Мультиагентное обучение с подкреплением: теория и приложения

Александр Панов

д.ф.-м.н., доцент
директор лаборатории CAIS AIRI
директор Центра когнитивного моделирования ИИИ МФТИ

Содержание

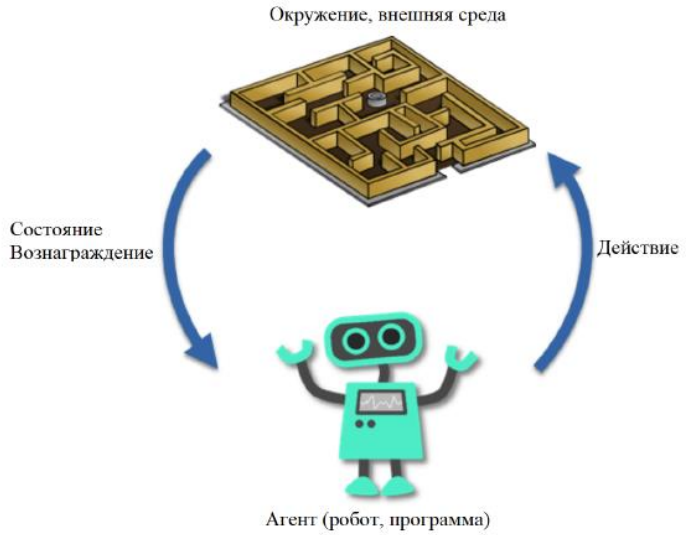
01

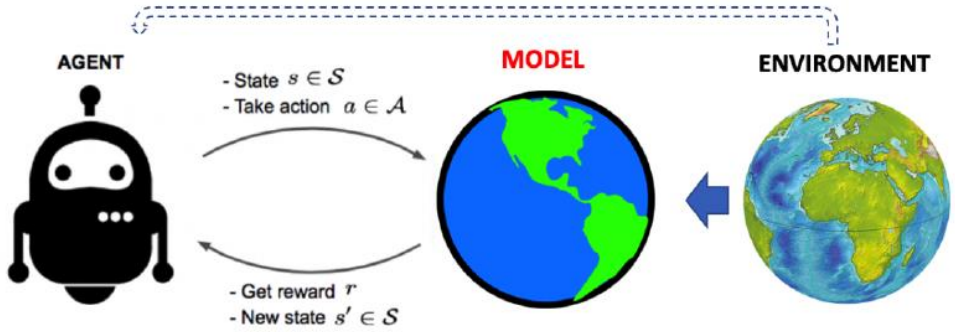
Обучение с подкреплением как задача оптимизации

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** процесса улучшения стратегии агента → алгоритмы оптимизации с ограничениями (PPO)
- **Неэффективность** 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)

$$\pi: S \to A$$

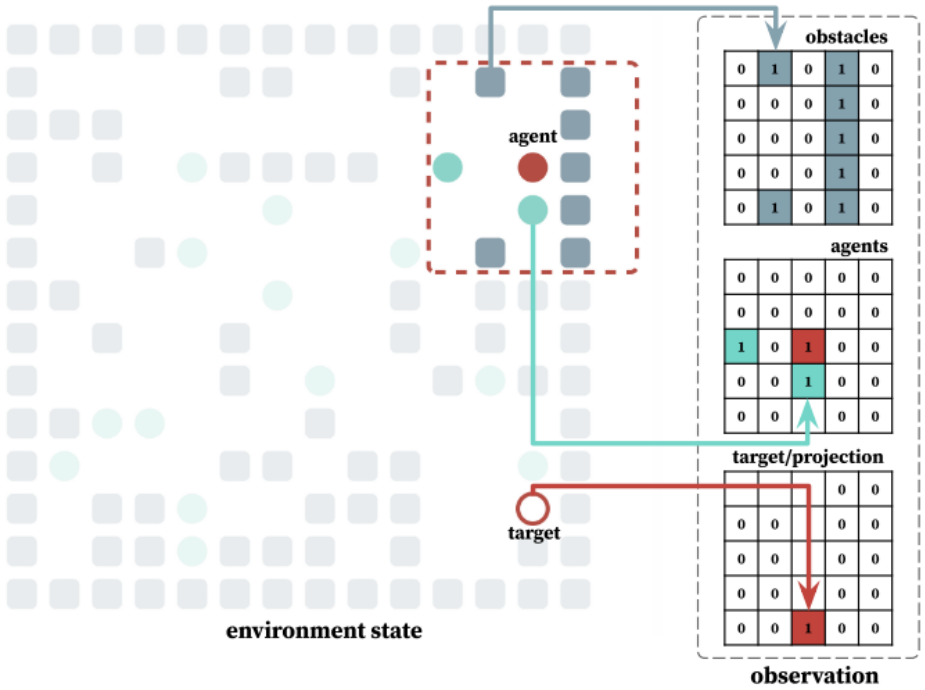
Цель агента — максимизировать ожидаемую **отдачу** по стратегии π:

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

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

Наблюдение и аппроксимация состояния

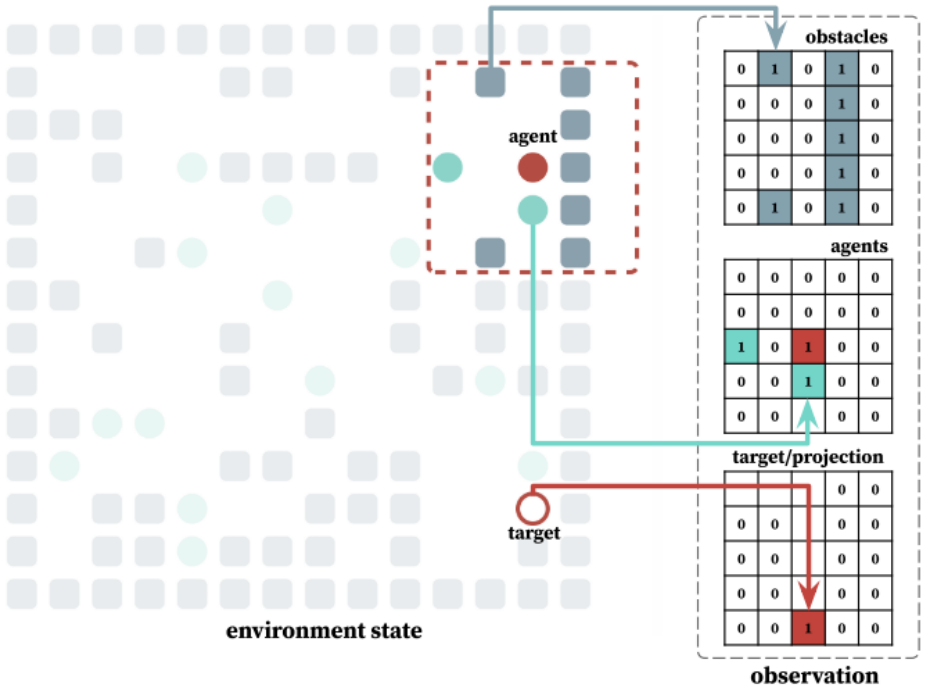
- Также необходимо отметить, что во многих средах условие полной наблюдаемости среды не выполняется
- Агент не имеет непосредственного доступа к информационному состоянию s_t в каждый момент времени, а получает от среды только так называемое **наблюдение** $o_t \in O$
- Последовательность прецедентов $o_1, a_1, r_2, o_2, a_2, r_3, \ldots$ 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**:

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

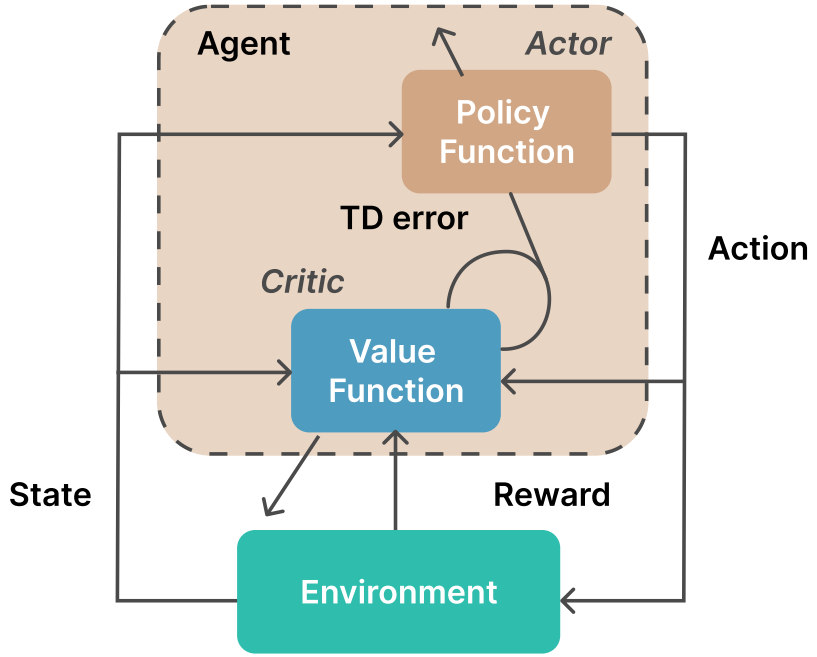
Градиент стратегии и актор-критик

- 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:

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

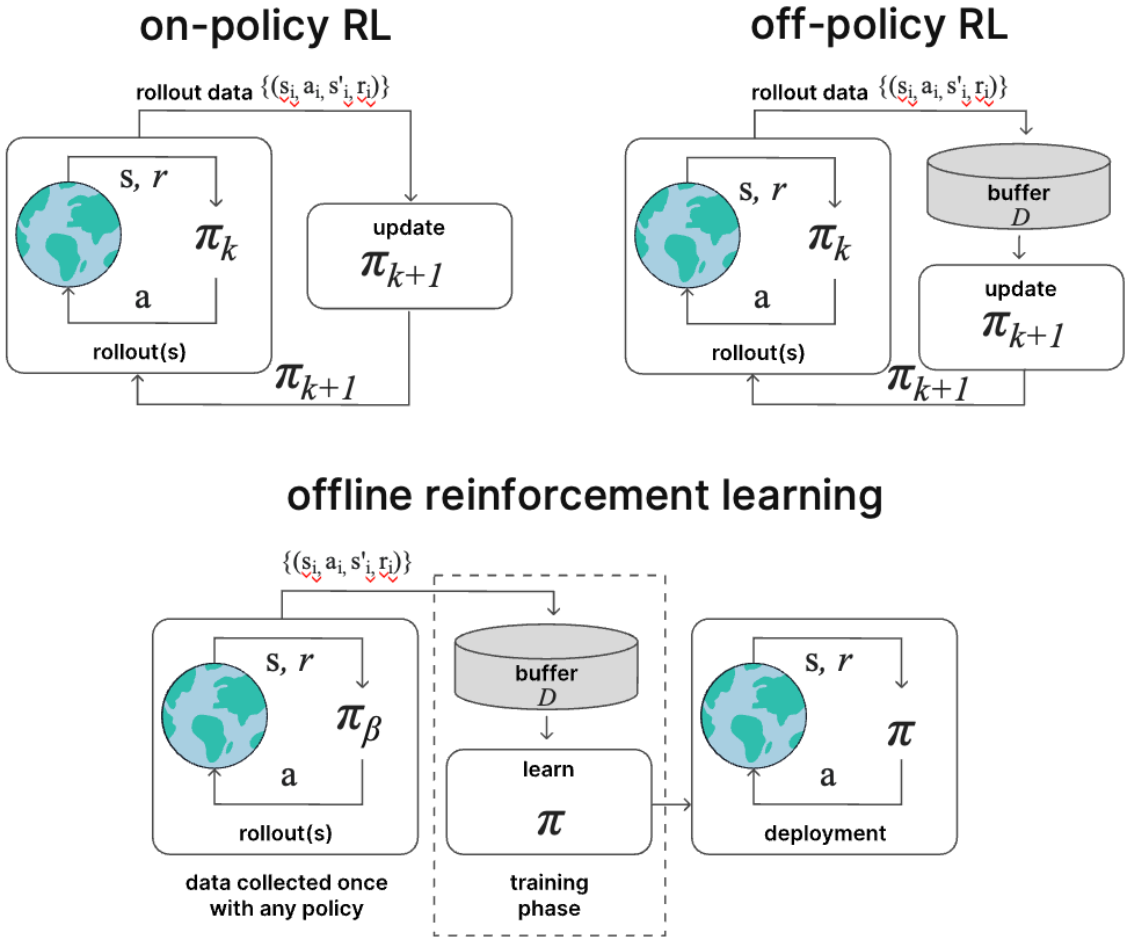
Для оценки значения функции полезности Q используется критик — получается архитектура актора-критика:

- веса **критика** обновляются с помощью функции потерь $\mathcal{L}(\theta)$, обычно чаще, чем веса стратегии,
- веса **актора** обновляются в соответствии с максимизацией функции полезности $J(w)$



Интерактивное и автономное обучение

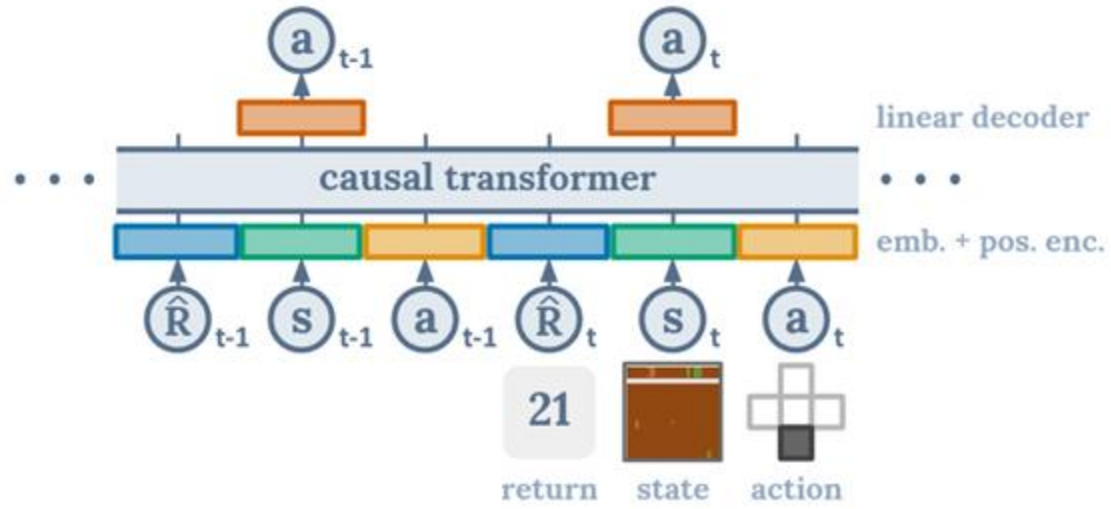
- **Идея:** использовать большие набора данных истории взаимодействия со средой
- **Задача:** найти хорошие примеры в наборе данных, включающих примеры и плохого поведения
- **Обобщение:** хорошее поведение в одних случаях может приводить к хорошему поведению и в других
- **«Сшивка» (stitching):** части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

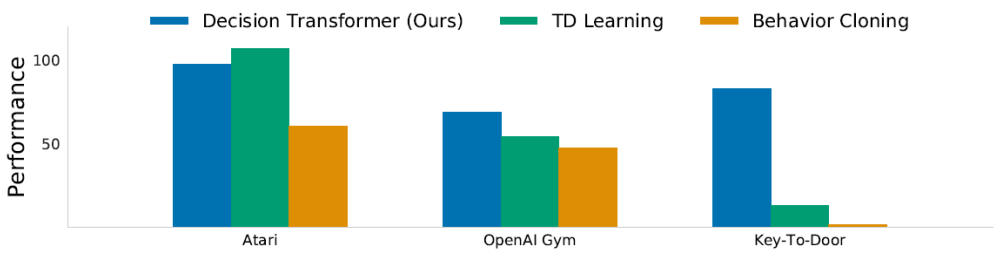


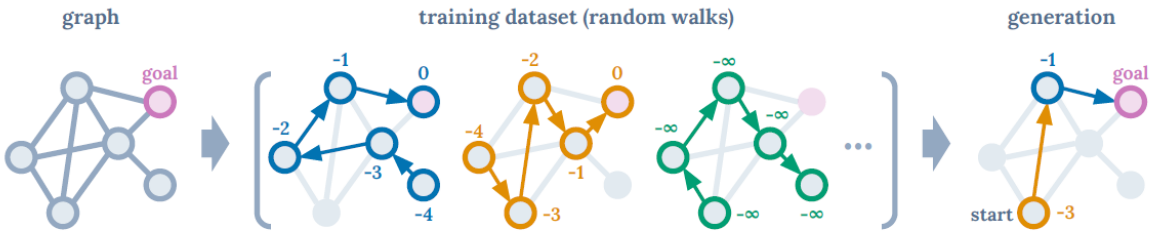
S. Levine et al. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. 2020. https://arxiv.org/abs/2005.01643

Трансформер решений

- Использование трансформера на **токенизированных** экспертных траекториях
- Использование **будущих вознаграждений** для маркера качества (обуславливание)
- **Генеративная модель** в качестве стратегии







Chen, Lili & Lu, Kevin & Rajeswaran, Aravind & Lee, Kimin & Grover, Aditya & Laskin, Michael & Abbeel, Pieter & Srinivas, Aravind & Mordatch, Igor. (2021). Decision Transformer: Reinforcement Learning via Sequence Modeling.

02

Многоагентный RL – особенности задачи

Задача мультиагентного обучения I

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 $P(s'|s, u): S \times U \times S \rightarrow [0, 1]$, в каждый момент времени t
- Вознаграждения генерируются в соответствии с функцией $r(s, \mathbf{u}): S \times S \rightarrow \mathbf{R}$, которая разделяется всеми агентами, а $\gamma \in [0, 1)$ – это дисконтирующий множитель

Задача мультиагентного обучения II

- 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

Задача мультиагентного обучения III

- Цель обучения – найти оптимальную функцию полезности действия
- Пусть $Q_{tot}(\tau_t, u_t, \theta)$ – общая функция полезности, параметризованная θ
- Пусть D – память прецедентов общих действий и историй наблюдений
- Тогда функция потерь для MARL – это TD ошибка для Q_{tot}

$$L(\theta) = \mathbb{E}_D[(r + \gamma \max Q_{tot}(\tau_{t+1}, u_{t+1}, \theta^-)) - Q_{tot}(\tau_{t+1}, u_{t+1}, \theta)]^2$$

- В процессе обучения проявляется дополнительный источник нестационарности изменения u_{t+1}
- Используется принцип централизованное обучение и децентрализованное выполнение с условием

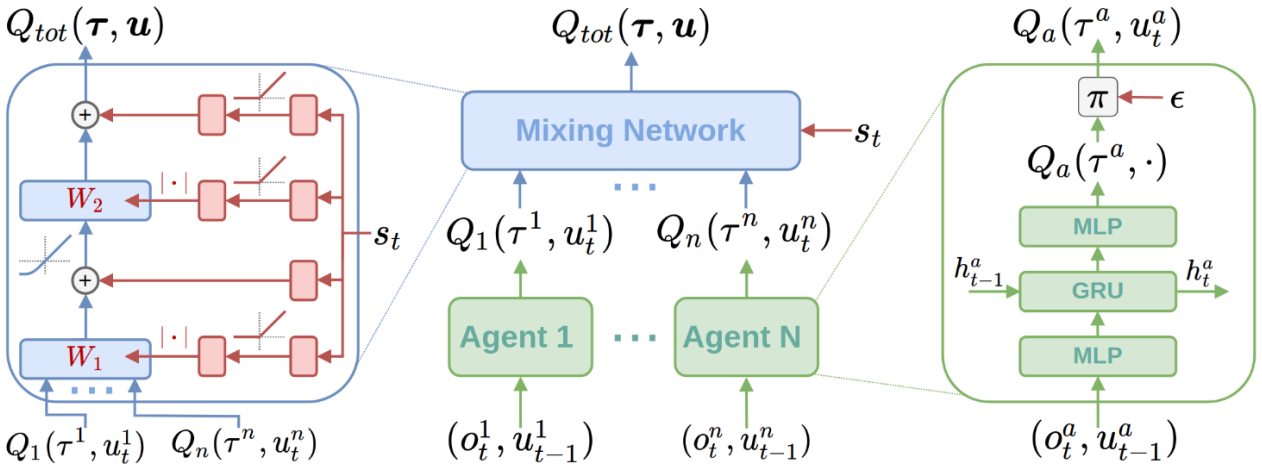
$$argmax\ Q^{\pi}(s, u) = (argmax\ Q_1(\tau^1, u^1), \dots, argmax\ Q_n(\tau^n, u^n))$$

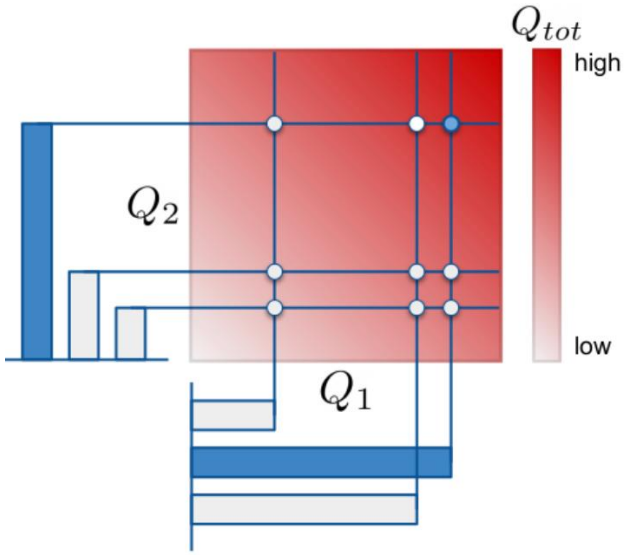
QMIX: монотонная факторизация функции полезности

- Определим монотонность как ограничение отношения Q_{tot} и Q_a:

$$\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \forall a \in A$$

- В условия монотонности достигается максимум Q_{tot} при максимизации Q_i





QMIX: Monotonic value function factorisation for deep multi-agent reinforcement Learning. JMLR 2020 // http://jmlr.org/papers/v21/20-081.html

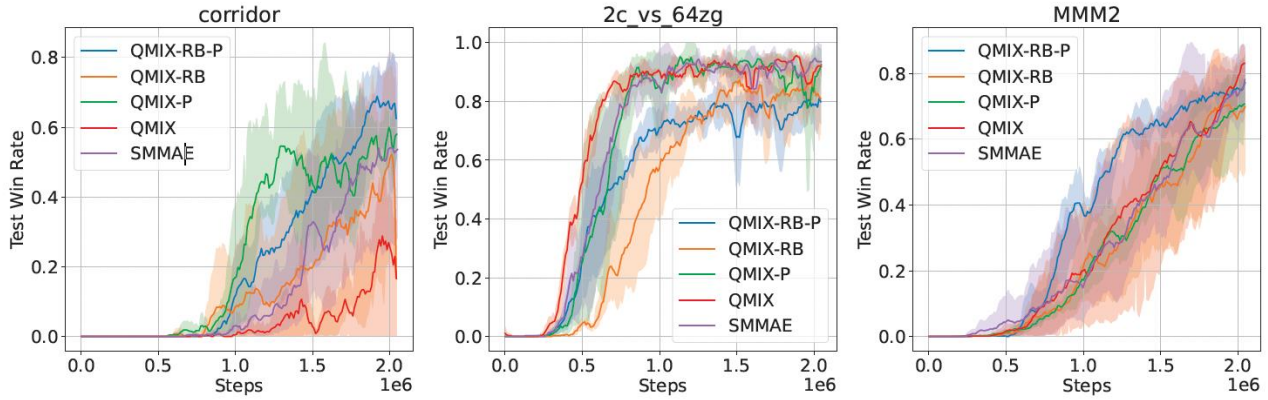
AIRI

03

Онлайн/оффлайн MARL

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 траекторий, 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

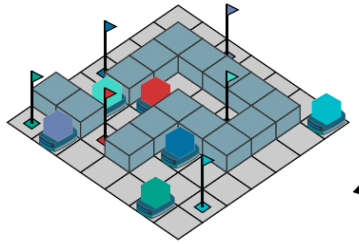
Algorithm 2: Sample transitions from replay buffer

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

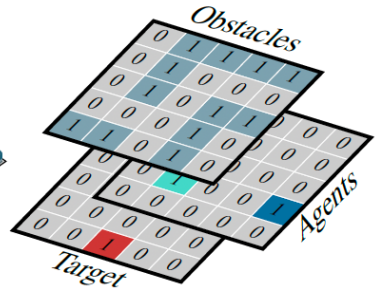
$\mathcal{B} \leftarrow ()$ `// Initialize batch as an empty list`
while $size(\mathcal{B}) < B$ **do**
  $i \sim \mathcal{U}(0, \text{size}(\mathcal{D}) - 1)$ `// Randomly sample starting index of a sequence`
  if $i + m < size(\mathcal{D})$ **then**
    $b \leftarrow \mathcal{D}[i : i+m]$
  else
 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

Borzilov A. et al. Rethinking Exploration and Experience Exploitation in Value-Based Multi-Agent Reinforcement Learning. FRC CSC RAS. Accessl 2025 // https://ieeexplore.ieee.org/document/10844859

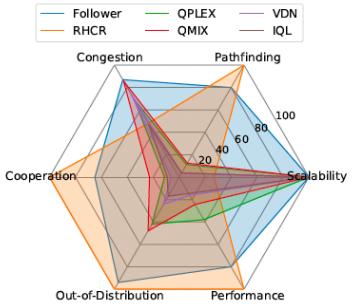
PogemaBench: оценка MAPF алгоритмов



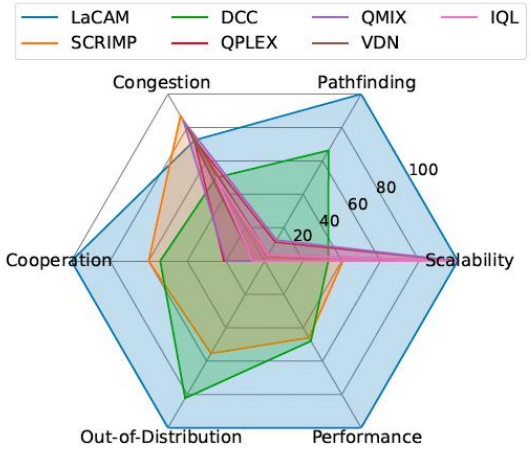
(a) `task`

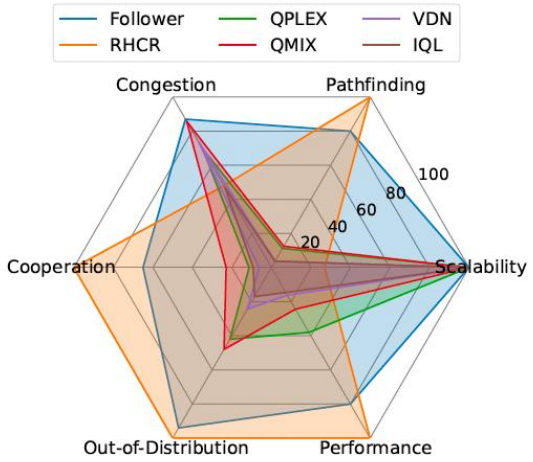


(b) `observation`



(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. AIRI. ICLR 2025 // 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

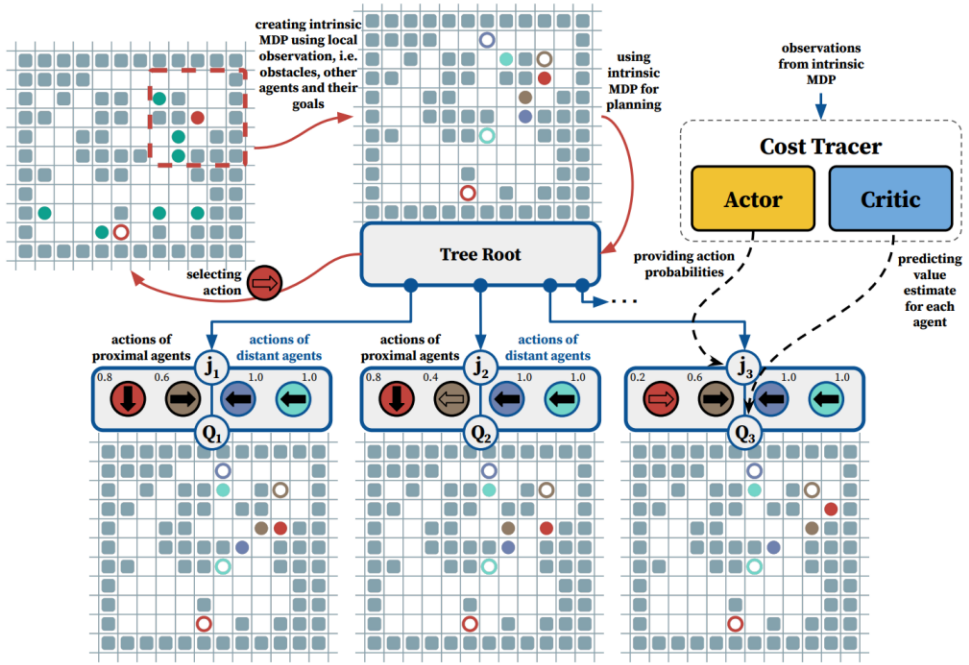
Обучаемые подходы в задаче поиска пути

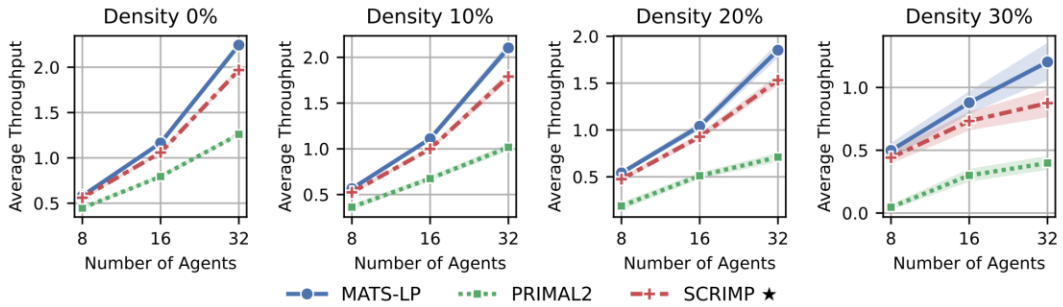
Новые SOTA методы на POGEMA окружении:

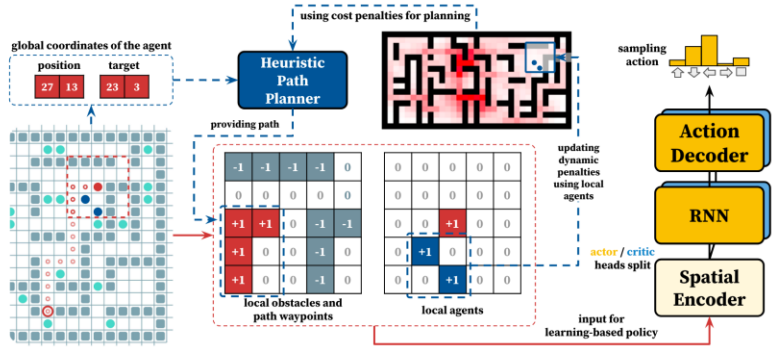
- Switcher – гибридная стратегия с переключением между эвристическим планировщиком и обучаемой распределенной RL-стратегией с памятью

- 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

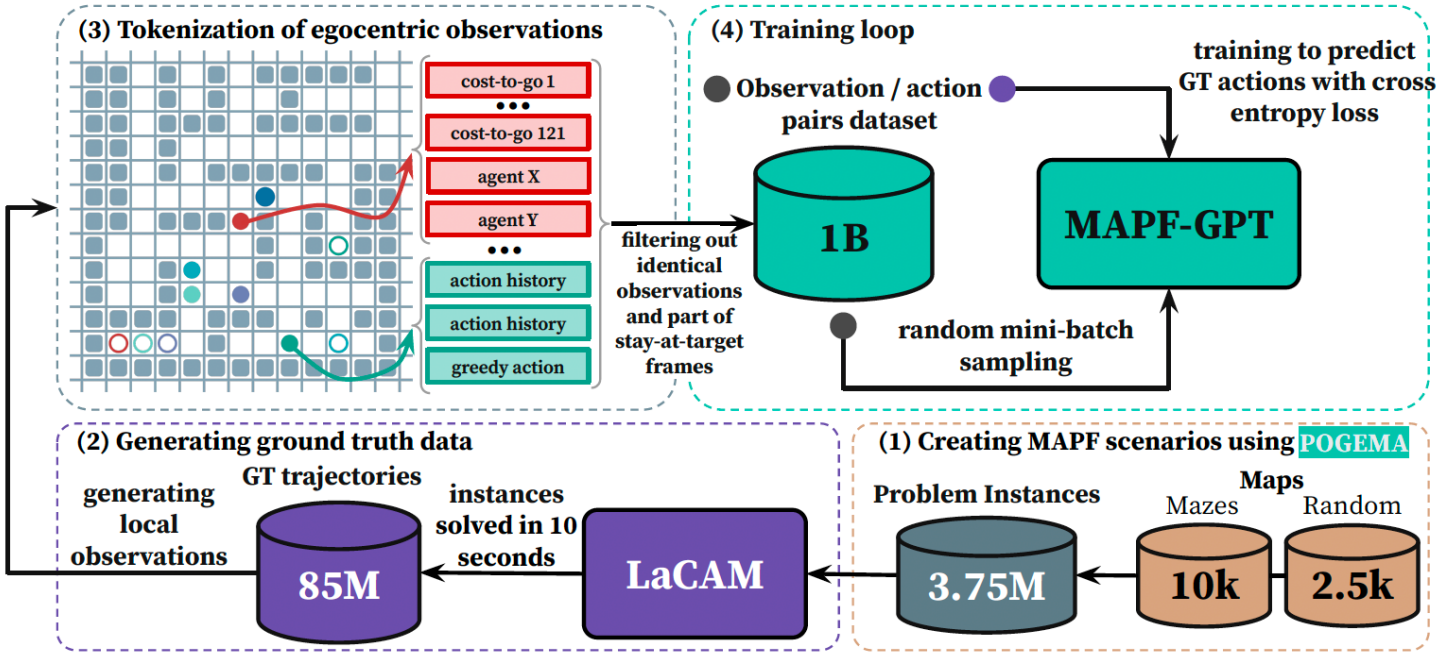


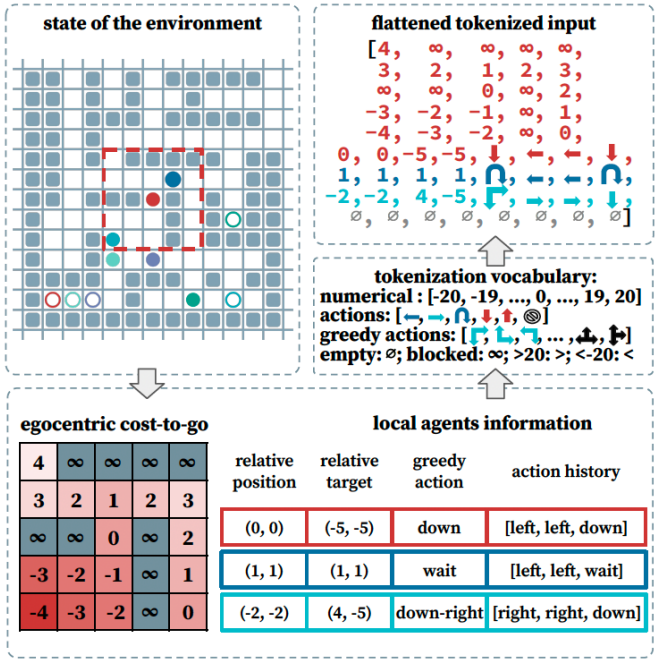


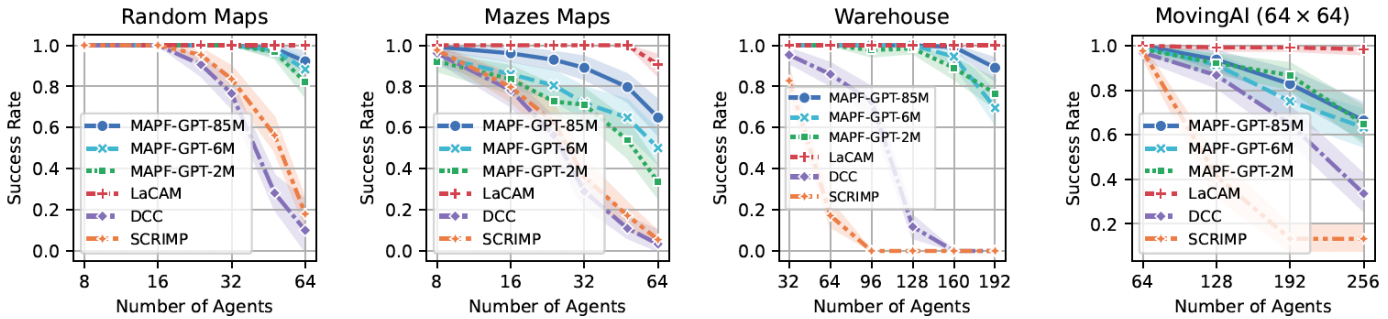


A. Skrynnik et al. Learn to Follow: Decentralized Lifelong Multi-Agent Pathfinding via Planning and Learning. AIRI & MIPT. AAAI 2024.
A. Andreychuk et al. Decentralized Monte Carlo Tree Search for Partially Observable Multi-agent Pathfinding. AIRI. AAAI 2023.

MAPF-GPT: базовая модель для задачи MAPF



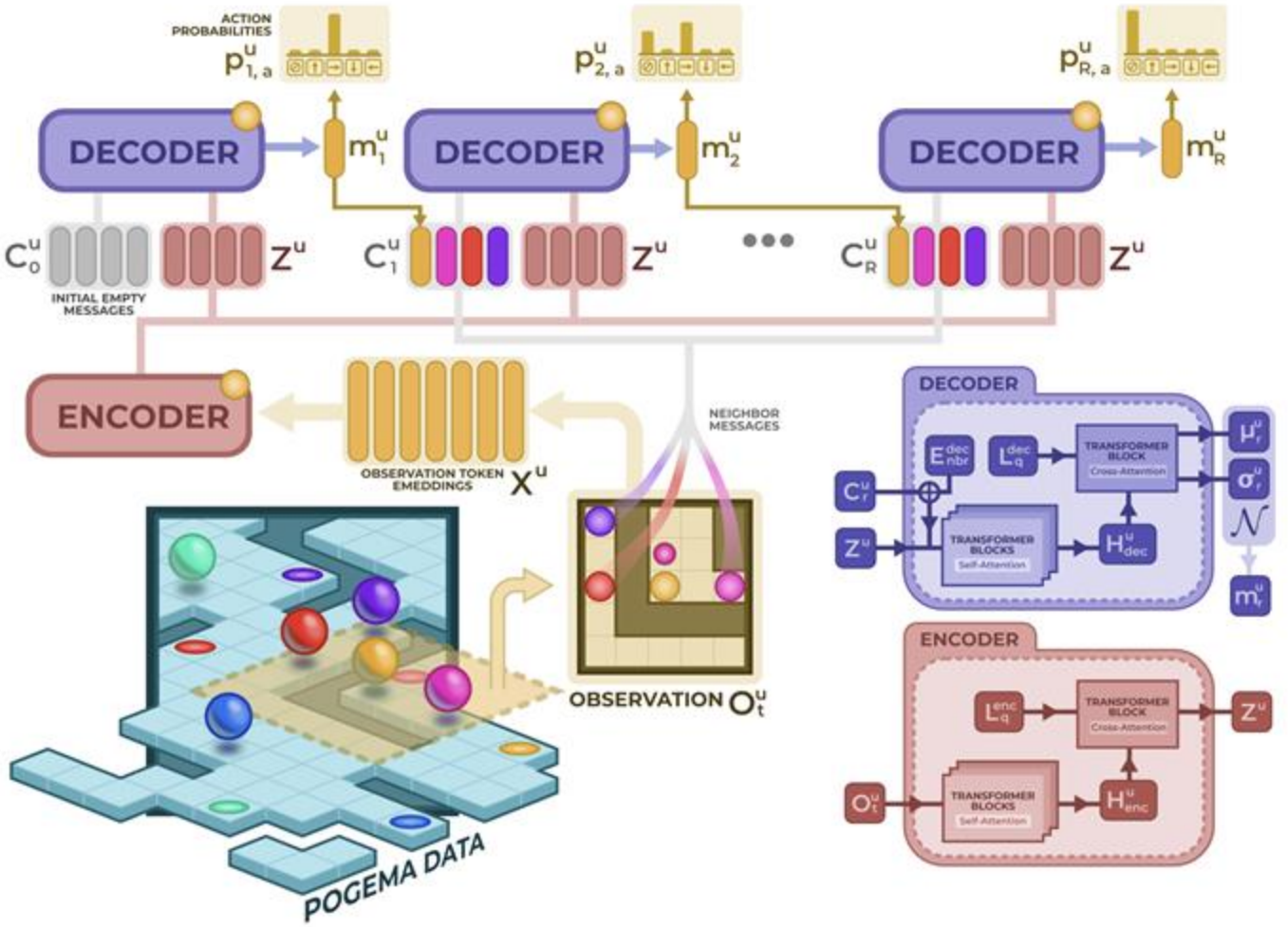




Локальная коммуникация во внутренних представлениях

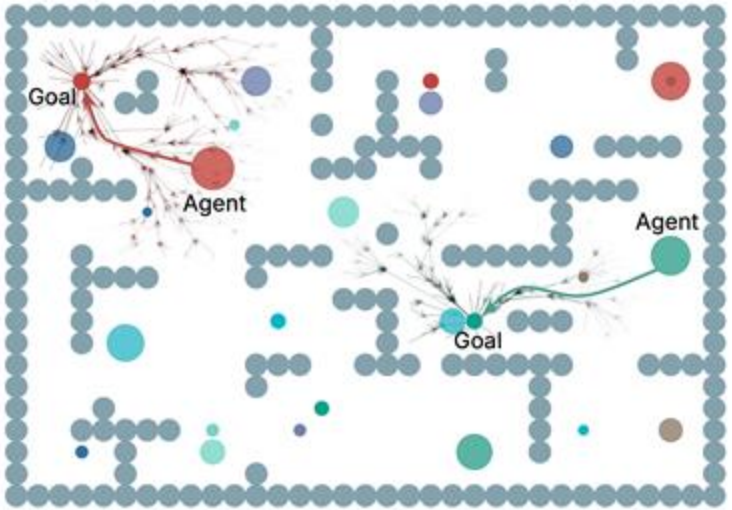
Decentralized Master-Mind

- Encoder / Decoder архитектура, encoder кодирует наблюдение, decoder позволяет агентам общаться в рекуррентном режиме
- Все те же свойства масштабируемости, что и MAPF-GPT, т.к. количество агентов для коммуникации ограничено
- Ряд улучшений из современных трансформерных архитектур (например, из Perceiver)



CAMAR - Continuous Actions Multi-Agent Routing

- Задача мультиагентной навигации в непрерывной постановке и с модифицируемой динамикой агентов
- Эффективная среда, реализованная на JAX
- Набор методов (обучаемых и классических), метрики, визуализация

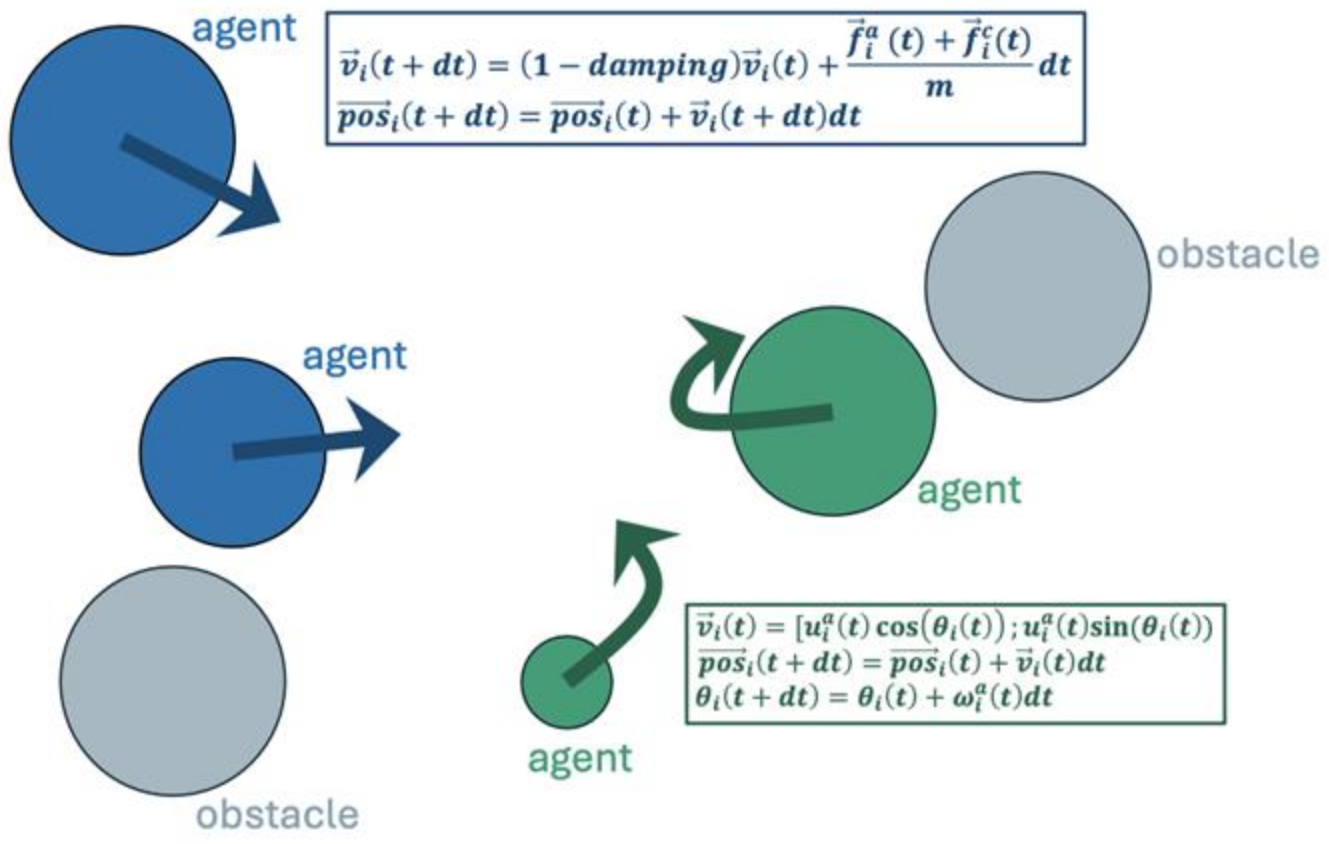


Environment / Simulator	Repository	Cont. Observations	Cont. Actions	GPU Support	Scalability >500 Agents	Partially observable	Heterogeneous agents	Performance >10K SPS	Python based	Procedural generation	Requires generalization	Evaluation protocols	Tests & CI	PyPI Listed
Waterworld (SISL) [24]	link	✓	✓	✗	✗	✓	✗	✓	✓	✗	✗	✗	✓	✓
RWare [11]	link	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓	✓
RWare (Jumanji) [19]	link	✗	✗	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓	✓
RWare (Pufferlib) [25]	link	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗	✓	✓
Trash Pickup (Pufferlib) [25]	link	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗	✓	✓
SMAC [18]	link	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
SMACv2 [15]	link	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗
SMAX (JaxMARL) [17]	link	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓
MPE [20, 16]	link	✓	✓	✗	✓	✓	✓	✗ / ✓ [3]	✓	✗	✓	✗	✓	✓
MPE (JaxMARL) [17]	link	✓	✓	✓	✓	✓	✓	✗ / ✓ [3]	✓	✗	✓	✗	✓	✓
JaxNav (JaxMARL) [26]	link	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓
Nocturne [27]	link	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗
POGEMA [12]	link	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
VMAS [4] [14]	link	✓	✓	✓	✗	✓	✓	✗ / ✓ [4]	✓	✗	✗ / ✓ [4]	✗	✓	✓
SMART [28]	link	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗
Gazebo [21]	link	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗
Webots [22]	link	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗
ARGoS [23]	link	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗
CAMAR (Ours)	link	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Pshenitsyn A. et al. CAMAR: Continuous Actions Multi-Agent Routing. AIRI. AAAI 2026 // http://arxiv.org/abs/2508.12845

Поддержка гетерогенных агентов

- Поддержка режима гетерогенных агентов
- И для размера и для динамики движения (например, HolonomicDynamic или DiffDriveDynamic)
- Агенты действуют совместно в общем пространстве



Pshenitsyn A. et al. CAMAR: Continuous Actions Multi-Agent Routing. AIRI. AAAI 2026 // http://arxiv.org/abs/2508.12845

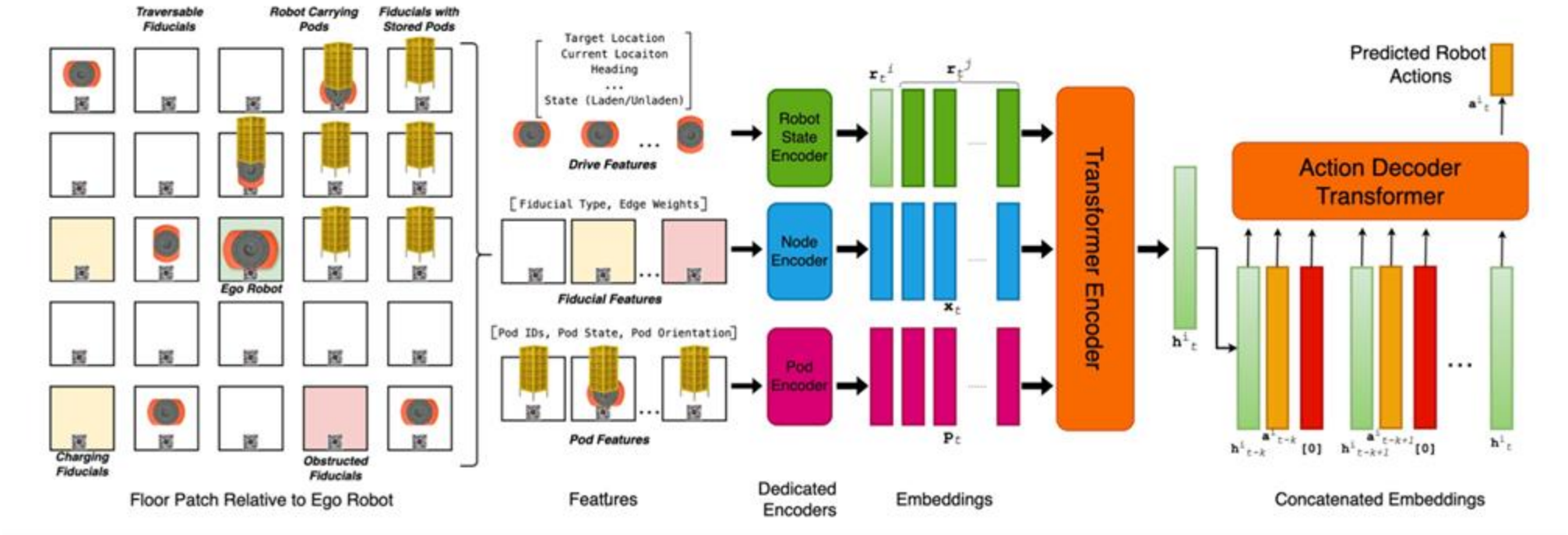
Результаты сравнения

	random_grid				labmaze_grid			
Algorithm	SR ↑	FT ↓	MS ↓	CO ↑	SR ↑	FT ↓	MS ↓	CO ↑
IPPO	0.410±0.001	1695±10	160.0±0.0	1.000±0.000	0.213±0.013	2104±14	160.0±0.0	1.000±0.000
MAPPO	0.830±0.001	984±5	151.4±0.3	1.000±0.000	0.568±0.004	1484±8	160.0±0.0	1.000±0.000
IDDPG	0.335±0.001	1851±10	160.0±0.0	1.000±0.000	0.167±0.000	2772±14	160.0±0.0	0.996±0.000
MADDPG	0.041±0.000	2508±12	160.0±0.0	0.913±0.001	0.027±0.000	2745±12	160.0±0.0	0.854±0.001
ISAC	0.115±0.001	2523±14	160.0±0.0	1.000±0.000	0.047±0.000	2808±12	160.0±0.0	1.000±0.000
MASAC	0.281±0.001	1843±11	160.0±0.0	0.856±0.001	0.105±0.001	2098±12	160.0±0.0	0.781±0.001
RRT*+IPPO	0.420±0.001	1426±9	160.0±0.0	1.000±0.000	0.511±0.001	1316±6	160.0±0.0	0.999±0.000
RRT*+MAPPO	0.828±0.001	971±5	150.4±0.3	1.000±0.000	0.556±0.001	1326±7	160.0±0.0	0.999±0.000
RRT*+IDDPG	0.280±0.001	2181±12	160.0±0.0	1.000±0.000	0.189±0.000	2635±14	160.0±0.0	0.997±0.000
RRT*+MADDPG	0.037±0.000	2953±15	160.1±0.0	0.984±0.000	0.037±0.000	2918±14	160.1±0.0	0.969±0.000
RRT*+ISAC	0.143±0.000	2618±13	160.0±0.0	1.000±0.000	0.058±0.000	2749±13	160.0±0.0	1.000±0.000
RRT*+MASAC	0.054±0.000	2511±14	160.0±0.0	1.000±0.000	0.034±0.000	2854±15	160.0±0.0	0.994±0.000
RRT*+PD	0.678±0.002	2010±59	160.0±0.0	0.997±0.000	0.692±0.004	1807±49	160.0±0.0	0.971±0.002
RRT+PD	0.413±0.014	2440±264	160.0±0.0	0.788±0.021	0.528±0.021	2049±251	160.0±0.0	0.558±0.025

04

Прикладные задачи

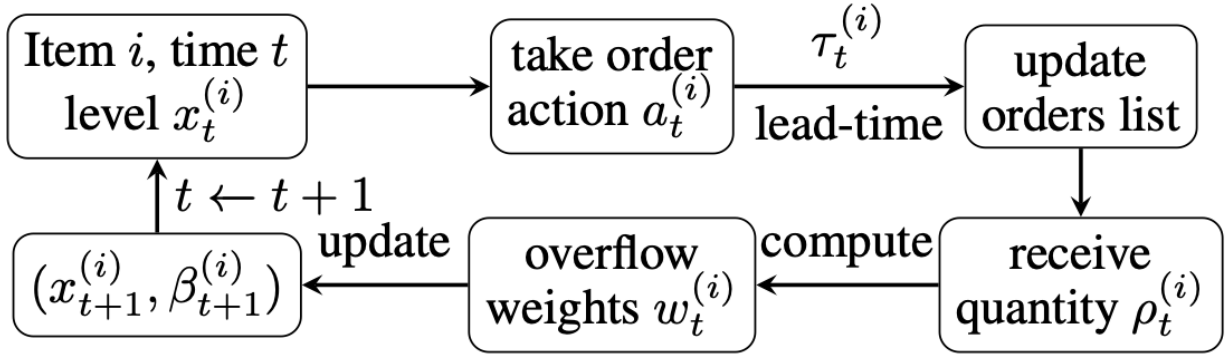
DeepFleet: Multi-Agent Foundation Models for Mobile Robots

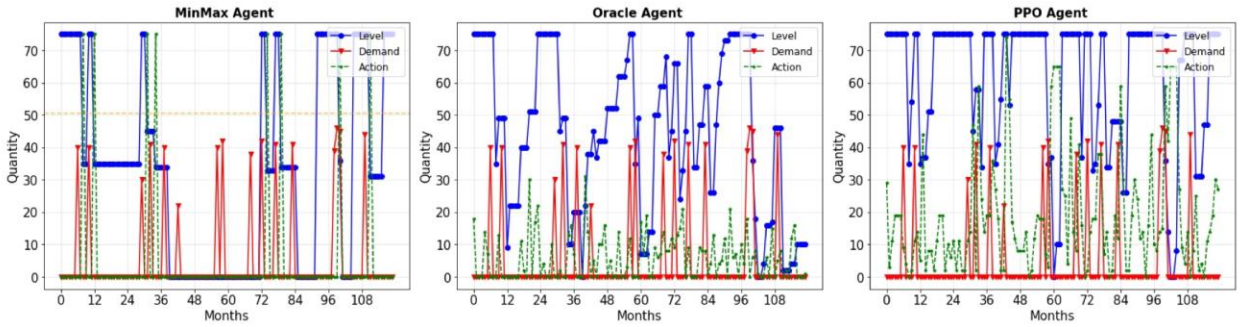


Авторегрессивный трансформер как модель движения роботов с моделью этажа/склада и с графовой моделью временного внимания

DEEPFLEET: Multi-Agent FoundationModels forMobile Robots. Amazon // https://arxiv.org/abs/2508.08574v1

MARL для задачи оптимизации поставки товаров



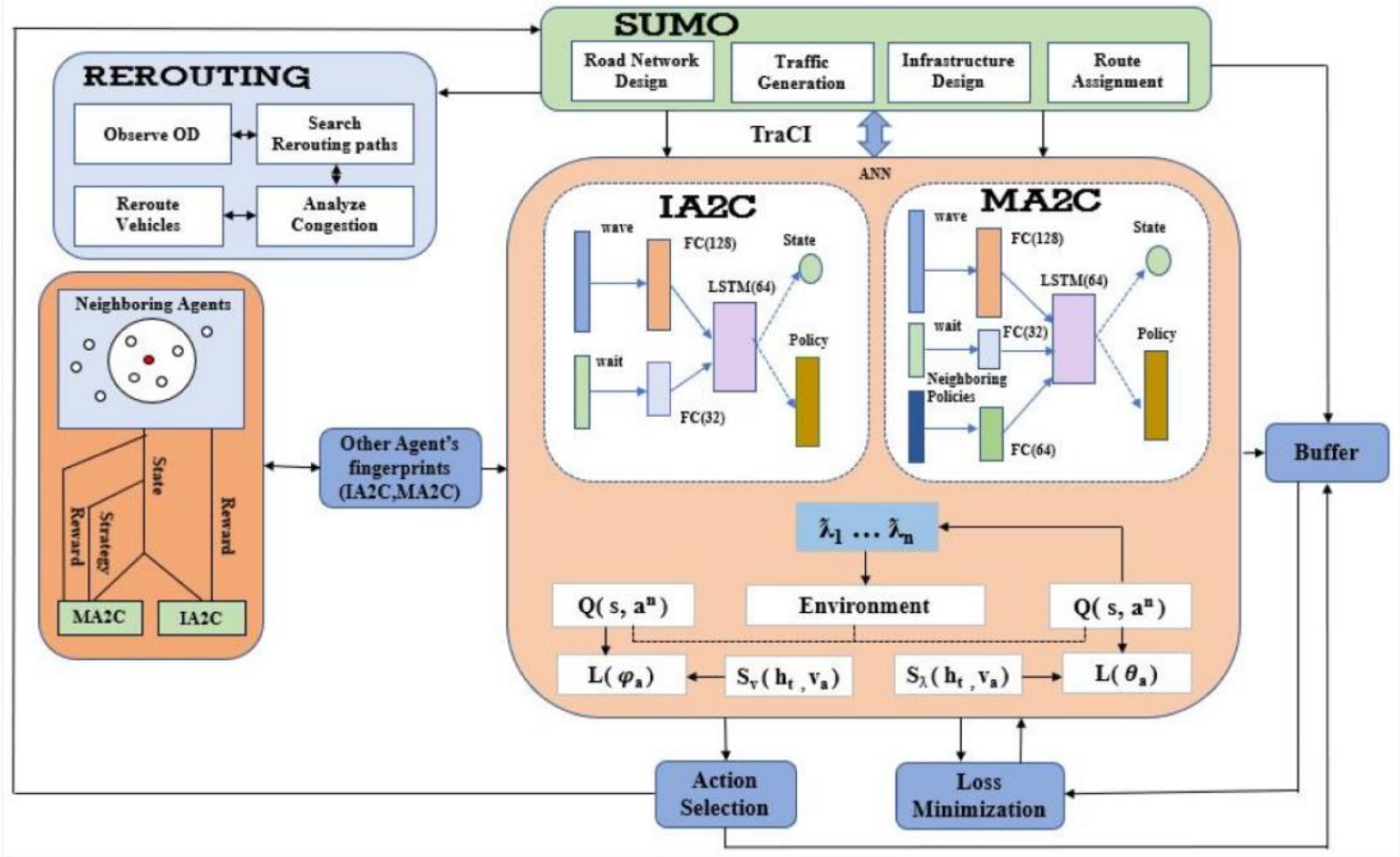


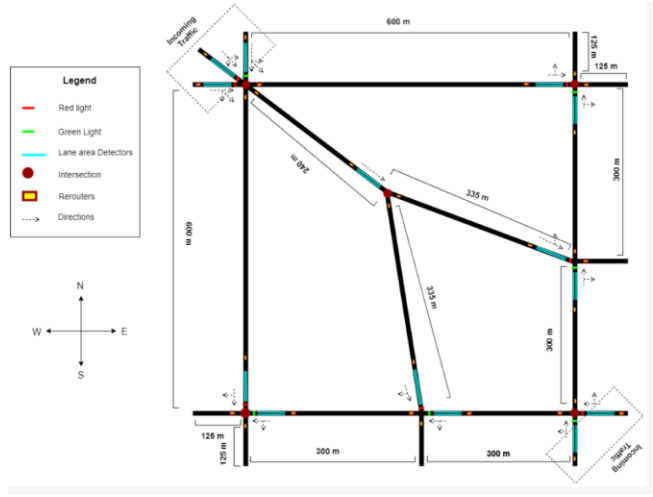
ID	MinMax	Oracle	PPO-D	PPO-C
0	48,554,986	10,183,088	4,863,202	**4,616,016**
1	52,993,931	16,917,389	6,865,991	**6,385,378**
2	70,467,282	21,426,806	8,727,215	**8,087,258**
3	72,220,832	12,722,887	9,854,047	**5,280,345**
4	79,235,272	16,976,630	8,801,628	**4,808,191**

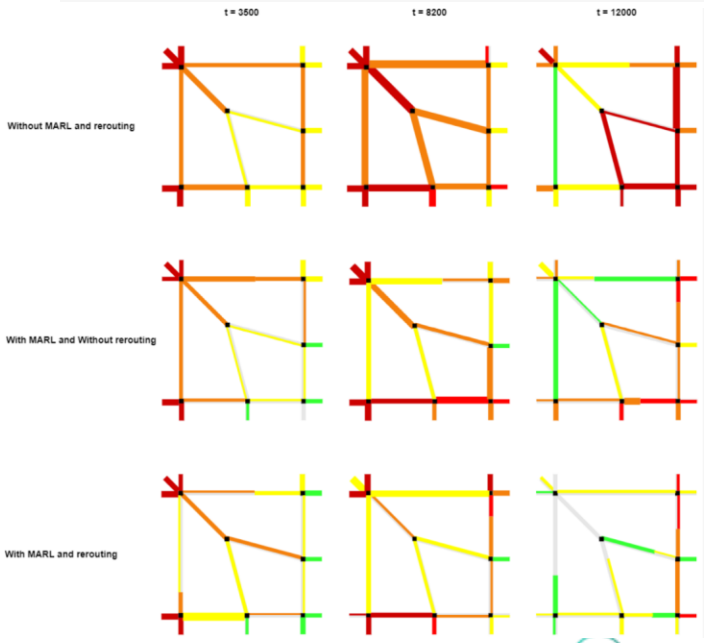
Average Cumulative cost in $ over 100 replications for different items over $T = 240$ months.

MARLIM: Multi-Agent Reinforcement Learning for Inventory Management, 2023

MARL для задачи управления транспортным потоком

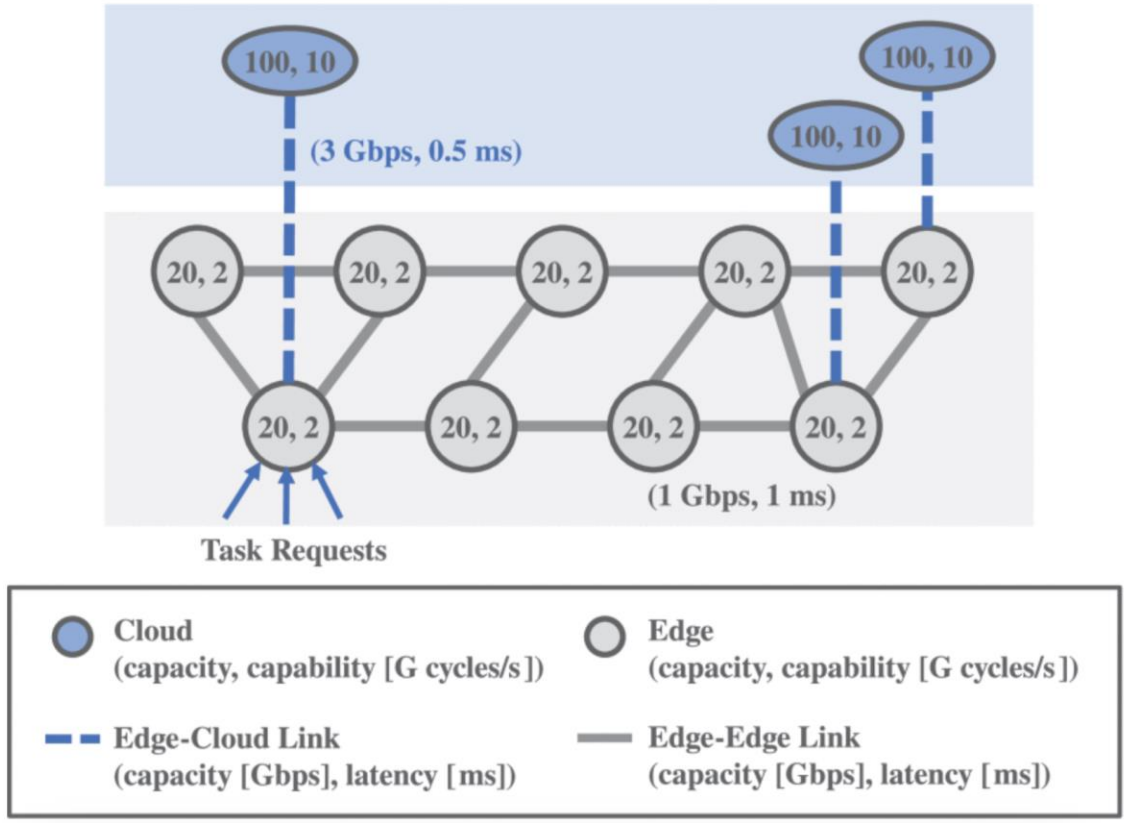


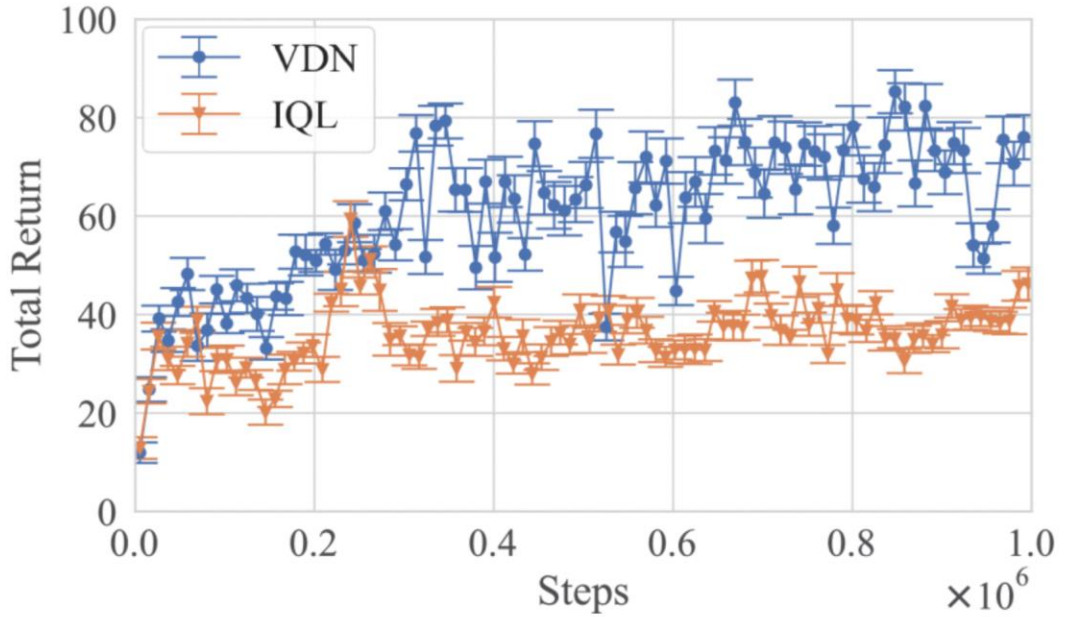


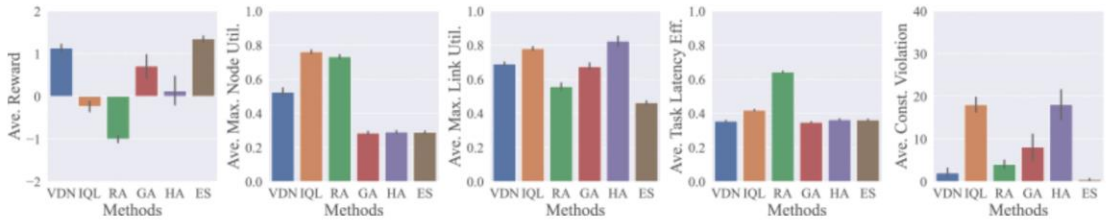


Multi-agent reinforcement learning for traffic flow management of autonomous vehicles, 2023

Оптимизация кооперативных сетевых вычислений







Multi-agent deep reinforcement learning for cooperative computing offloading and route optimization in multi cloud-edge networks, 2024

Итоги и перспективные направления

1. Трансформерные модели благодаря особой структуре памяти хорошо поддаются дообучению
2. Основные режимы дообучения, которые могут использовать в различной комбинации: SFT и RL
3. Для VLM и LLM за обобщение отвечает RL, за запоминание новых данных - SFT
4. Для языковых моделей многообещающими является уплотнение вознаграждения: многошаговый RL, Монте-Карло поиск
5. Для VLM/VLA моделей необходима более тонкая настройка выхода модели и поддержка онлайн режима
6. Challenge: размышляющие мультимодальные модели, действующие в среде

Контакты

Александр Панов

директор лаборатории CAIS AIRI
директор ЦКМ ИИИ МФТИ

panov@airi.net

grafft.github.io

t.me/ai_panov

airi.net

airi_research_institute

AIRI Institute

AIRI Institute

Telegram

Artificial Intelligence
Research Institute