|\boldsymbol{z}_t) + \bar{H})\right]$$

$$J_{WM} = \mathbb{E}_{s_t,a_t,r_t,s_{t+1}\sim D}\left[\beta_T\|\boldsymbol{z}_t + \Delta\boldsymbol{z} - \boldsymbol{z}_{t+1}\|^2 + \beta_R\left(R(\boldsymbol{z}_t, a_t) - r_t\right)^2\right]$$

$$\boldsymbol{z}_t = SLATE(s_t)$$

AIRI

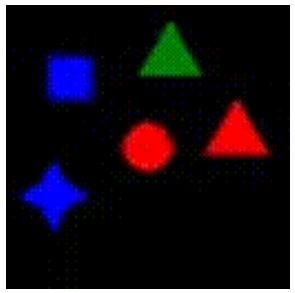
Среды

Navigation
5x5

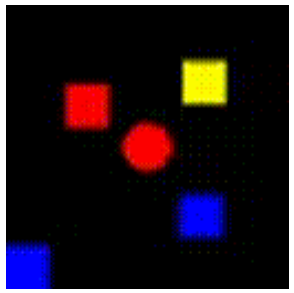
Navigation
10x10

PushingNoAgent
5x5

Pushing
7x7



Object Goal



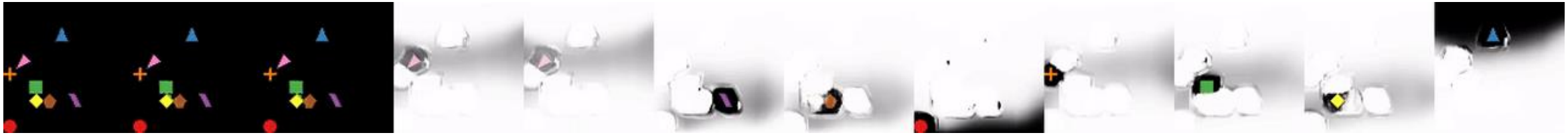
Object Interaction

Object Reaching

SLATE

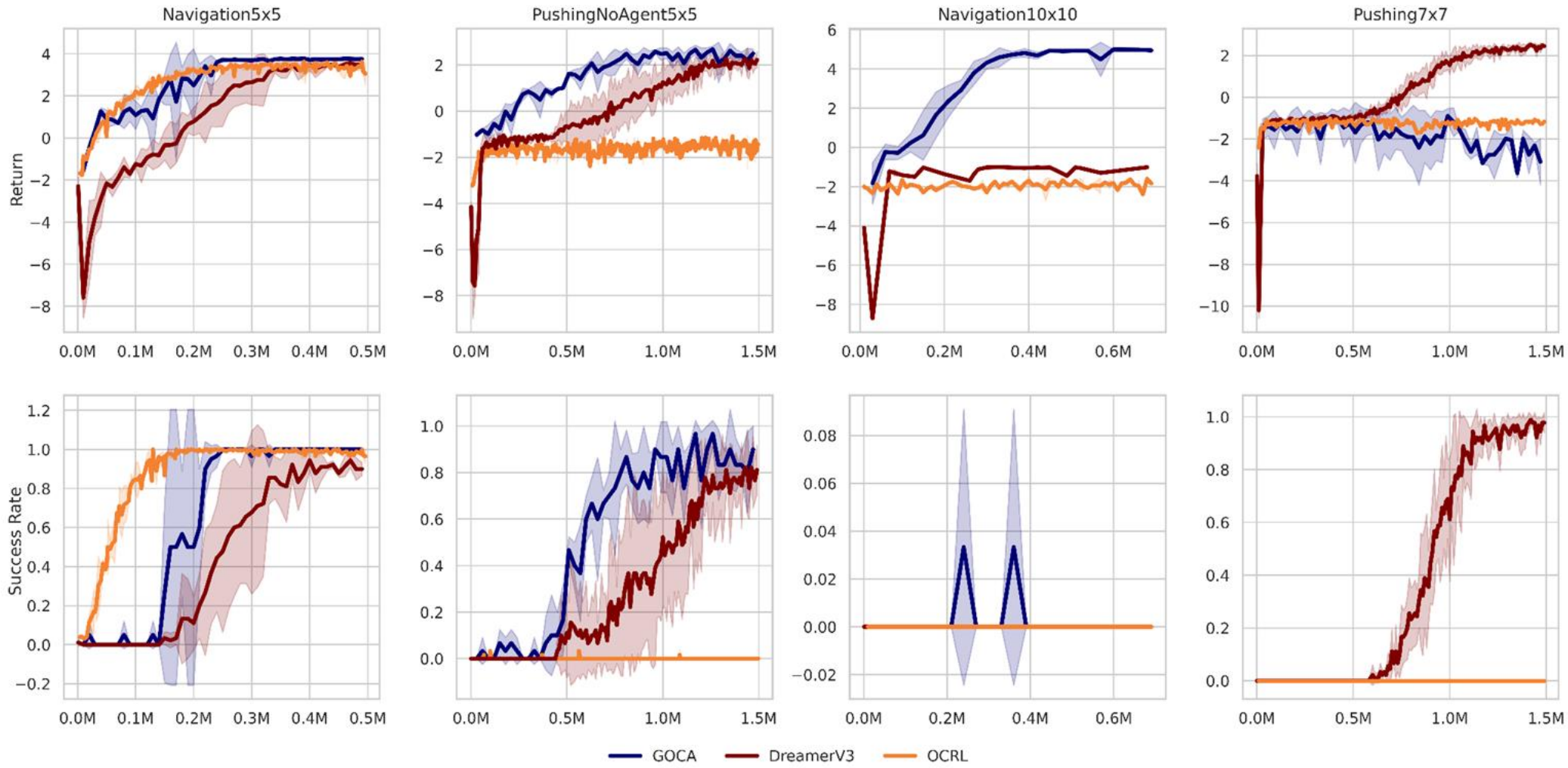
- 100K наблюдений, собранных равномерной случайной стратегией
- Предобучение SLATE в течение 150 эпох
- Преинициализация слотов значениями с предыдущего состояния для сопоставления слотов:

$$\boldsymbol{z}_{t+1} = SLATE(s_{t+1}|prev\_slots = \boldsymbol{z}_t)$$

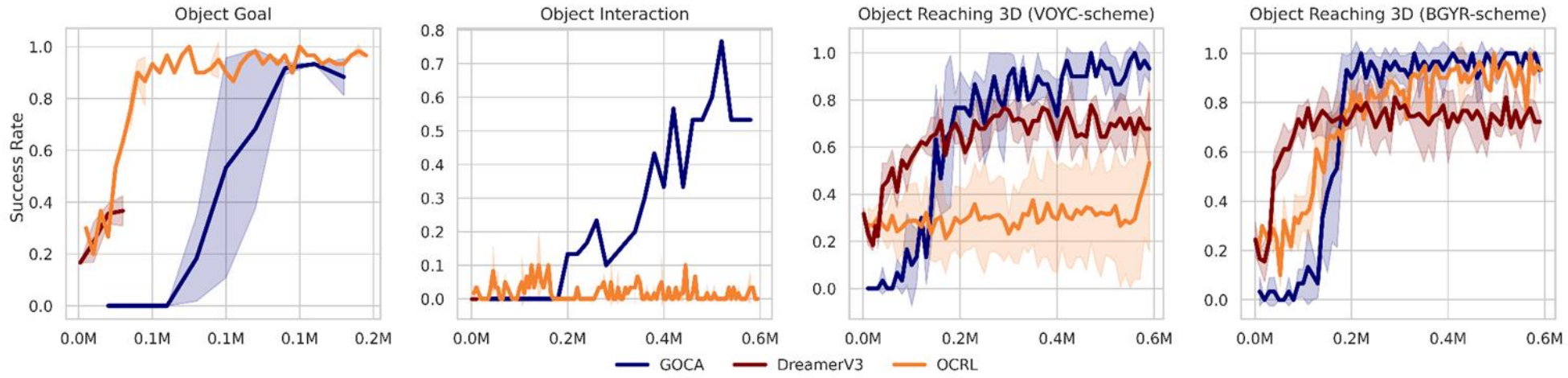


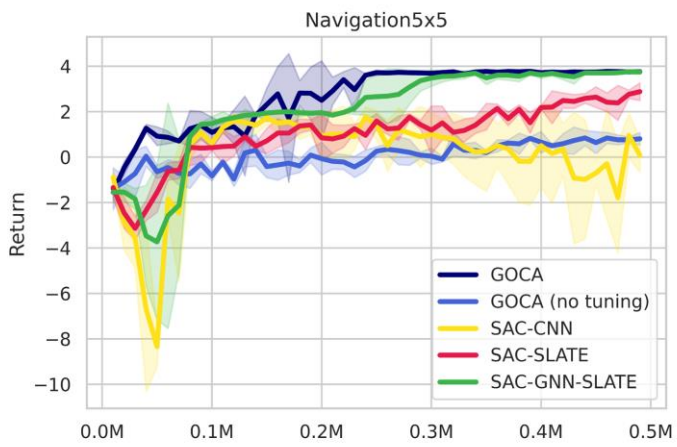
AIRI

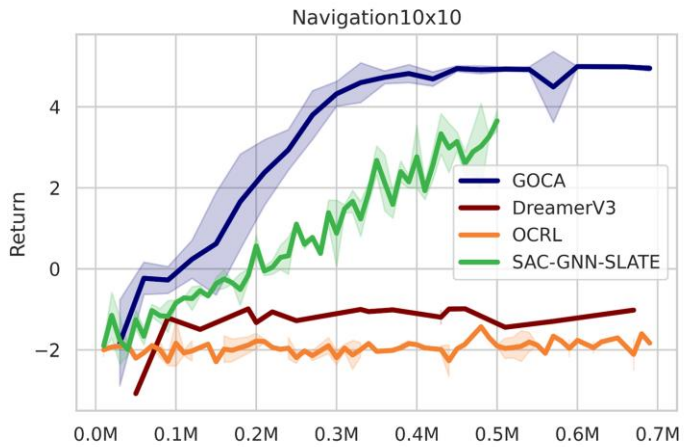
Результаты

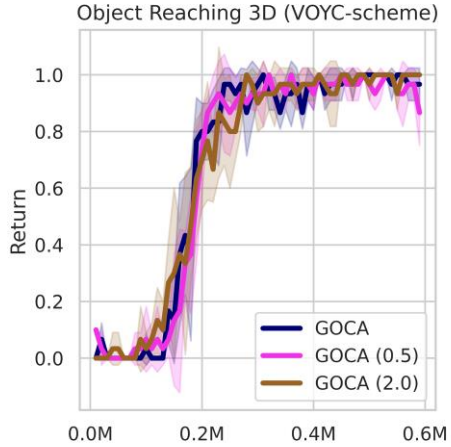

Navigation5x5
PushingNoAgent5x5
Navigation10x10
Pushing7x7
Return
Success Rate
0.0M
0.1M
0.2M
0.3M
0.4M
0.5M
0.5M
1.0M
1.5M
0.6M
GOCA
DreamerV3
OCRL

Результаты

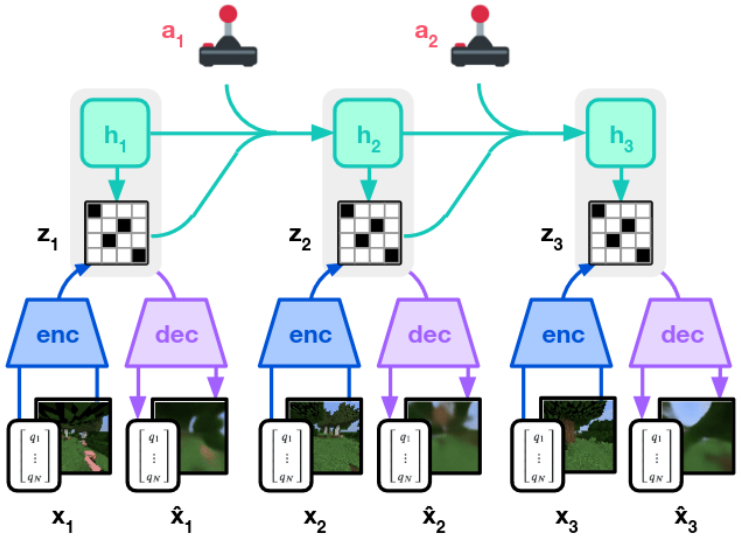

Object Goal
Object Interaction
Object Reaching 3D (VOYC-scheme)
Object Reaching 3D (BGYR-scheme)
Success Rate
GOCA
DreamerV3
OCRL


Navigation5x5
Return
GOCA
GOCA (no tuning)
SAC-CNN
SAC-SLATE
SAC-GNN-SLATE

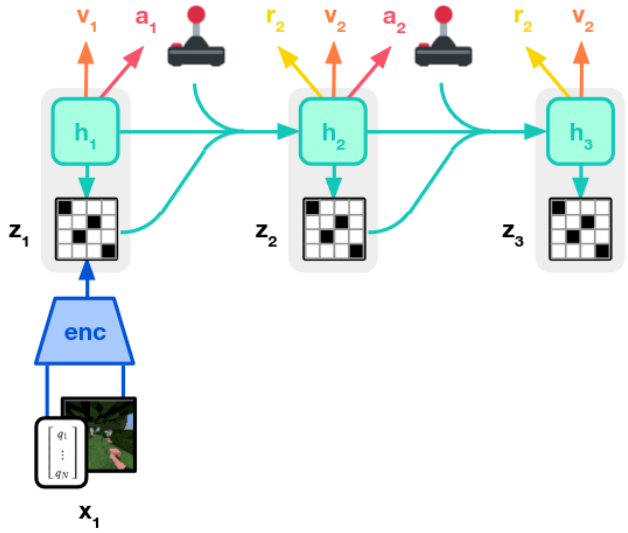

Navigation10x10
Return
GOCA
DreamerV3
OCRL
SAC-GNN-SLATE


Object Reaching 3D (VOYC-scheme)
Return
GOCA
GOCA (0.5)
GOCA (2.0)

DreamerV3: рекуррентная модель мира



(a) World Model Learning



(b) Actor Critic Learning

Модель пос-ти: $h_t = f_{GRU}(h_{t-1}, z_{t-1}, a_{t-1})$

Кодировщик: $z_t \sim q_\phi(z_t|h_t, x_t)$

Предсказание динамики: $\hat{z}_t \sim p_\phi(\hat{z}_t|h_t)$

Предсказание воз: $\hat{rew}_t \sim p_\phi(\hat{rew}_t|h_t, z_t)$

Предсказание зав: $\hat{cont}_t \sim p_\phi(\hat{cont}_t|h_t, z_t)$

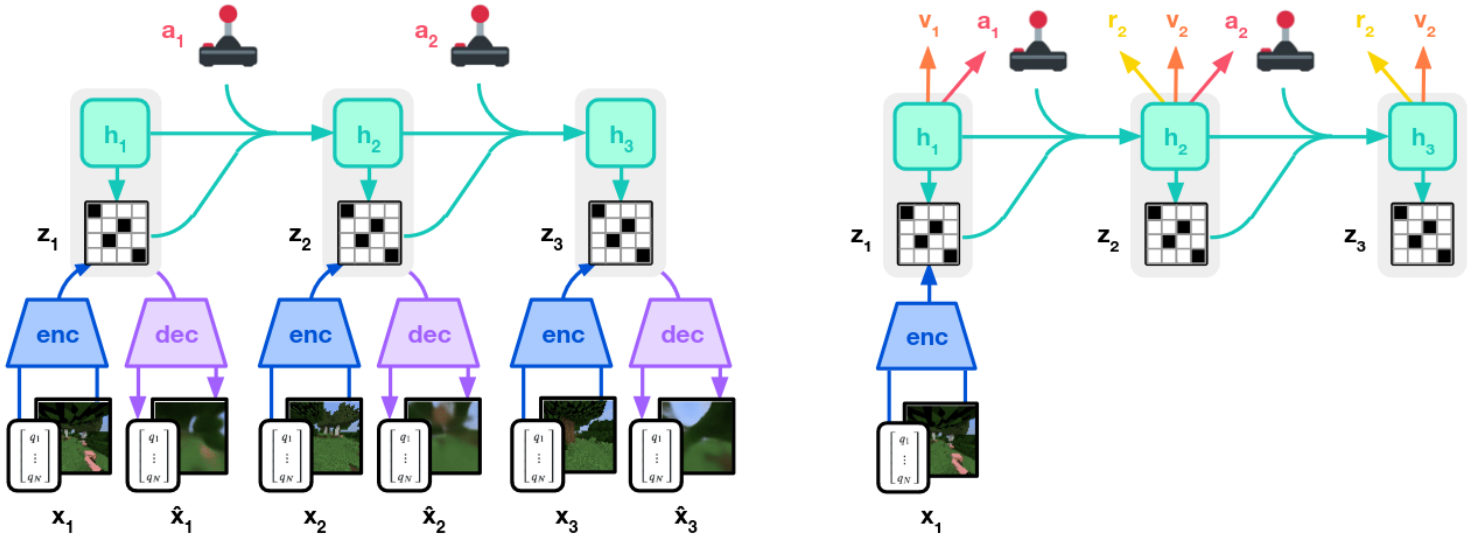
Декодер: $\hat{x}_t \sim p_\phi(\hat{x}_t|h_t, z_t)$

Рекуррентная модель в пространстве состояний (RSSM)

Danijar Hafner et al. Mastering Diverse Domains through World Models, 2023

AIRI

DreamerV3: обучение модели мира



(a) World Model Learning **(b)** Actor Critic Learning

Функции потерь:

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi}\left[\sum_{t=1}^{T}(\beta_{pred}\mathcal{L}_{pred}(\phi) + \beta_{dyn}\mathcal{L}_{dyn}(\phi) + \beta_{rep}\mathcal{L}_{rep}(\phi))\right]$$

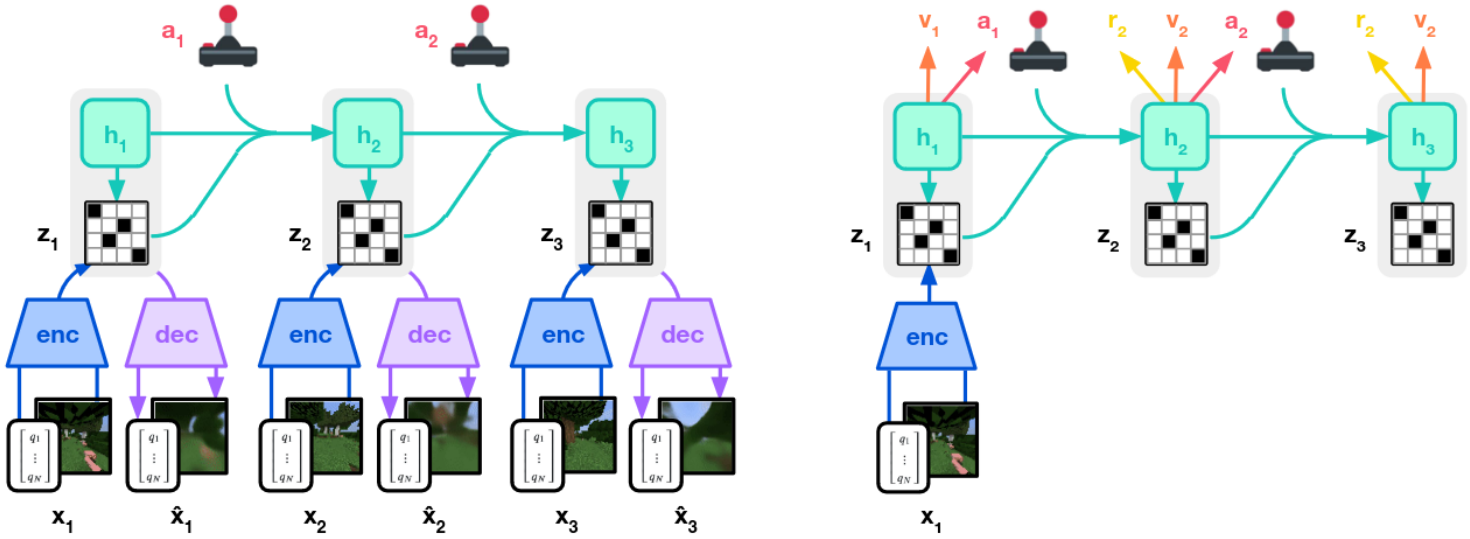
$$\mathcal{L}_{pred}(\phi) = -\ln p_\phi(x_t|h_t, z_t) - \ln p_\phi(rew_t|h_t, z_t) - \ln p_\phi(cont_t|h_t, z_t)$$

$$\mathcal{L}_{dyn}(\phi) = \max\left(1, \mathrm{KL}\left[\mathrm{sg}(q_\phi(z_t|h_t, x_t)) \parallel p_\phi(z_t|h_t)\right]\right)$$

$$\mathcal{L}_{rep}(\phi) = \max\left(1, \mathrm{KL}\left[q_\phi(z_t|h_t, x_t) \parallel \mathrm{sg}(p_\phi(z_t|h_t))\right]\right)$$

AIRI

DreamerV3: обучение актора-критика



(a) World Model Learning (b) Actor Critic Learning

Критик:

$$v_t \sim p_\psi(R_t|h_t, z_t)$$

$$R_t^\lambda = \hat{rew}_t + \gamma \hat{cont}_t \left[(1-\lambda)v_t + \lambda R_{t+1}^\lambda\right]$$

$$\mathcal{L}(\psi) = -\sum_{t=1}^{T} \ln p_\psi(R_t^\lambda|h_t, z_t)$$

Actor:

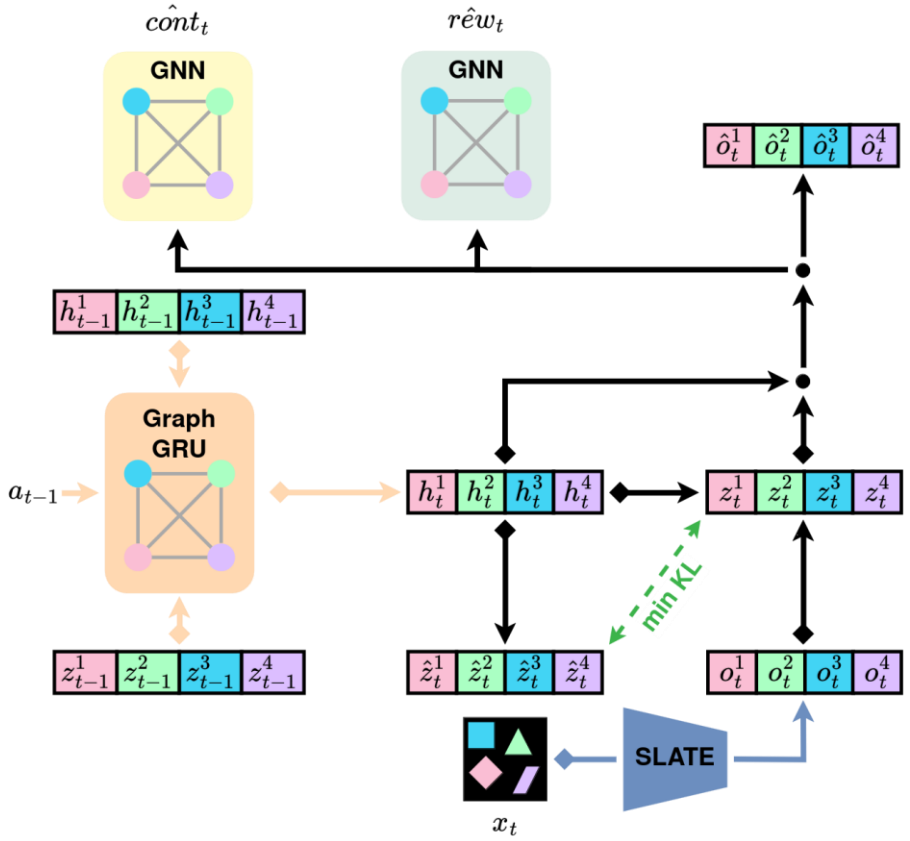
$$a_t \sim p_\theta(a_t|h_t, z_t)$$

$$S = \text{EMA}\left[\text{Per}(R_t^\lambda, 95) - \text{Per}(R_t^\lambda, 5), 0.99\right]$$

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \text{sg}\left[(R_t^\lambda - v_t)/\max(1, S)\right] \ln p_\theta(a_t|h_t, z_t) + \eta \text{H}\left[p_\theta(a_t|h_t, z_t)\right]$$

AIRI

OC-Dreamer: модель мира



$$\overline{h}_t = \{h_t^1, \ldots, h_t^K\} \qquad \overline{z}_t = \{z_t^1, \ldots, z_t^K\} \qquad \overline{o}_t = \{o_t^1 \ldots o_t^K\}$$

Модель пос-ти: $\overline{h}_t = f_{GGRU}(\overline{h}_{t-1}, \overline{z}_{t-1}, a_{t-1})$

Экстрактор объектов: $\overline{o}_t = \mathrm{SLATE}(x_t)$

Кодировщик: $z_t^i \sim q_\phi(z_t^i | h_t^i, o_t^i)$

Предсказание дин: $\hat{z}_t^i \sim p_\phi(\hat{z}_t^i | h_t^i)$

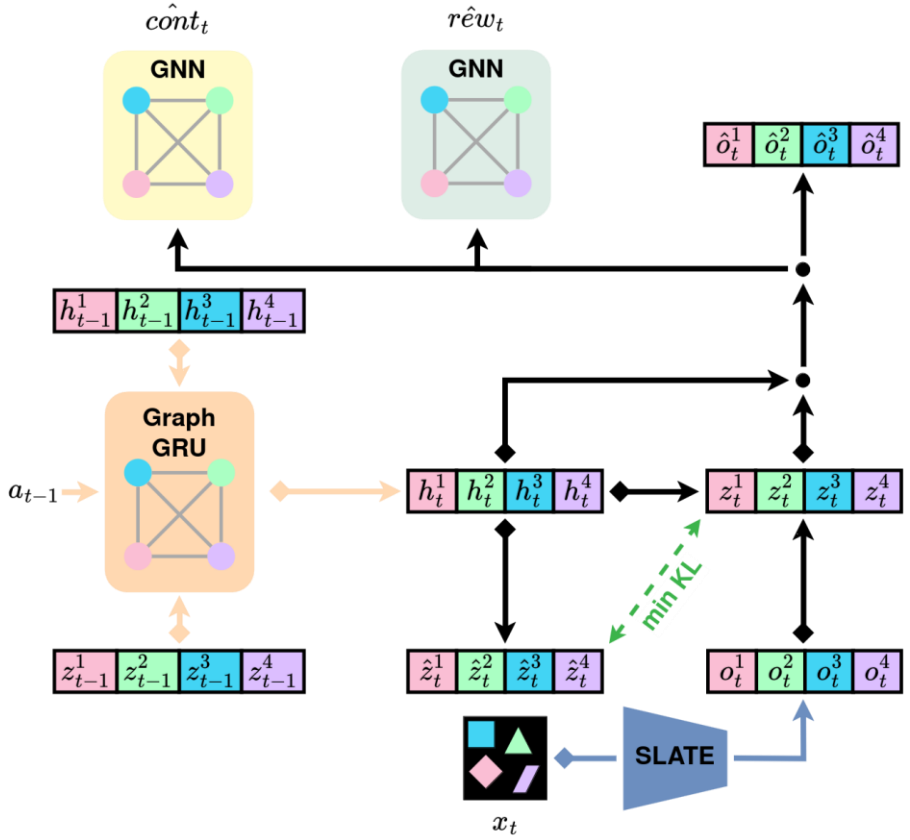
Предсказание возн: $\hat{rew}_t \sim p_\phi(\hat{rew}_t | \overline{h}_t, \overline{z}_t)$

Предсказание зав: $\hat{cont}_t \sim p_\phi(\hat{cont}_t | \overline{h}_t, \overline{z}_t)$

Декодеровщик: $\hat{o}_t^i \sim p_\phi(\hat{o}_t^i | h_t^i, z_t^i)$

Общая MLP

OC-Dreamer: объектная RSSM



$$\overline{h}_t = \{h_t^1, \ldots, h_t^K\} \qquad \overline{z}_t = \{z_t^1, \ldots, z_t^K\} \qquad s_{t-1}^i = h_{t-1}^i \oplus z_{t-1}^i$$

$$\overline{h}_t = f_{GGRU}(\overline{h}_{t-1}, \overline{z}_{t-1}, a_{t-1}) :$$

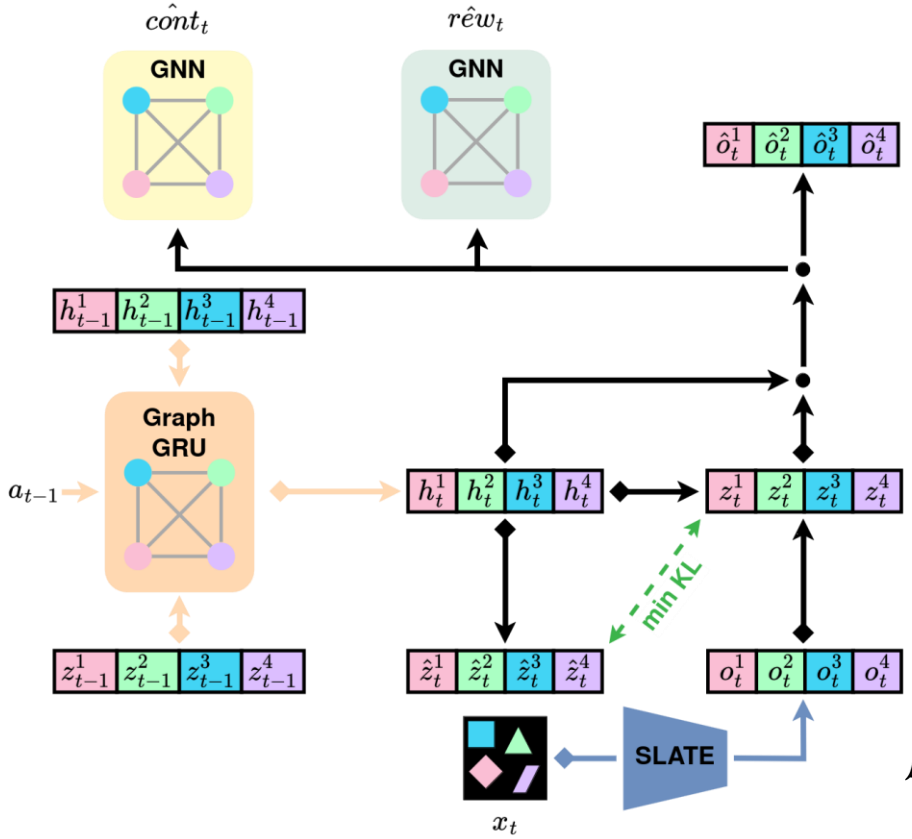
$$y_h^i = \mathtt{node}_h(s_{t-1}^i, a_{t-1}, \textstyle\sum_{j \neq i} \mathtt{edge}_h(s_{t-1}^i, s_{t-1}^j, a_{t-1}))$$

$$r^i, c^i, u^i \overset{\text{split}}{=} y_h^i$$

$$h_t^i = \sigma(u^i) \odot \tanh\left(\sigma(r^i) \odot c^i\right) + (1 - \sigma(u^i)) \odot h_{t-1}^i$$

AIRI

OC-Dreamer: предсказание вознаграждения



$$\overline{h}_t = \{h_t^1, \ldots, h_t^K\} \qquad \overline{z}_t = \{z_t^1, \ldots, z_t^K\} \qquad s_{t-1}^i = h_{t-1}^i \oplus z_{t-1}^i$$

$$\hat{rew}_t = p_\phi(\hat{rew}_t | \overline{h}_t, \overline{z}_t) :$$

$$y_{rew}^i = \texttt{node}_{rew}(s_{t-1}^i, \textstyle\sum_{j \neq i} \texttt{edge}_{rew}(s_{t-1}^i, s_{t-1}^j))$$

$$rew_t = \texttt{MLP}_{rew}\left(y_{rew}^1 \oplus \ldots \oplus y_{rew}^K\right)$$

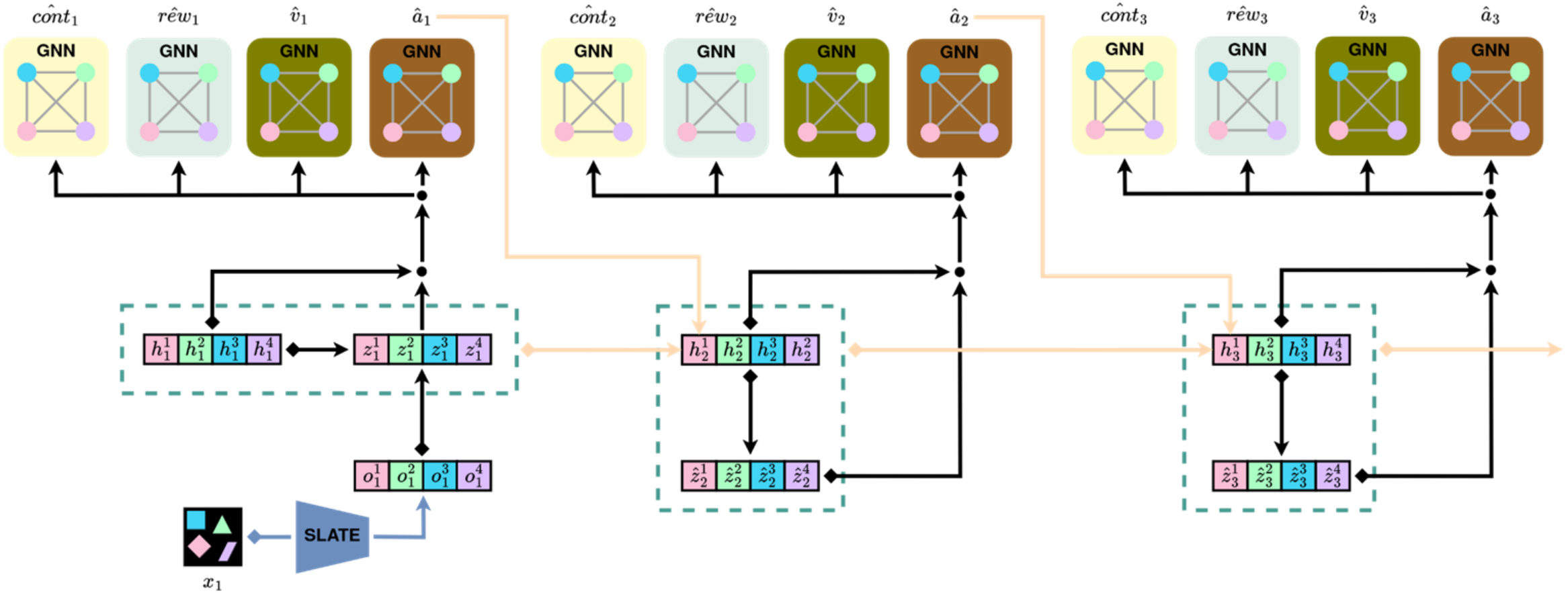
$$\mathcal{L}_{pred}(\phi) = -\textstyle\sum_{i=1}^K \left[\ln p_\phi(o_t^i | h_t^i, z_t^i)\right] - \ln p_\phi(rew_t | \overline{h}_t, \overline{z}_t) - \ln p_\phi(cont_t | \overline{h}_t, \overline{z}_t)$$

$$\mathcal{L}_{dyn}(\phi) = \textstyle\sum_{i=1}^K \max\left(1, \mathrm{KL}\left[\mathrm{sg}(q_\phi(z_t^i | h_t^i, o_t^i)) \parallel p_\phi(z_t^i | h_t^i)\right]\right)$$

$$\mathcal{L}_{rep}(\phi) = \textstyle\sum_{i=1}^K \max\left(1, \mathrm{KL}\left[q_\phi(z_t^i | h_t^i, o_t^i) \parallel \mathrm{sg}(p_\phi(z_t^i | h_t^i))\right]\right)$$

AIRI

OC-Dreamer: обучение актора-критика



Актор: $a_t \sim p_\theta(a_t|\overline{h}_t, \overline{z}_t)$

Критик: $v_t \sim p_\psi(R_t|\overline{h}_t, \overline{z}_t)$

AIRI

OC-Dreamer vs DreamerV3

	DreamerV3		OC-Dreamer	
Component	Model	Layers	Model	Layers
Sequence model	$h_t = f_{GRU}(h_{t-1}, z_{t-1}, a_{t-1})$	MLP and GRU	$\overline{h}_t = f_{GGRU}(\overline{h}_{t-1}, \overline{z}_{t-1}, a_{t-1})$	GNN and GRU
Objects extractor	N / A	N / A	$\overline{o}_t = \texttt{SLATE}(x_t)$	SLATE (frozen)
Encoder	$z_t \sim q_\phi(z_t \mid h_t, x_t)$	CNN and MLP	$z_t^i \sim q_\phi(z_t^i \mid h_t^i, o_t^i)$	SHARED-MLP
Dynamics predictor	$\hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t)$	MLP	$\hat{z}_t^i \sim p_\phi(\hat{z}_t^i \mid h_t^i)$	SHARED-MLP
Reward predictor	$\hat{rew}_t \sim p_\phi(\hat{rew}_t \mid h_t, z_t)$	MLP	$\hat{rew}_t \sim p_\phi(\hat{rew}_t \mid \overline{h}_t, \overline{z}_t)$	GNN and MLP
Continue predictor	$\hat{cont}_t \sim p_\phi(\hat{cont}_t \mid h_t, z_t)$	MLP	$\hat{cont}_t \sim p_\phi(\hat{cont}_t \mid \overline{h}_t, \overline{z}_t)$	GNN and MLP
Decoder	$\hat{x}_t \sim p_\phi(\hat{x}_t \mid h_t, z_t)$	CNN and MLP	$\hat{o}_t^i \sim p_\phi(\hat{o}_t^i \mid h_t^i, z_t^i)$	SHARED-MLP
Critic	$\hat{v}_t \sim p_\psi(\hat{v}_t \mid h_t, z_t)$	MLP	$\hat{v}_t \sim p_\psi(\hat{v}_t \mid \overline{h}_t, \overline{z}_t)$	GNN and MLP
Actor	$\hat{a}_t \sim p_\theta(\hat{a}_t \mid h_t, z_t)$	MLP	$\hat{a}_t \sim p_\theta(\hat{a}_t \mid \overline{h}_t, \overline{z}_t)$	GNN and MLP

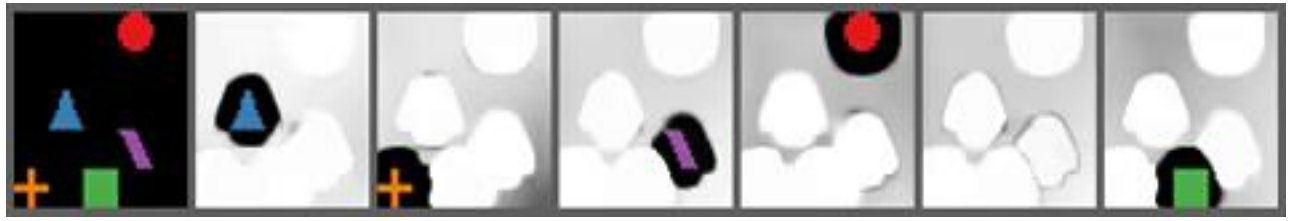
Среды: Shapes2D

Navigation
5x5

Navigation
10x10

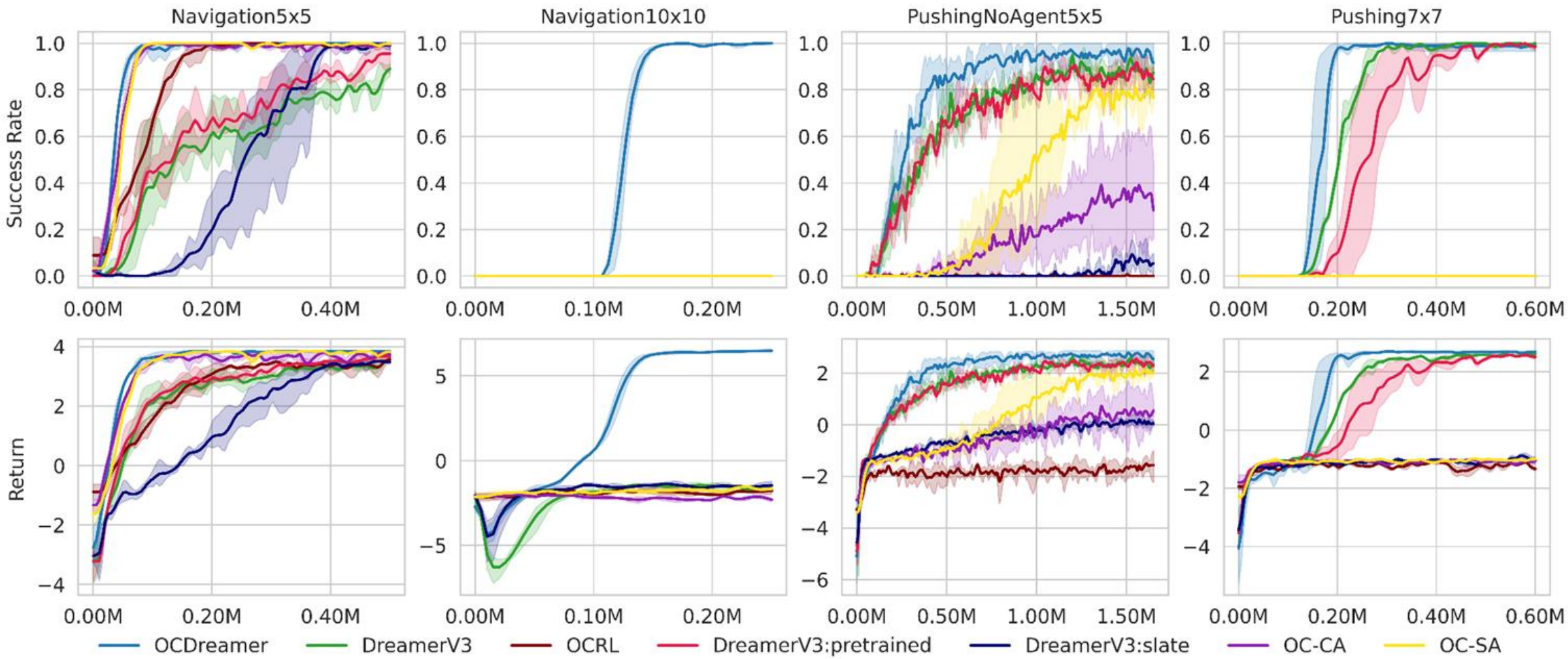
PushingNoAgent
5x5

Pushing
7x7



Visualization of object attention maps generated
by SLATE

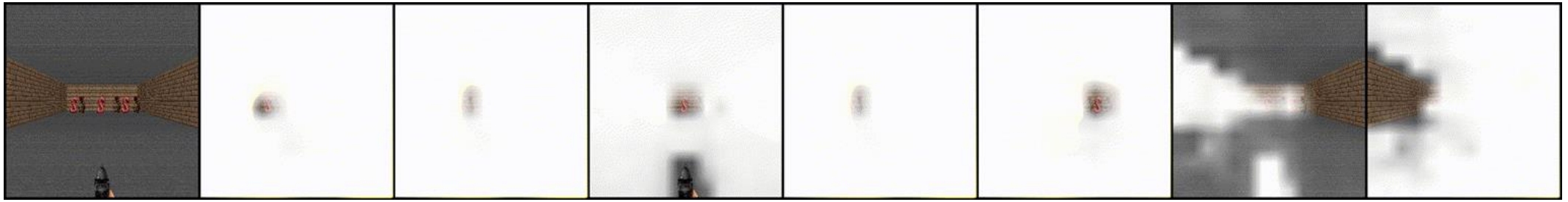
Результаты: Shapes2D



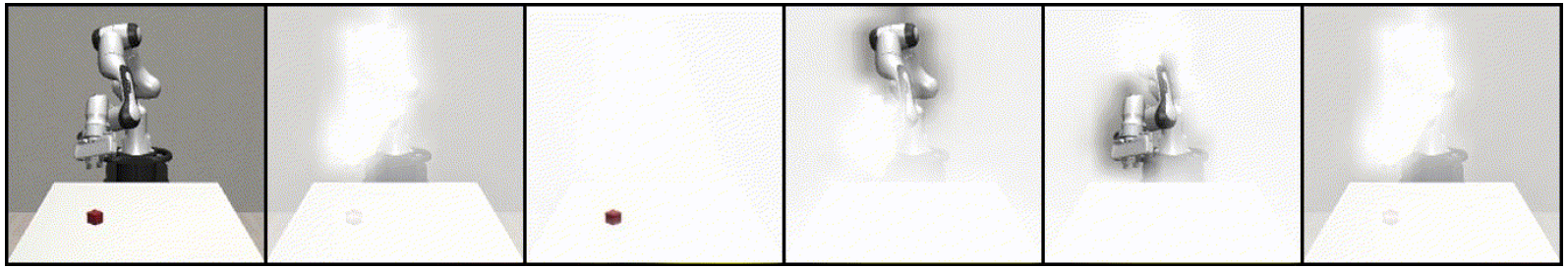
Среды: CausalWorld, VizDoom, Robosuite

CausalWorld: Block Lifting (SLATE)

VizDoom: Defend The Line (DINOSAur)

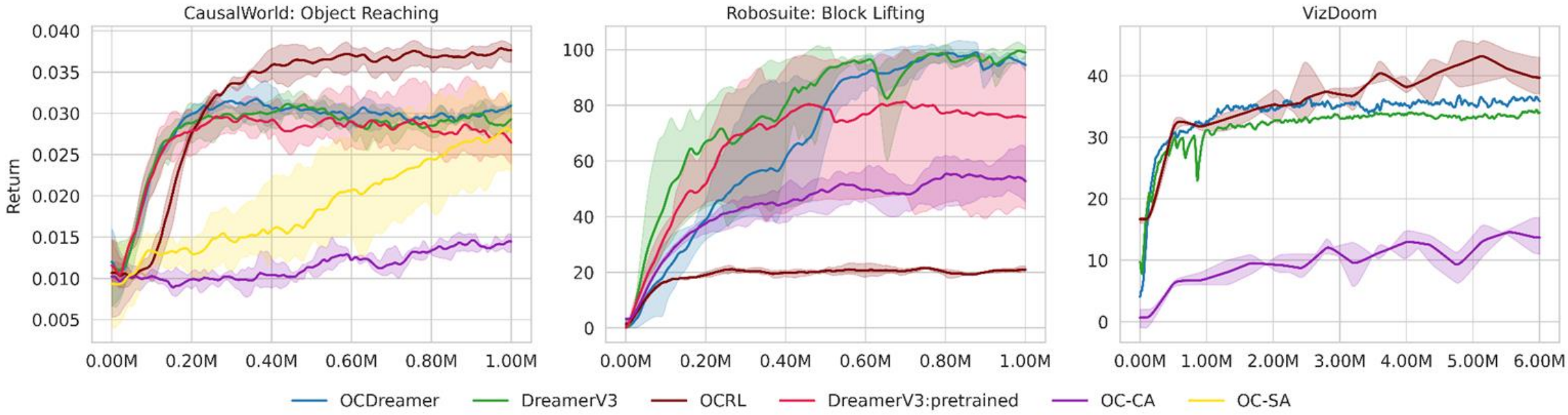


Robosuite: Block Lifting (DINOSAur)



AIRI

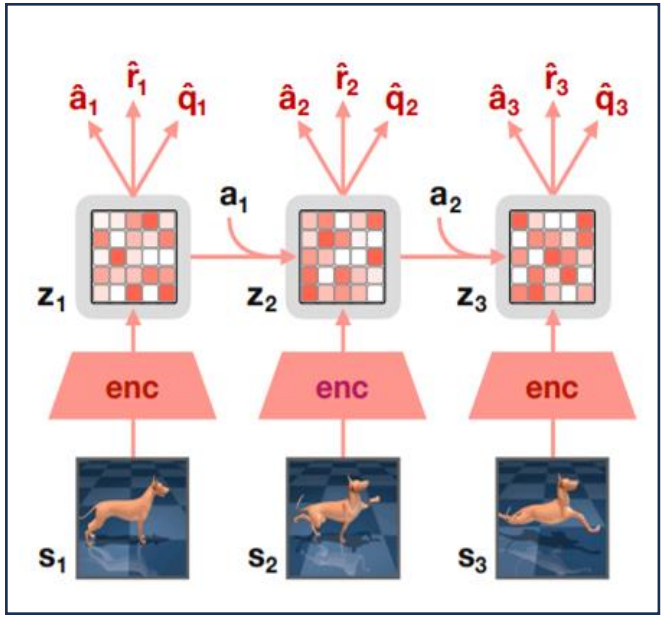
Результаты: CausalWorld, VizDoom, Robosuite



Модель мира в TD-MPC

$$
\begin{aligned}
&\text{Encoder} && \mathbf{z} = h(\mathbf{s}, \mathbf{e}) \\
&\text{Latent dynamics} && \mathbf{z}' = d(\mathbf{z}, \mathbf{a}, \mathbf{e}) \\
&\text{Reward} && \hat{r} = R(\mathbf{z}, \mathbf{a}, \mathbf{e}) \\
&\text{Terminal value} && \hat{q} = Q(\mathbf{z}, \mathbf{a}, \mathbf{e}) \\
&\text{Policy prior} && \hat{\mathbf{a}} = p(\mathbf{z}, \mathbf{e})
\end{aligned}
$$

$$
\mu^*, \sigma^* = \arg\max_{(\mu,\sigma)} \mathop{\mathbb{E}}_{(\mathbf{a}_t, \mathbf{a}_{t+1}, \ldots, \mathbf{a}_{t+H}) \sim \mathcal{N}(\mu, \sigma^2)} \left[\gamma^H Q(\mathbf{z}_{t+H}, \mathbf{a}_{t+H}) + \sum_{h=t}^{H-1} \gamma^h R(\mathbf{z}_h, \mathbf{a}_h) \right]
$$

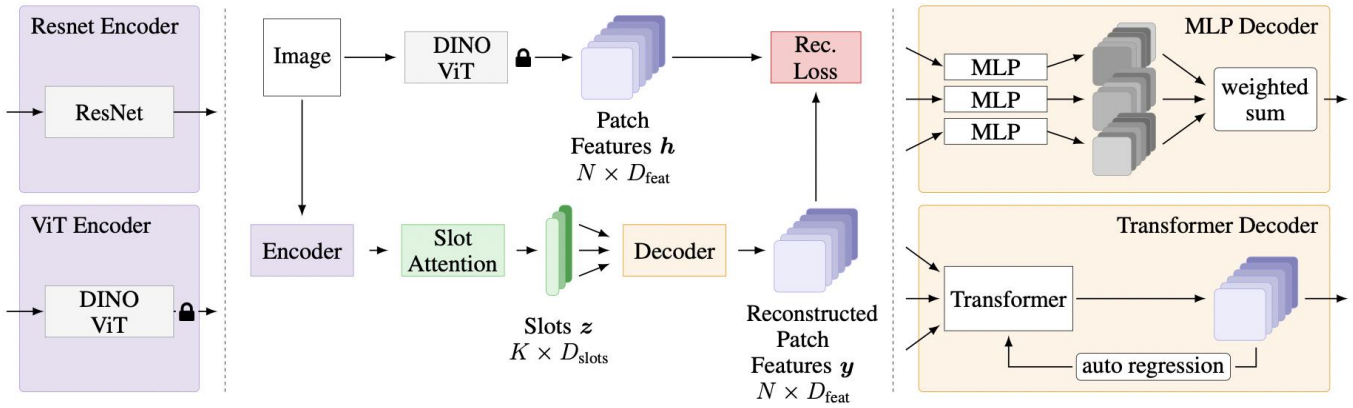


- Управление на основе прогнозирующих моделей и априорной стратегией
- Локальная оптимизация стратегии на основе MPPI (Model Predictive Path Integral)
- SimNorm схема нормализации при проецировании в латентное пространство вектора z

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In International Conference on Machine Learning (ICML), 2022.
Hansen, Nicklas, Hao Su, and Xiaolong Wang. "TD-MPC2: Scalable, Robust World Models for Continuous Control." The Twelfth International Conference on Learning Representations, 2024.

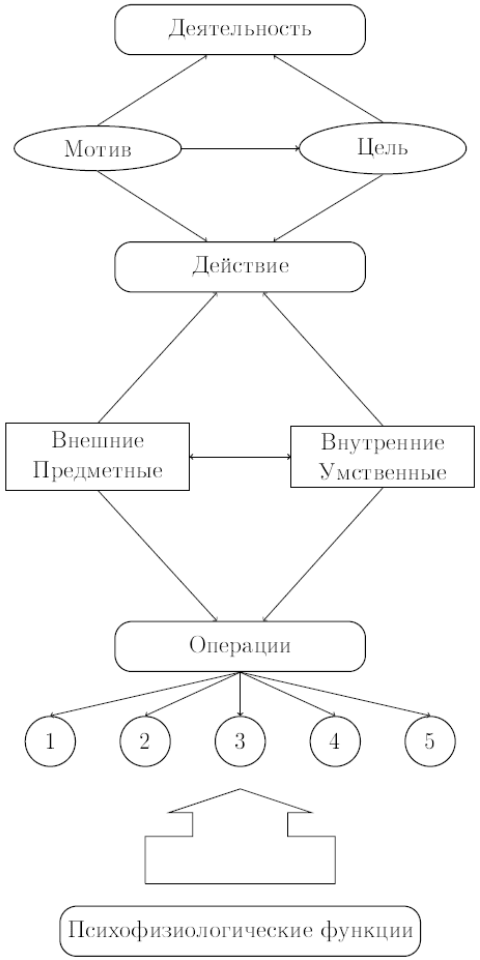
AIRI

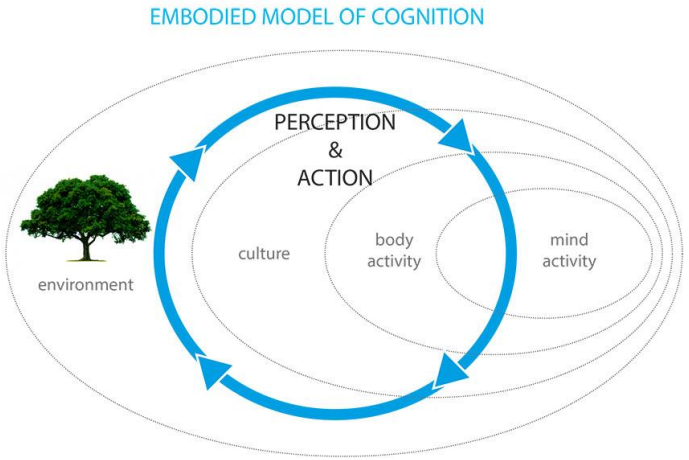
Dinousar – выделение объектов из реалистичных изображений



Seitzer, Maximilian, et al. "Bridging the Gap to Real-World Object-Centric Learning." The Eleventh International Conference on Learning Representations, 2023.

Теория деятельности





Мышление можно рассматривать (моделировать) только с учетом его взаимосвязи с физическим субстратом (телом)

Важнейшая задача в робототехнике — автономное планирование поведения

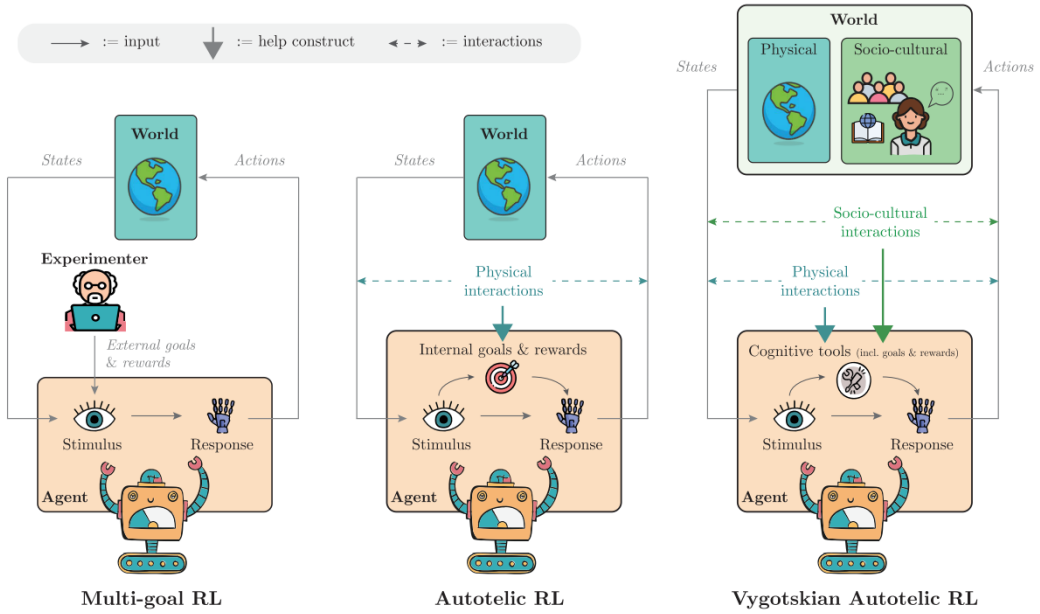
Поведение человека – двойная иерархическая структура мотивы-цели и действия-операции

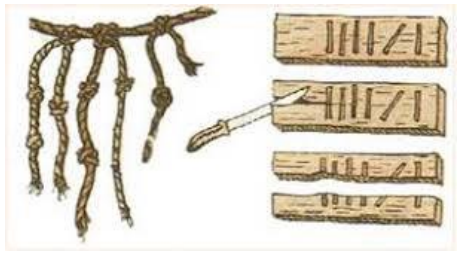
Действия человека предметны, цели носят социальный характер

Сознание и деятельность непрерывно связаны

Wilson, Margaret. «Six views of embodied cognition». Psychonomic Bulletin & Review. 2002.
Леонтьев А.Н. Деятельность. Сознание. Личность. М.: Политиздат, 1977. Вып. Изд. 2-е. 304 с.

Культурно-исторический подход





Социальная среда — главный источник развития личности

Развитие когнитивных функций происходит через использование «психологических орудий»

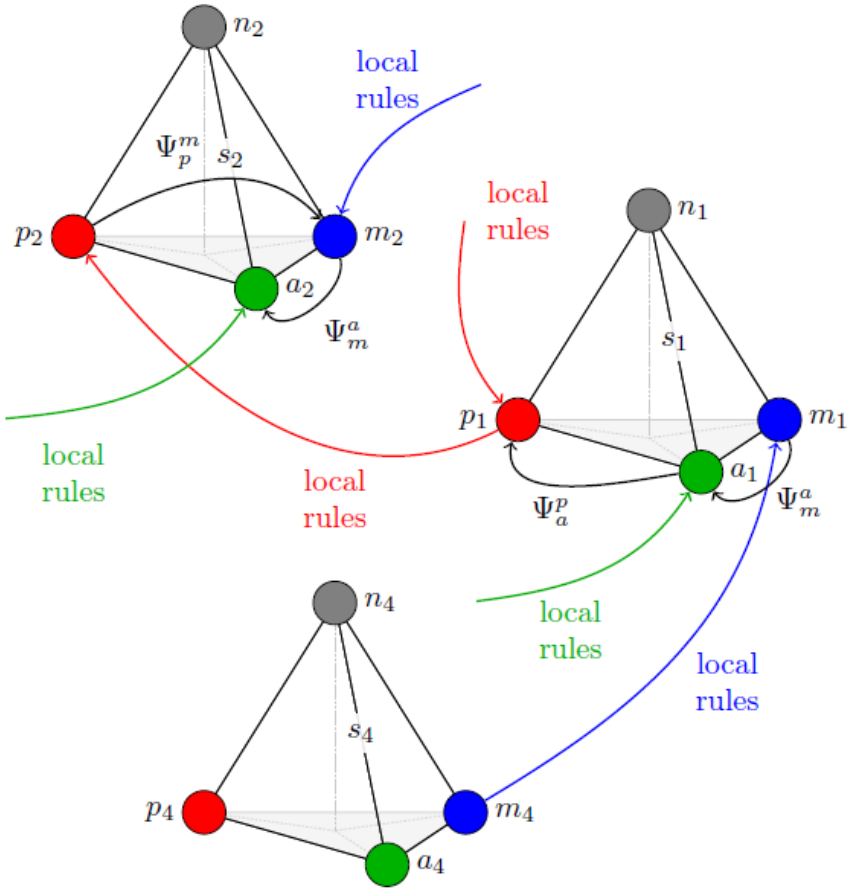
Развитие — этапы овладения системой знаков-символов (письмо, счет и т.п.)

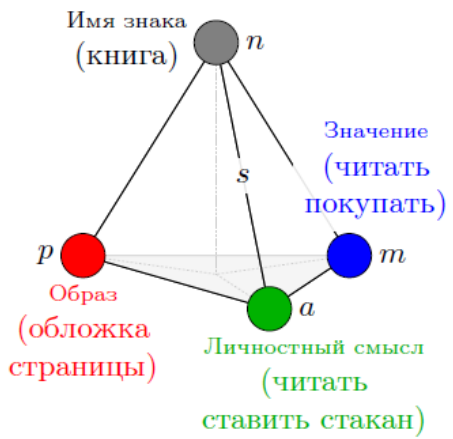
Сознание развивается через диалог: диалог ребенка со взрослым, взрослого со взрослым

Знак — это искусственно созданный человеком стимул, средство для управления своим поведением и поведением других

Colas C. et al. Vygotskian Autotelic Artificial Intelligence: Language and Culture Internalization for Human-Like AI. 2022.

Знаковая картина мира





Представляемая сущность описывается тремя каузальными структурами:

образ – представление взаимосвязи внешних сигналов и внутренних характеристик агента – сенсо-моторное представление,

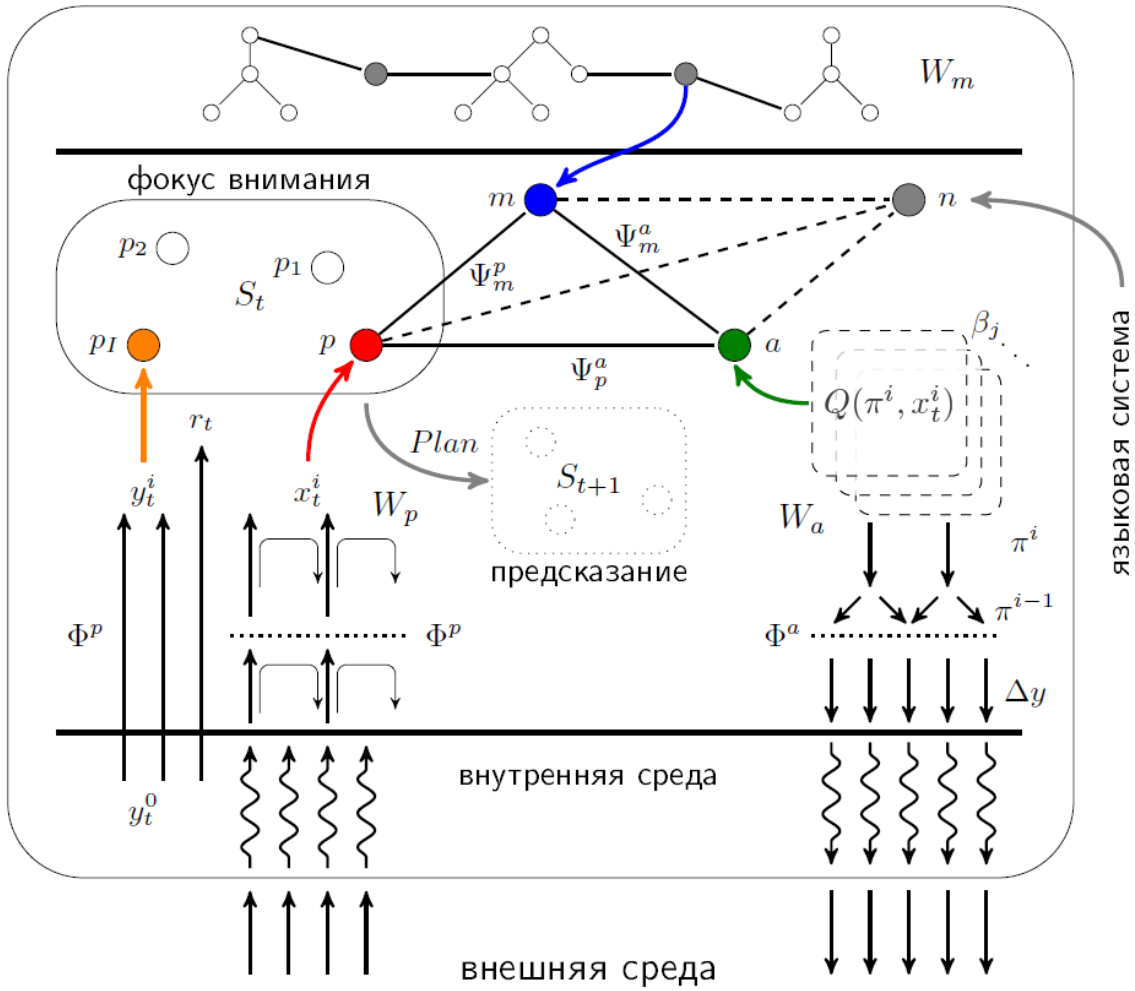
значение – обобщенное согласованное знание о соотношениях во внешнем мире, представленное в системе языка,

личностный смысл – ситуационная потребностно-мотивационная интерпретация значений

Осипов Г.С. и др. Знаковая картина мира субъекта поведения. М.: Физматлит, 2018. 264 с.
Осипов Г.С., Панов А.И., Чудова Н.В. Управление поведением как функция сознания. I. Картина мира и целеполагание // Известия Российской академии наук. Теория и системы управления. 2014. № 4. С. 49–62.

AIRI

Реализация в архитектуре семиотического агента



Иерархическая реализация процедур распознавания Φ^p и разворачивания действий Φ^a

Использование значений для разбиения задачи на подзадачи и поиска потенциально применимых к ситуации действий

Оценка действий в текущей ситуации с помощью функции полезности

Разделение внутренних и внешних признаков

Осипов Г.С., Панов А.И. Синтез рационального поведения когнитивного семиотического агента в динамической среде // Искусственный интеллект и принятие решений. 2020. № 4. С. 80–97.

05

Робототехнические приложения

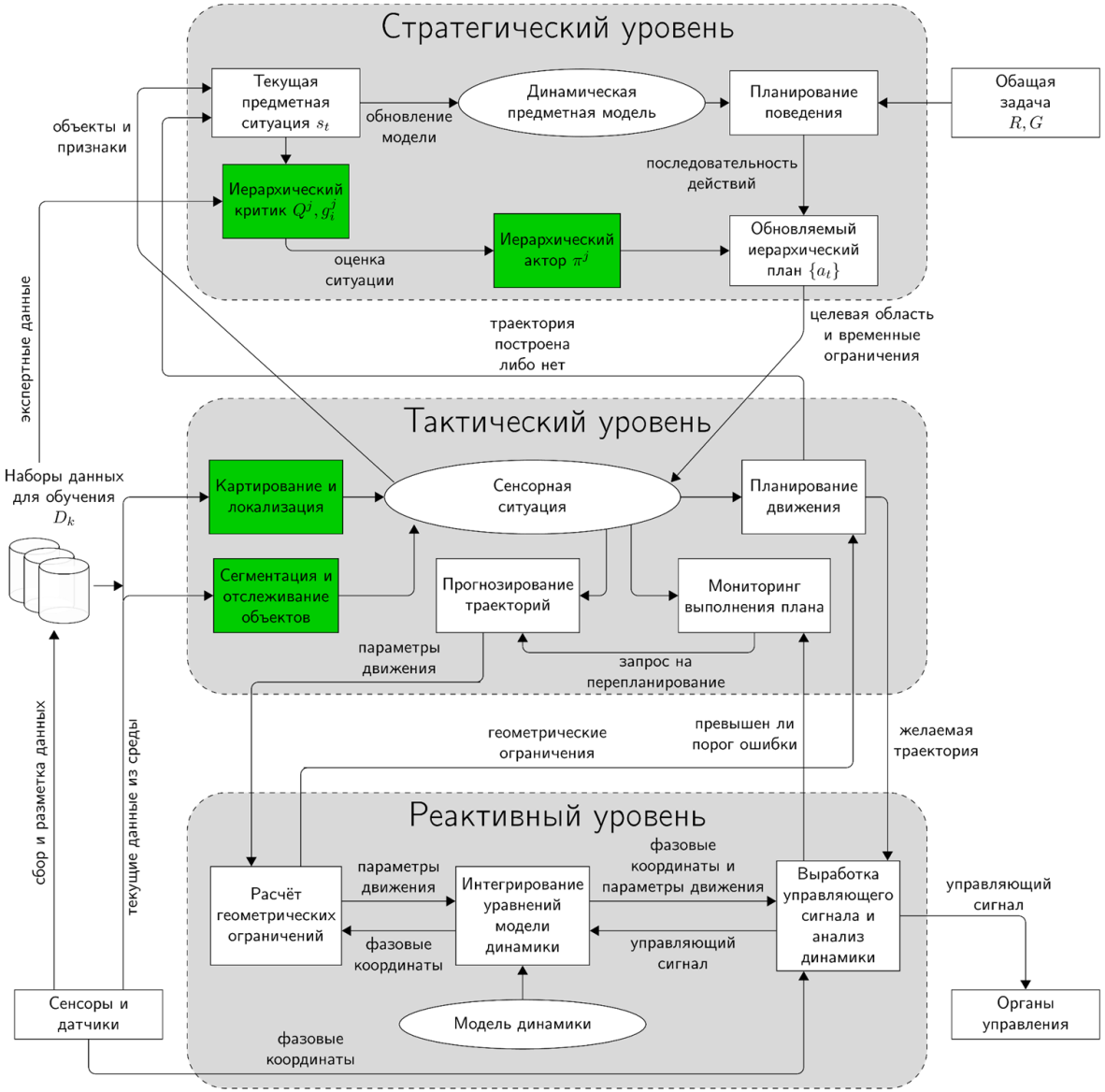
Робот как воплощенный когнитивный агент

- **Воплощенный (embodied) агент** — дословно, имеющий тело, телесный, материализованный
- Воплощение в контексте интеллектуальных агентов – наличие опосредованной связи с внешней средой
- Воплощение характеризуется набором **признаков** «тела» агента, влияющих на динамику процессов во внешней среде, и набором доступных действий, меняющих внешнюю среду
- **Когнитивный (cognitive) агент** — дословно, знающий, способный познавать
- Когнитивность в контексте интеллектуальных агентов – наличие способности **обучаться выполнению задач** в среде и переносить полученный опыт (знания) в новые условия (среды и задачи)
- Способность к обучению подразумевает моделирование (реализацию) нескольких **когнитивных функций человека**: научение, память, планирование, целеполагание

STRL: система управления

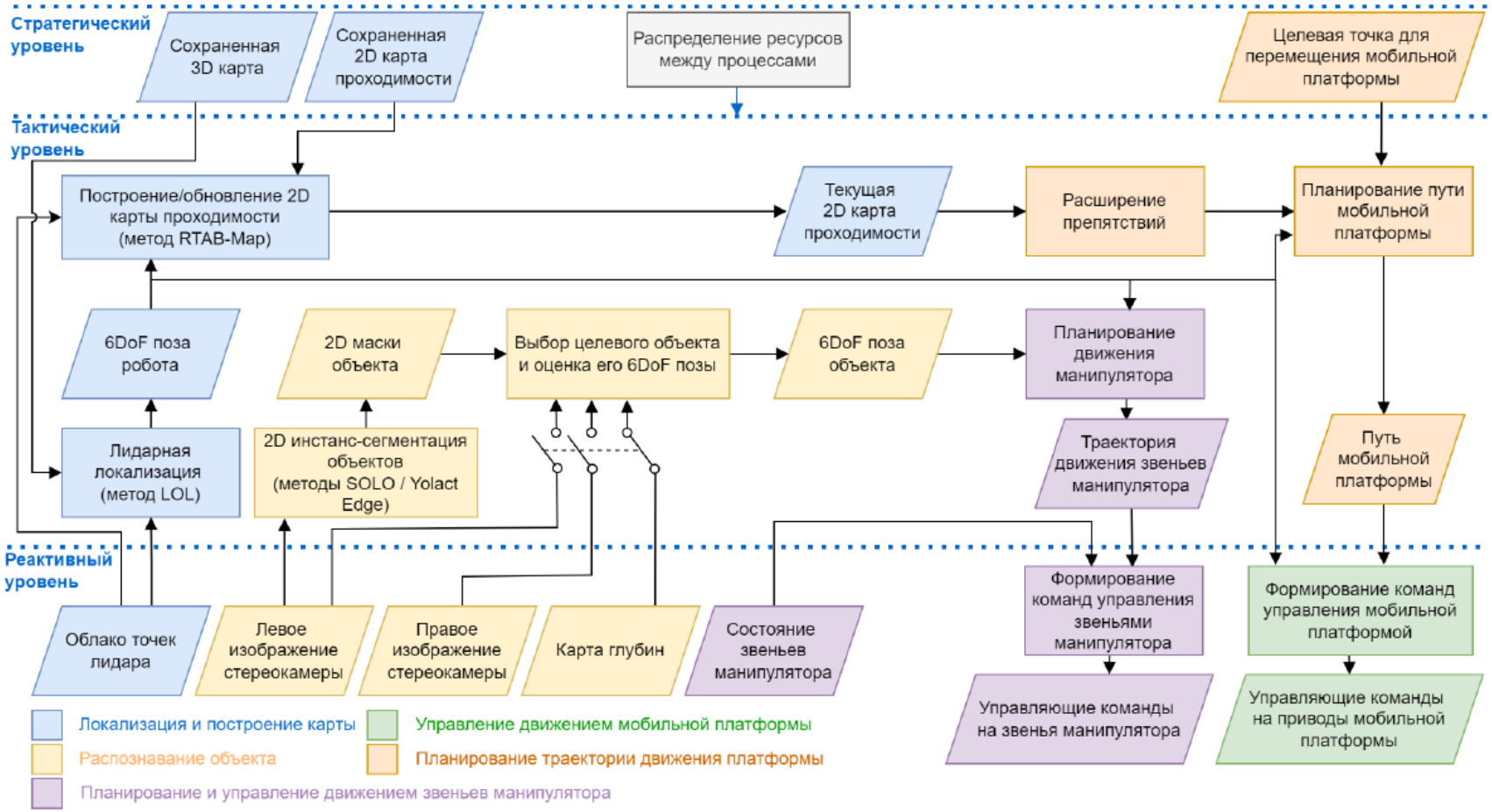
- Иерархическая система управления интеллектуальными агентами и их группами
- Strategic, tactic, reactive layered

Лабораторный мобильный робот Husky с манипулятором UR5



Emel'yanov S. and etc. Multilayer cognitive architecture for UAV control // Cognitive Systems Research. 2016. Т. 39. С. 58–72.
Миронов К.В. и др. STRL-Robotics: интеллектуальное управление поведением робототехнической платформы в человеко-ориентированной среде // Искусственный интеллект и принятие решений. 2023, №2.

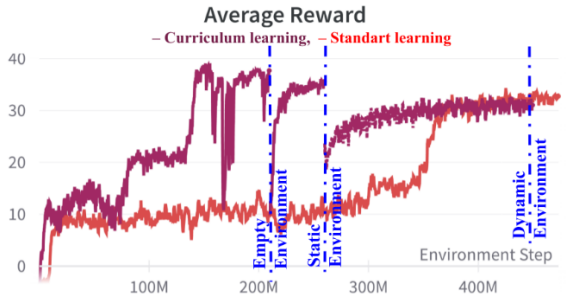
STRL-Robotics: ROS реализация

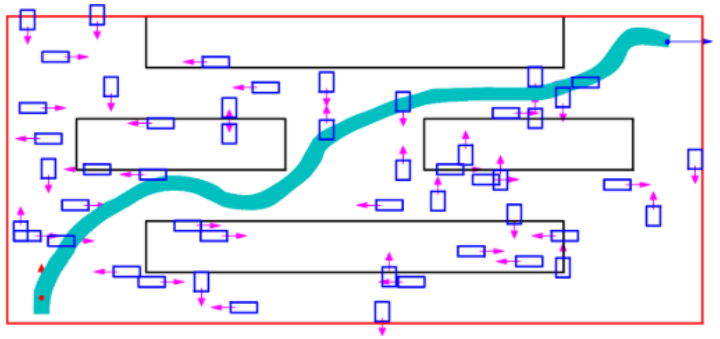


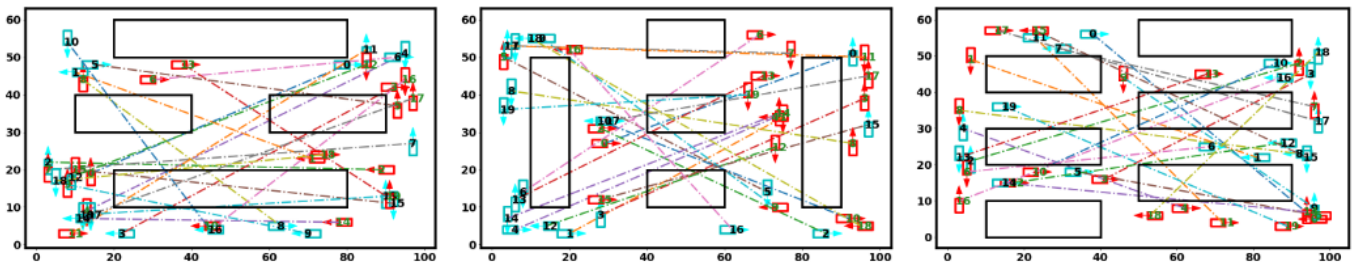
ROS1 реализация архитектуры управления git.mipt.ru/cogmodel/strl/strl_robotics

POLAMP: обучаемые примитивы для планирования перемещений

- **Предобученные примитивы движения д**ля избегания столкновений с динамическими препятствиями и учета кинодинамических ограничений робота
- В качестве верхнеуровневого планировщика выступает **RRT или A***
- **Показывает SOTA результаты** в сложных сценариях (например, на большой парковочной зоне)





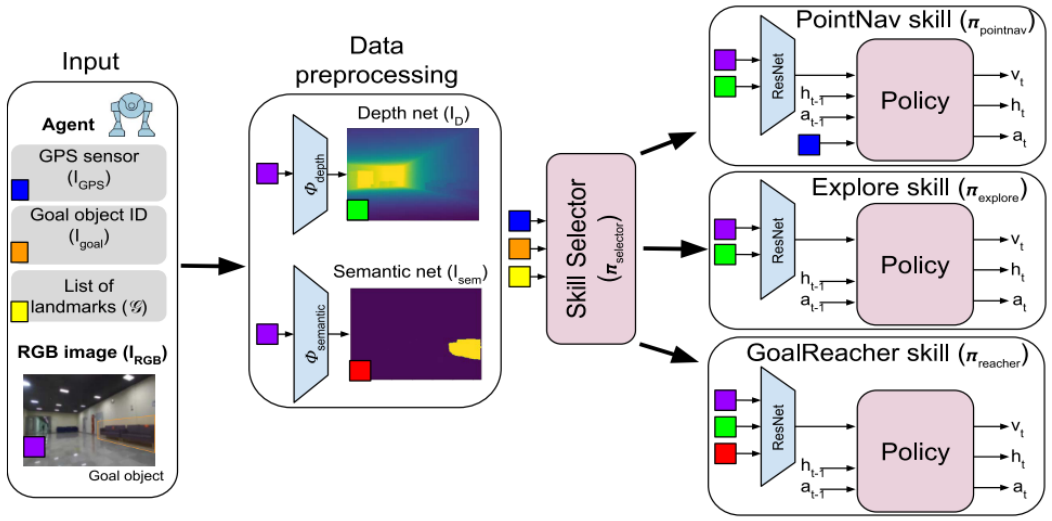


Map	Planner	SR,%	TTR,%	Samples,%	Time,%
1	POLAMP-RRT	100	100	100	100
	POLAMP-A*	100	93	179	103
	RRT-ES	90	120	3851	104
	RL-RRT	40	96	2424	578
	SST*	85	140	4124	111
	SST*(3x)	90	113	11576	351
2	POLAMP-RRT	100	100	100	100
	POLAMP-A*	100	78	121	85
	RRT-ES	62.5	143	1322	107
	RL-RRT	4.5	153	677	308
	SST*	82.5	123	1225	102
	SST*(3x)	100	100	3405	293
3	POLAMP-RRT	100	100	100	100
	POLAMP-A*	100	84	143	89
	RRT-ES	31	102	3426	98
	RL-RRT	8	126	1532	407
	SST*	58.8	141	3560	101
	SST*(3x)	85	111	9073	287

Angulo B. et al. Policy Optimization to Learn Adaptive Motion Primitives in Path Planning With Dynamic Obstacles. AIRI & MIPT. RA-L. https://arxiv.org/abs/2212.14307

HLPO: базовая модель навигации с опорными точками

- Обучение задаче навигации в помещениях
- Иерархическая структура стратегии:
  63% SR в фотореалистичном симуляторе
  Habitat
- Работает на реальном роботе после
  дообучения на реконструированной карте



Method	GT semantic		Learned semantic	
	Success	SoftSPL	Success	SoftSPL
E2E RL	0.18	0.35	0.11	0.24
SemExp	0.24	0.26	0.11	0.17
Planning	0.31	0.26	0.15	0.18
Auxiliary RL	0.51	0.34	0.19	0.19
2RL	0.46	0.33	0.20	0.21
HLPO (Plan)	0.68	0.43	0.37	0.30
HLPO	**0.75**	**0.38**	**0.46**	**0.30**
HLPO (Map)	0.90	0.54	0.61	0.42

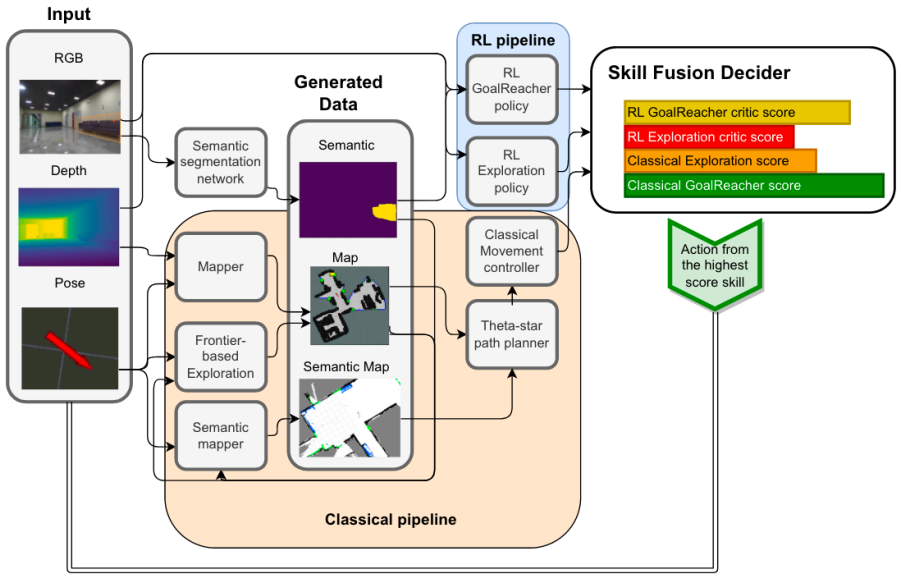
A. Staroverov et al. Hierarchical Landmark Policy Optimization for Visual Indoor Navigation. AIRI & RAS. Access.
http://doi.org/10.1109/ACCESS.2022.3182803

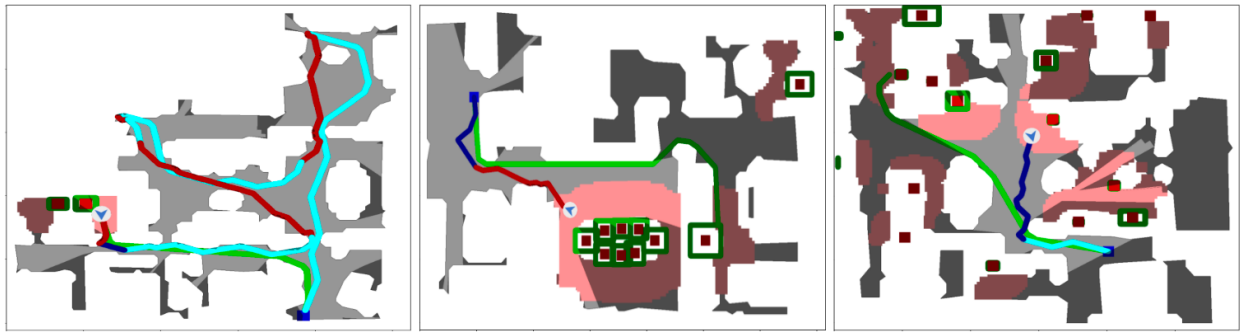
SkillFusion: интеграция умений в базовой модели

- **Задача поиска объектов** в незнакомой фотореалистичной среде по RGB-D камере и GPS
- Интеграция различных классических и RL умений в обучаемом контроллере
- **Победа на соревновании Habitat 2023** и трансфер на реального робота

Method	Success	SPL	SoftSPL
DDPPO [1]	0.18	0.10	0.35
SemExp [14]	0.24	0.14	0.26
Auxiliary RL [8]	0.51	0.29	0.34
SkillFusion	**0.64**	**0.36**	**0.38**

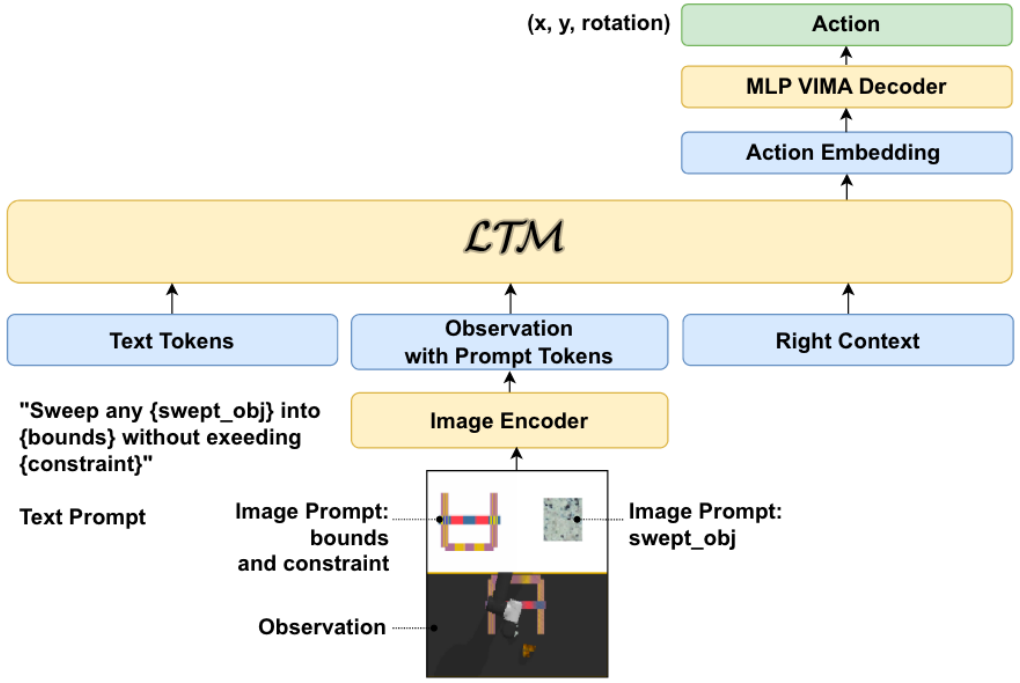
Skill		Metrics		
Explore	GoalReacher	Success	SPL	SoftSPL
Classical	Classical	0.410	0.182	0.263
RL	RL	0.403	0.224	0.321
RL+Classical	Classical	0.511	0.299	0.309
RL+Classical	RL+Classical	0.547	0.316	0.365

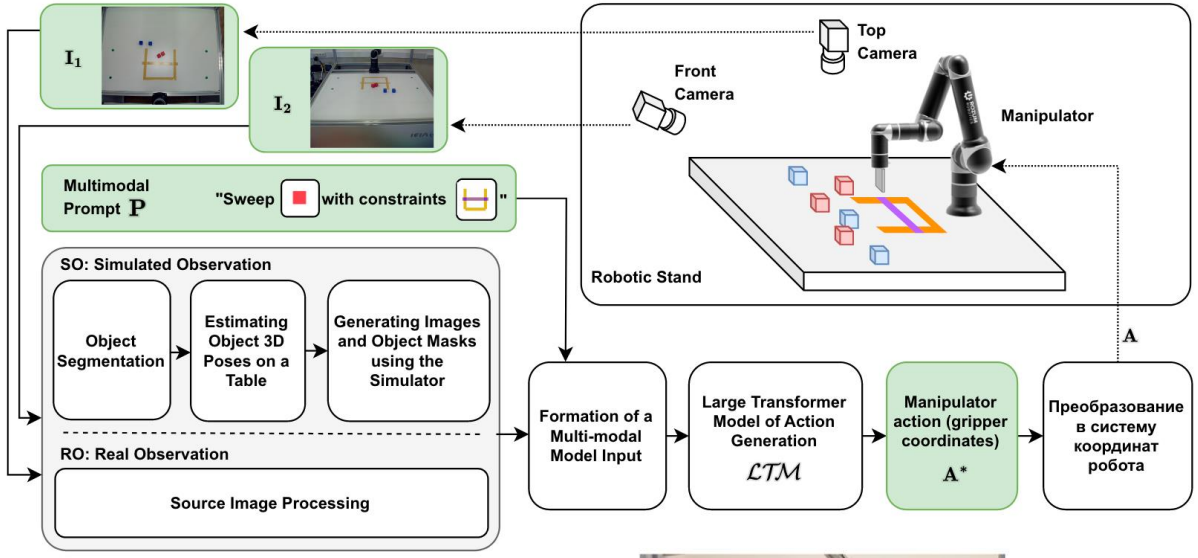


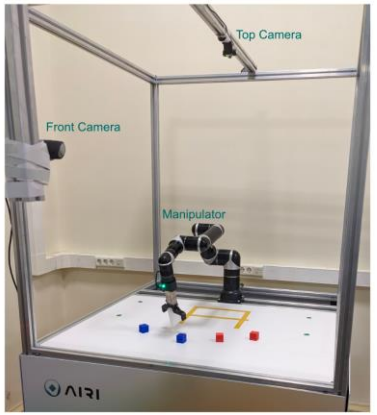


Staroverov A. et al. Hierarchical Landmark Policy Optimization for Visual Indoor Navigation. AIRI & RAS. Access. 2022.
Staroverov A. et al. Skill Fusion in Hybrid Robotic Framework for Visual Object Goal Navigation. AIRI & MIPT. Robotics. 2023.

RozumFormer: перенос базовой модели на реального робота



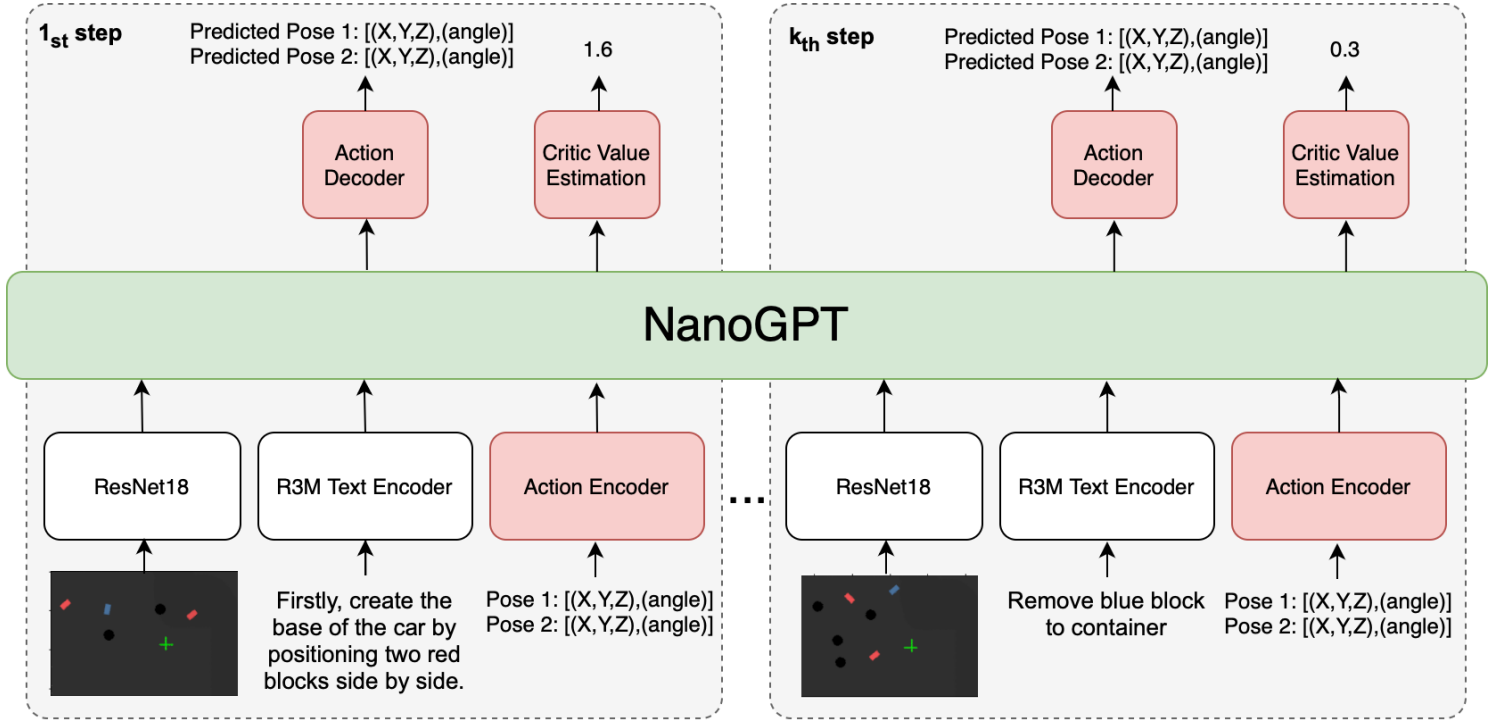


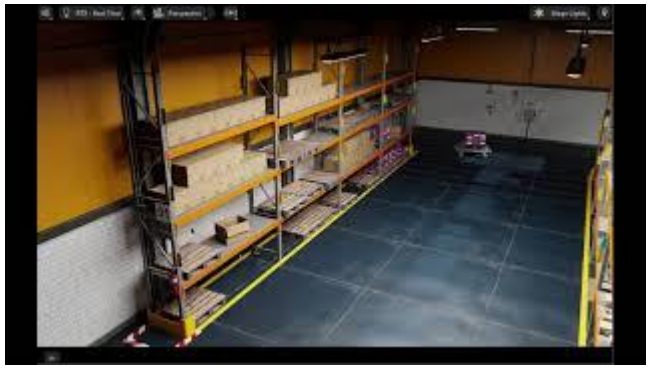


Staroverov A. et al. Fine-tuning Multimodal Transformer Models for Generating Actions in Virtual and Real Environments // IEEE Access. 2023.
https://ieeexplore.ieee.org/document/10323309

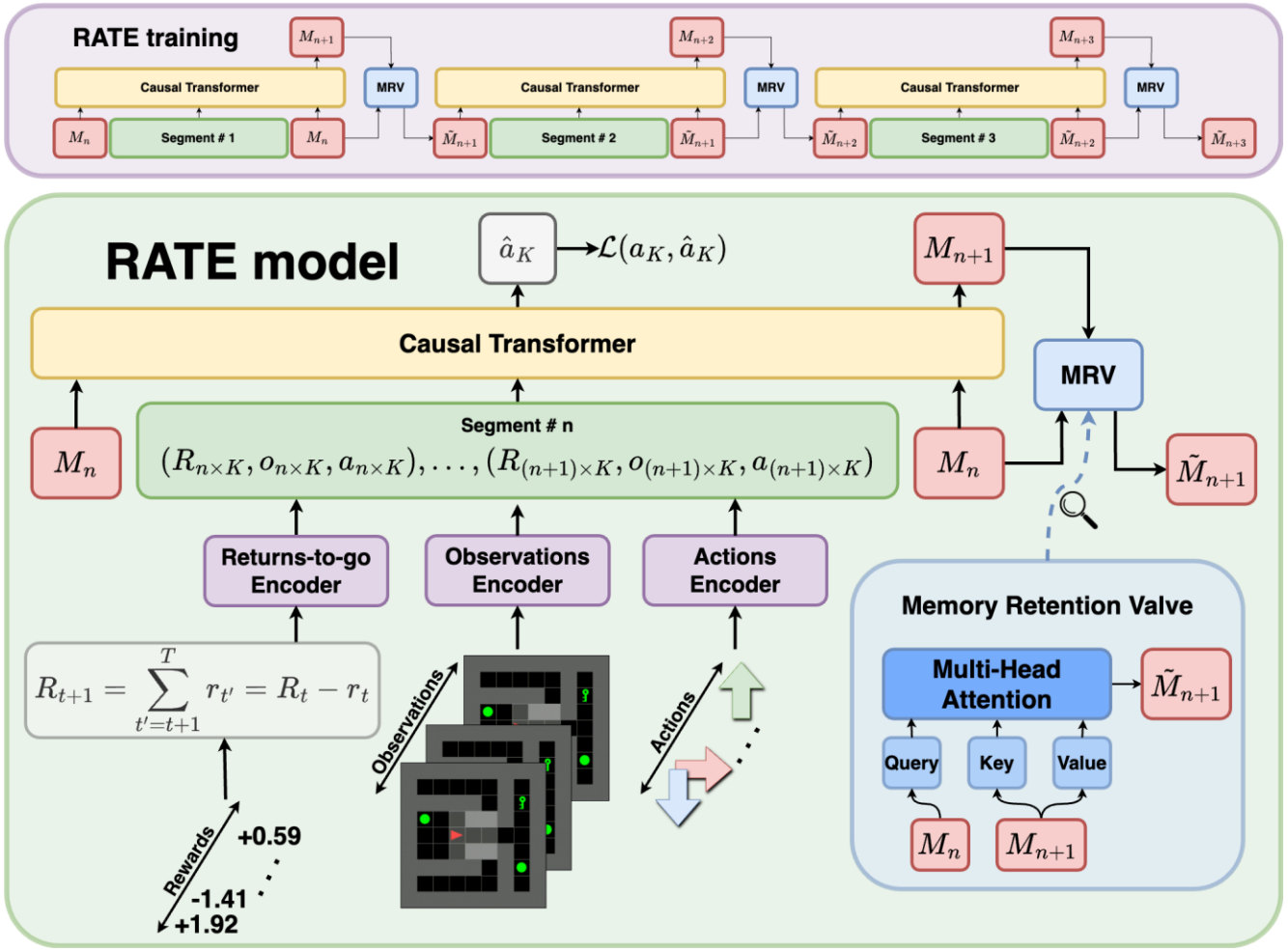
AIRI

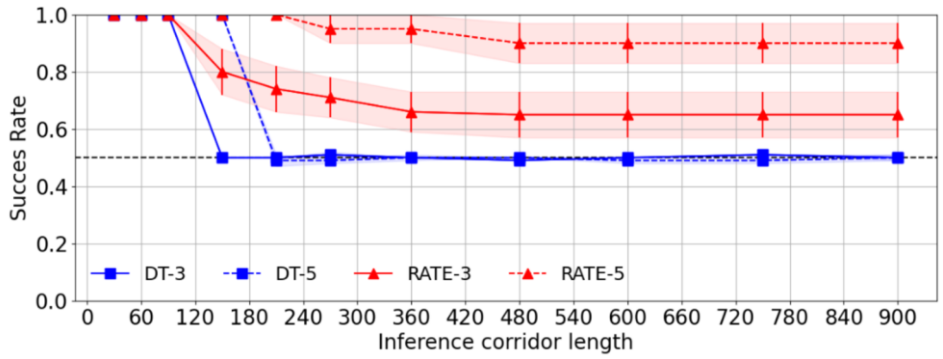
PlanFormer: мультимодальная модель мира и действий

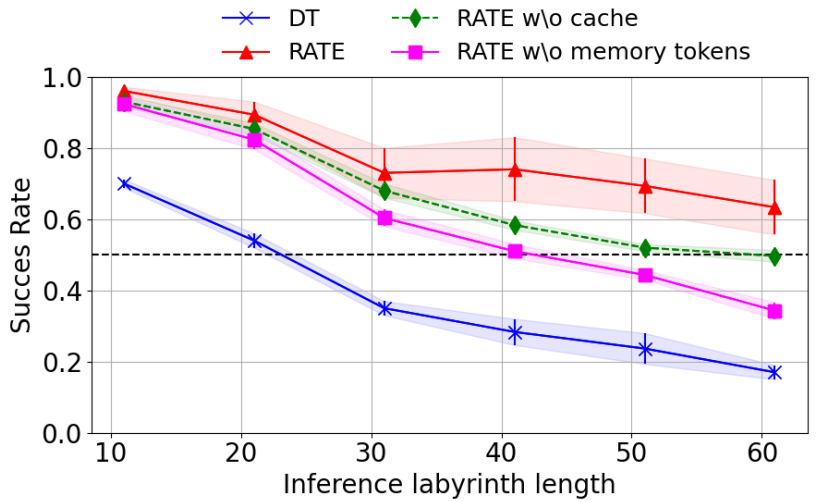




Recurrent Action Transformer with Memory (RATE)







Cherepanov E. et al. Recurrent Action Transformer with Memory // ICML 2024 Next Generation of Sequence Modeling Architectures Workshop. 2024.
http://arxiv.org/abs/2306.09459v4

Итоги и перспективные направления

1. **Обучение с подкреплением –** особый вид нестационарной оптимизационной задачи
2. Обучение с подкреплением на основе **модели мира** – эффективный способ уменьшения количества выборок из среды
3. **Объектная декомпозиция** – естественный путь к более глубокой генерализации в обучении с подкреплением
4. **Выделение объектов –** сложная задача в обучении без учителя, работает в хорошо структурированных средах
5. Объектно-центричное обучение с подкреплением использует **объектную декомпозицию** оптимизационной задачи
6. Необходимо использовать **графовые нейронные сети** для моделирования взаимодействия объектов

AIRI

Контакты

Александр Панов

в.н.с. AIRI & ФИЦ ИУ РАН, директор ЦКМ МФТИ

panov@airi.net

AIRI

airi.net

airi_research_institute

AIRI Institute

AIRI Institute

Telegram

Artificial Intelligence
Research Institute