Аннотация

Работа посвящена разработке и исследованию методов моделирования временной памяти в условиях частичной наблюдаемости. Рассмотрены подходы, вдохновлённые нейробиологией: распределённая хеббовская временная память (DHTM) и временное предсказательное кодирование (TPC). Для модели DHTM предложена новая архитектура, основанная на адаптивном росте и объединении дендритных сегментов памяти, что позволило существенно уменьшить их общее количество, повысить устойчивость к шумам и улучшить точность предсказаний.

Адаптирован метод TPC для задач обучения с подкреплением (RL) в частично наблюдаемых средах. TPC реализует нейробиологически мотивированные принципы предсказательного кодирования и формирует компактные и обобщённые латентные представления состояний среды. Экспериментально показано, что предложенная адаптация TPC обеспечивает эффективную генерализацию состояний, что подтверждено анализом латентных представлений, демонстрирующих устойчивость и способность к дифференциации даже в условиях значительной неопределённости наблюдений.

Экспериментальная проверка проводилась в средах Pinball, AnimalAI и Grid-World. Результаты свидетельствуют о высоком потенциале предложенных методов DHTM и TPC в задачах обработки последовательной информации и принятия решений в условиях частичной наблюдаемости.

Оглавление

Стр.

Глава 1. Обзор литературы и теоретические основы 10
1.1 Временная память и частично-наблюдаемые среды 10
1.2 Теоретические и биологические основы временной памяти 11
1.3 Классические модели временной памяти 14
1.3.1 Скрытые марковские модели (HMM) и факторные HMM . 14
1.3.2 Рекуррентные нейронные сети, LSTM, GRU и RWKV . . . 16
1.3.3 Структурированные когнитивные графы (CSCG) 17
1.3.4 Echo State Networks, Liquid State Machines & State Space Models 18
1.4 Иерархическая временная память 21
1.4.1 Биологическая модель 21
1.4.2 Обучение HTM 22
1.5 Предсказательное кодирование и сети Хопфилда 24
1.5.1 Классические сети Хопфилда и их обобщения 24
1.5.2 Предсказательное кодирование: классические и современные модели 26

Глава 2. Распределенная хеббовская временная память (DHTM) 30
2.1 Алгоритм распространения доверия и факторная скрытая марковская модель 30
2.2 Классическая модель 31
2.3 Объединение сегментов 33
2.4 Адаптивное пороговое значение 35

Глава 3. Временное предсказательное кодирование (TPC) 37
3.1 Предсказательное кодирование и временные представления в RL 37
3.2 Временное предсказательное кодирование в RL-сценарии 38

Глава 4. Эксперименты 42
4.1 Распределенная Хеббовская временная память (DHTM) 42
4.1.1 Среды и сценарий обучения 42

4.1.2 Результаты в средах Pinball и AnimalAI 43
4.1.3 Сравнение с другими методами 44
4.2 Временное предсказательное кодирование (TPC) 46
4.2.1 Точность предсказаний в случайных траекториях 47
4.2.2 Точность предсказаний в RL-сценарии 48
4.2.3 Сравнение с другими моделями в RL 48
4.2.4 Кластеризация латентных представлений и обобщающая способность 49

Заключение 52

Список литературы 54

Введение

Моделирование памяти и обработка последовательной информации – одна из ключевых задач в искусственном интеллекте и когнитивных науках [1]. Способность системы запоминать прошлый опыт и делать прогнозы – неотъемлемое свойство интеллекта, необходимое для принятия решений во времени, понимания естественного языка, анализа временных рядов и других задач [2–4], где данные поступают последовательно. При этом особый интерес представляет временная память – механизм, позволяющий эффективно кодировать и извлекать информацию о последовательностях событий. В человеческом мозге память реализуется с поразительной эффективностью: мы можем онлайн, то есть в режиме реального времени, учиться на единичном опыте [5], формировать представления о скрытых причинах наблюдаемых явлений и делать долгосрочные прогнозы [1]. Для искусственных систем подобные способности остаются серьезным вызовом.

Актуальность проблемы. Современные алгоритмы машинного обучения добились больших успехов в решении отдельных задач, однако модель памяти, сопоставимая по гибкости и надежности с биологической, до сих пор не создана. В классических подходах к моделированию последовательностей широко применяются рекуррентные нейронные сети (RNN) [6] и их варианты, например LSTM и GRU [7]. Эти модели способны аппроксимировать произвольные динамики, но опираются на алгоритм обратного распространения ошибок во времени (BPTT) для обучения, что влечет ряд проблем [8]. Обратное распространение через длинные временные последовательности страдает от нестабильности градиентов (затухание или взрыв), требуя специальных методов для частичного решения, таких как усеченные BPTT или более устойчивые архитектуры (например, RWKV или Linear Recurrent Unit [9]). Кроме того, RNN-модели обычно обучаются офлайн на полных датасетах и с трудом адаптируются онлайн к нестационарным данным. Это затрудняет их применение в сценариях, где данные поступают потоково, а распределение может со временем меняться (например, при обучении агента в реальном времени). Ещё один недостаток стандартных RNN – отсутствие явной вероятностной интерпретации: в отличие от статистических моделей, таких как скрытые марковские модели (HMM) [10],

нейросети не предоставляют напрямую меры неопределенности своих предсказаний. Это осложняет их использование в условиях стохастической динамики, где для эффективного планирования требуется оценка неопределенности состояния.

Другой важный класс задач, остро демонстрирующий ограничения современных моделей памяти, – это задачи обучения с подкреплением (reinforcement learning, RL) в условиях частичной наблюдаемости. В таких задачах агент получает последовательность наблюдений, которые не однозначно определяют состояние среды; формально среда моделируется как частично наблюдаемый марковский процесс принятия решений (POMDP) [11]. Решение POMDP требует от агента поддерживать внутреннюю память или belief state – распределение вероятностей по возможным скрытым состояниям – на основе всех полученных наблюдений и действий. Классические методы решения POMDP опираются на вычислительно затратное отслеживание belief-state или планирование на пространстве состояний, рост которого экспоненциален от длины истории [12]. В практических RL алгоритмах частичная наблюдаемость обычно реализуется с помощью тех же рекуррентных нейросетей (например, добавлением LSTM к агенту для запоминания предыдущих наблюдений) [13]. Однако, как отмечалось выше, обучение памяти через backpropagation затруднительно и не гарантирует надежного поведения вне распределения тренировочных данных. Кроме того, в онлайн режиме классические RL-агенты испытывают проблемы с катастрофическим забыванием и нестабильностью обучения, так как распределение данных меняется по мере взаимодействия агента со средой [14].

В то же время, биологические системы демонстрируют выдающиеся способности к обучению в POMDP-ситуациях [15]. Животные и люди могут из неполных и шумных сенсорных потоков быстро извлекать существенные признаки, формировать внутренние представления окружающего мира и обновлять их при поступлении новой информации. Мозг, очевидно, решает задачу скрытого состояния более эффективно, чем современные алгоритмы. Это вдохновляет исследователей разрабатывать биологически правдоподобные подходы к моделированию памяти – такие, которые опираются на известные нейрофизиологические принципы работы мозга. Предполагается, что алгоритмы, учитывающие архитектуру и правила обучения реального мозга (такие как локальные хеббовские правила пластичности [16;17], кодирование информации в виде рас-

пределенных разреженных паттернов активности [18], спайковая динамика нейронов и т.д.), могут обладать лучшей стабильностью, способностью к непрерывному обучению и энергоэффективностью, присущей биологическим системам. Например, локальные хеббовские правила обучения, аналогичные механизмам синаптической пластичности в мозге, не требуют глобального дифференцируемого сигнала ошибки и могут работать в онлайн-режиме, обновляя память при поступлении нового опыта [17]. Также считается, что разделение нейронных входов на сегрегированные дендритные компоненты позволяет биологическим нейронам обрабатывать контекстные сигналы [19] (входы от других нейронов, связанные с предшествующей активностью) отдельно от прямых сенсорных входов. Такие механизмы потенциально повышают способность сети предсказывать будущие события на основе последовательностей стимулов [20].

Постановка цели и задач. Настоящая работа посвящена развитию методов моделирования временной памяти, объединяющих вероятностный формализм и принципы, вдохновленные нейробиологией, для применения в задачах обучения с подкреплением при частично наблюдаемых состояниях. Цель исследования – разработать, улучшить и исследовать новые модели памяти, способные к непрерывному обучению в динамических средах, обладающие биологической правдоподобностью и улучшенные по сравнению с существующими подходами характеристики эффективности. Для достижения поставленной цели необходимо решить следующие задачи:

- **Анализ литературы и теории.** Изучить современное состояние исследований в области моделирования временной памяти и представления состояний в условиях частичной наблюдаемости. Особое внимание уделить вероятностным моделям последовательностей (HMM, факторным HMM и алгоритмам belief propogation), нейровдохновленным моделям памяти (Hierarchical Temporal Memory [21], предсказательное кодирование и др. [22;23]), а также работам по биологически правдоподобному обучению агентов в POMDP-средах (online learning, active inference и т.п.). На основе анализа выявить ограничения существующих подходов и обосновать выбор архитектурных решений для разрабатываемых моделей.
- **Усовершенствование архитектуры DHTM.** Для модели DHTM [24], на основе идей иерархической временной памяти (HTM) и фак-

торных скрытых марковских моделей [25], спроектировать гибридную логику роста и использования дендридных сегментов с целью улучшения биологической правдоподобности и решения проблем связанных с чрезмерным ростом сегментов, ухудшением качества.

– **Адаптация алгоритмов временного предсказательного кодирования [26] (TPC) для POMDP в RL среде.** Исследовать возможность применения принципов предсказательного кодирования (Predictive Coding, PC) для формирования внутреннего представления RL агента в частично-наблюдаемой среде. Разработать архитектуру, в которой механизм Temporal Predictive Coding (TPC) интегрирован в алгоритм обучения агента в POMDP RL среде с предположением, что такой агент будет выучивать скрытую модель динамики среды.
– **Экспериментальная проверка и сравнение.** Реализовать предложенные модели и провести серию экспериментов в средах с частичной наблюдаемостью. Для DHTM – протестировать запоминающую способность модели в средах AnimalAI и Pinball с различными архитектурными реализациями и сравнить предложенный метод роста сегментов с классическими подходами. Для агента с TPC – оценить качество формирования скрытых состояний в частично-наблюдаемых средах и итоговую эффективность в сравнении с базовыми моделями в для данной задачи (LSTM [7], RWKV [9], CSCG [27], EC-agent [28]). Проанализировать, в какой мере био-правдоподобные свойства моделей (локальность обучения, разреженность, непрерывное обновление) способствуют улучшению показателей.

Научная новизна. В рамках работы предложены два оригинальных подхода к моделированию временной памяти и состояния агента.

Первый – усовершенствованная модель DHTM, представляющая собой новый вариант вероятностной эпизодической памяти, сочетающей графовую модель (фактор-граф аналогичный Factorial HMM) с нейронной реализацией в виде HTM-подобных сетей. Усовершенствования касаются логики роста и использования дендридных сегментов в процессе обучения DHTM.

Второй – интеграция принципов предсказательного кодирования в контур обучения с подкреплением на основе модели мира. Предложен вариант RL-агента, использующего TPC для оценки скрытого состояния в POMDP. Здесь па-

мять основана на минимизации локальных ошибок предсказания, что ближе к нейрофизиологическим теориям работы коры [29]. Ожидается, что такая архитектура позволит агенту эффективно извлекать структуру среды из наблюдений и сохранять устойчивость обучения в условиях частичной наблюдаемости. Так же проводится анализ и сравнение скрытого представления TPC в такой среде.

Практическая значимость. Разработанные подходы могут найти применение в системах искусственного интеллекта, требующих надежной работы в реальном времени и адаптации к меняющейся среде. Например, алгоритмы потенциально пригодны для управления роботами и автономными агентами, которым необходимо быстро учиться на опыте, запоминать новые объекты и последовательности действий в динамической обстановке (как это делает биологический гиппокамп [30]) и при этом избегать катастрофического забывания прошлых навыков. Благодаря вероятностной природе представления, такие алгоритмы могут улучшить планирование и принятие решений в условиях неопределенности, обеспечивая оценку доверия к своим предсказаниям. Интеграция таких подходов способна повысить семантическую интерпретируемость внутренних состояний агента и его устойчивость к сенсорному шуму: агент, предсказывающий сенсорный поток, способен обнаруживать аномалии и новизну во входных данных, что важно для безопасного применения ИИ (например, в автономном вождении). Кроме того, биологически правдоподобные алгоритмы лучше подходят для аппаратной реализации на нейроморфных ускорителях, воспроизводящих спайковую динамику нейронов [31]. В целом, результаты работы способствуют сближению нейронауки и глубокого обучения: используя подсмотренные у мозга принципы, можно строить более эффективные и интерпретируемые системы ИИ, а с другой стороны – проверять нейрофизиологические теории с помощью компьютерных моделей. В следующей главе представлен подробный обзор литературы и теоретических основ, на которых базируется проведенное исследование.

Глава 1. Обзор литературы и теоретические основы

1.1 Временная память и частично-наблюдаемые среды

Память в самом общем виде можно определить как способность системы сохранять информацию о прошлых состояниях и событиях и использовать эту информацию для влияния на поведение в будущем. В контексте интеллектуальных систем особую роль играет временная память – механизм, который отвечает за кодирование последовательностей. Если статическая память (ассоциативная или семантическая) хранит связи между одновременными или неупорядоченными фактами, то временная – запечатлевает порядок возникновения событий во времени. Биологические исследования показывают, что в мозге существуют специализированные структуры для различного вида памяти. В частности, гиппокамп и связанная с ним система вовлечены в быстрое запоминание эпизодов – конкретных последовательностей пережитых событий, привязанных к контексту и времени. В гиппокампе обнаружены нейроны, избирательно активирующиеся в определенные моменты времени в ходе последовательности (так называемые «time cells») [32], что указывает на его роль в хронометрировании событий и формировании временных ассоциаций. Неокортекс, напротив, обеспечивает более медленное статистическое обучение – извлечение общих шаблонов и последовательных закономерностей из многократного опыта. Предполагается, что кортикальные нейронные сети обучаются предсказывать входные сигналы на разных иерархических уровнях, постепенно формируя модель мира.

С точки зрения искусственного интеллекта, память необходима прежде всего там, где данные частично наблюдаемы. В полностью наблюдаемом случае (как, например, в классических играх типа шахмат) оптимальное решение может быть принято, опираясь только на текущее состояние — предыдущие шаги не влияют на будущее, кроме как через настоящее. Но если среда частично наблюдаема или нестационарна, агенту жизненно важно учитывать историю. Частичная наблюдаемость формально описывается моделью POMDP [11; 12], где на каждом шаге t агент получает наблюдение o_t, стохастически зависящее от истинного скрытого состояния s_t, и совершает действие a_t, влияющее на пе-

реход состояния к s_{t+1}. Награда r_t может зависеть от (s_t, a_t). Задача агента — максимизировать суммарную ожидаемую награду, не имея непосредственного доступа к s_t. Классическим подходом является обобщение принципа оптимальности Беллмана на случай частичной информации: вместо s_t рассматривается belief $b_t = P(s_t \mid o_1, \ldots, o_t, a_1, \ldots, a_{t-1})$ — апостериорное распределение по состояниям, полученное путем байесовской фильтрации всего опыта. Этот b_t и является внутренней памятью, агрегирующей исторические данные. Однако точное вычисление и обновление belief-состояния даже при известных моделях переходов и наблюдений является ресурсно тяжёлым (размер пространства состояний может быть огромным, а belief — бесконечным семейством распределений). Поэтому в практических задачах прибегают к приближённым методам: моделированию последовательностей наблюдений [13]. Иными словами, вместо явного вычисления распределения $P(s_t \mid \text{history})$ агент строит некоторую функцию:

$$h_t = f_\theta(h_{t-1}, o_t, a_{t-1}),$$

где h_t — вектор внутреннего состояния (эмбеддинг памяти), обновляемый рекуррентно на каждом шаге.

1.2 Теоретические и биологические основы временной памяти

Биологически-правдоподобная временная память – способность нейронных систем кодировать и воспроизводить последовательности событий во времени – опирается на ряд нейрофизиологических механизмов. Разреженные распределённые представления [18] (Sparse Distributed Representations) являются ключевым принципом: информация кодируется активностью небольшой доли нейронов из большой популяции. Разреженное кодирование повышает ёмкость памяти и устойчивость к шуму, поскольку перекрытие активных множеств нейронов минимально. Вероятность случайного совпадения паттерна при разрежённости крайне мала – например, если активно лишь $a = 1\%$ нейронов из n общего числа, то совпадение по θ синапсам из s возможных на дендритном сегменте имеет малую вероятность, оцениваемую комбинаторной формулой (см. Eq. 1.1 ниже). В контексте памяти это означает, что нейронные ансамбли могут

однозначно представлять множество различных состояний/воспоминаний без взаимных помех.

Другим принципом является функциональная специализация дендритных сегментов. Возбуждающие синапсы пирамидального нейрона распределены по ветвям дендрита, образуя локальные кластеры. Каждая небольшая группа (порядка 10-20 соседних синапсов) способна порождать NMDA-спайк – локальное дендритное возбуждение – если все эти синапсы одновременно активны. Таким образом, отдельный сегмент дендрита выступает как совпадение-шаблон (coincidence detector), распознающий специфический паттерн входов. Один нейрон с тысячами синапсов фактически способен хранить сотни независимых паттернов – каждый сегмент отвечает за распознавание своего шаблона входной активности. Выходной аксон нейрона активируется (генерируется потенциал действия) если достаточное число сегментов возбудились вместе (суммарная деполяризация достигает порога). Благодаря этому механизму один нейрон может быть одновременно частью множества временных паттернов, а активация конкретного набора его сегментов сигнализирует о распознанной последовательности входов.

Критически важна организация неокортекса в виде кортикальных колонок (миниколонок). Миниколонка – это вертикальное объединение 50-100 нейронов, получающих общие входные сигналы. Предполагается, что в памяти последовательностей миниколонка кодирует узнавание определённого входного признака, тогда как разные нейроны внутри колонки отвечают за различные контексты этого признака во временной последовательности. В каждом моменте времени из каждой миниколонки активен не более одного нейрона (или небольшой процент), что обеспечивает разреженное представление. Такая организация соответствует наблюдаемым свойствам коры: нейроны в колонке имеют схожие рецептивные поля, но дифференцируются по контексту предыдущих состояний.

Правила пластичности играют центральную роль в формировании временной памяти [16; 17]. Классическое правило хеббовского обучения гласит: “нейроны, активирующиеся вместе, образуют связь” – синаптическая эффективность усиливается, если пресинаптический и постсинаптический нейрон активны совместно. Временная версия этого принципа, Spike-Timing-Dependent Plasticity (STDP), учитывает порядок активации: если пресинаптический нейрон активен незадолго до постсинаптического, синапс усиливается, а при обрат-

ном порядке – ослабляется. STDP обеспечивает связывание событий во времени: синапсы “запоминают”, какое предшествующее состояние предвосхищало текущее, что критично для последовательной памяти. В моделях временной памяти часто используются локальные правила обучения, имитирующие STDP или его обобщения, для настройки связей между нейронами в последовательностях.

Наконец, различные структуры мозга демонстрируют специализированные нейроны, участвующие в кодировании времени [30]. В гиппокампе обнаружены “временные клетки”, которые последовательно активируются при прохождении животным определённого временного интервала, независимо от конкретных внешних сигналов. Эти клетки дополняют известные “клетки места” и указывают, что гиппокамп хранит не только пространственные, но и временные паттерны опыта. В неокортексе, вероятно, аналоги временных клеток проявляются через описанный механизм контекстно-зависимой активации нейронов в колонках. Таким образом, биологические основы памяти последовательностей включают разреженное представление, активные дендриты, локальные правила пластичности и специализированные нейронные ансамбли, обеспечивающие прогнозирование и узнавание событий во времени.

$$P_{\text{false match}} = \frac{\sum_{i=\theta}^{s} C_s^i \cdot C_{n-s}^{a-i}}{C_n^a} \tag{1.1}$$

где n – общее число нейронов, a – число активных нейронов (размер паттерна, $a \ll n$), s – число синапсов на дендритном сегменте, и θ – порог срабатывания сегмента (минимальное число совпавших синапсов). Эта формула задаёт вероятность того, что сегмент ложно “распознает” случайный паттерн ($\geq \theta$ из s синапсов совпадут с активными нейронами паттерна). При разреженных паттернах эта вероятность чрезвычайно мала (например, $\sim 10^{-20}$ при $n = 200000$, $a = 2000$, $s = 10$, $\theta = 10$), что обеспечивает надёжность такой памяти.

1.3 Классические модели временной памяти

В вычислительной нейронауке и машинном обучении разработан ряд моделей, способных сохранять и предсказывать последовательности. Классическими подходами являются как вероятностные графовые модели, так и нейросетевые архитектуры. Ниже приводится обзор основных моделей временной памяти, их формализм и свойства.

1.3.1 Скрытые марковские модели (HMM) и факторные HMM

Исторически задачу моделирования последовательностей стали решать сначала методами статистического обучения — в частности, с помощью скрытых марковских моделей (Hidden Markov Models, HMM [10; 33]). HMM предполагает, что динамика скрытого состояния является марковским процессом:

$$P(s_{t+1} \mid s_t, \ldots, s_1) = P(s_{t+1} \mid s_t).$$

Наблюдения при этом независимы и порождаются скрытым состоянием:

$$P(o_t \mid s_{1:t}, o_{1:t-1}) = P(o_t \mid s_t).$$

Структура HMM может быть представлена ориентированным графом (динамической байесовской сетью), где узлы скрытых переменных s_t образуют цепь, а от каждого s_t исходит ребро к наблюдению o_t. Для вычисления апостериорного распределения по состояниям на основе последовательности наблюдений применяется алгоритм прямого–обратного прохода, он же алгоритм Баума–Вельша, он же частный случай общего алгоритма суммы-произведения на фактор-графе. Суть его состоит в рекуррентном вычислении прямых сообщений

$$\alpha_t(s_t) = P(o_1, \ldots, o_t, s_t),$$

и обратных сообщений

$$\beta_t(s_t) = P(o_{t+1}, \ldots, o_T \mid s_t),$$

которые позволяют найти оценку:

$$P(s_t \mid o_{1:T}) \propto \alpha_t(s_t) \cdot \beta_t(s_t).$$

Обучение параметров HMM (матрицы переходов состояний и вероятности эмиссий наблюдений) классически осуществляется через EM-алгоритм, чередующий:

- **E-шаг** — вычисление ожидаемых достаточных статистик посредством упомянутого алгоритма;
- **M-шаг** — пересчёт параметров по максимуму правдоподобия.

HMM позволяет интерпретировать результаты в терминах вероятностей и оптимальности. Однако ее ограниченность — в относительно малом числе скрытых состояний и стационарной динамике. Если процесс, генерирующий последовательность, сложен (например, сочетает несколько независимых факторов влияния), классическая HMM становится неэффективной: число состояний потребуется экспоненциально большое, чтобы «зашить» все комбинации факторов.

Одним из решений стала разработка факторизованных HMM [25] (Factorial HMM). В Factorial HMM предполагается, что скрытое состояние на каждом шаге состоит из нескольких независимых компонент:

$$s_t = \left(s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(K)}\right),$$

каждая из которых эволюционирует как отдельная марковская цепь. Наблюдение o_t зависит одновременно от всех компонент $s_t^{(1:K)}$. Такая факторизация позволяет экспоненциально сократить число состояний, представив общее состояние как **декартово произведение факторов**. Графовая структура Factorial HMM — это несколько скрытых цепей, сходящихся к одному наблюдению в каждый момент времени. Однако факторизация усложняет инференс: прямой–обратный алгоритм становится вычислительно затратным, поскольку компоненты не полностью независимы (они связываются общим наблюдением). Тем не менее, алгоритмы на основе суммы-произведения могут быть применены и к

факторным моделям: распространения сообщений проходят по более сложному графу с перекрёстными связями между факторными цепями через узлы наблюдений. В литературе показано, что Factorial HMM способен выучивать распределённые представления и выявлять скрытые факторы в данных, но практическое применение ограничено из-за сложности обучения (EM в большом пространстве параметров, необходимость большого объёма данных). Также стоит отметить, что классические HMM, будучи статистически обоснованными, не сразу приспособлены к режиму онлайн обучения: стандартный EM требует нескольких проходов по датасету, хотя существуют его стохастические модификации.

1.3.2 Рекуррентные нейронные сети, LSTM, GRU и RWKV

Альтернативным направлением, доминирующим в последние десятилетия, стало использование рекуррентных нейронных сетей (RNN) для моделирования последовательностей [6; 7; 34]. RNN по сути учит функцию перехода:

$$h_t = f_\theta(h_{t-1}, o_t),$$

где h_t — скрытое состояние сети (не имеющее прямой вероятностной интерпретации, но служащее «памятью»). Рекуррентные сети с достаточным числом нейронов являются универсальными аппроксиматорами динамических систем, поэтому смогли достичь успехов во многих задачах (распознавание речи, машинный перевод и др.). Однако, как отмечалось во введении, RNN страдают от проблем обучения. Архитектуры с gating-механизмами (LSTM, GRU) смягчают эти проблемы, вводя отдельные параметры для сохранения долгосрочной информации (через «ячейки» и гейты забывания/считывания). Тем не менее, даже LSTM при очень длинных зависимостях (сотни и тысячи шагов) испытывают трудности; к тому же для их успешного обучения требуется тщательно подбирать шаг обучения и использовать большие размеченные последовательные датасеты. В последние годы популярность приобрели модели-трансформеры, которые в задачах последовательностей вытесняют RNN, однако и они применимы в основном в режиме оффлайн-обучения (обучение на большом статичном корпусе с последующим применением) и не решают проблем частич-

ной наблюдаемости per se. Интересно, что на фоне успехов трансформеров начался обратный процесс «возрождения» RNN: появляются архитектуры вроде RWKV [9] (Receptance Weighted Key-Value). Это попытка объединить сильные стороны трансформеров (эффективное обучение на длинных последовательностях за счёт параллелизма) и RNN (эффективная авторегрессивная генерация без квадратичной сложности). В модели RWKV вводятся понятия “ключей” и “значений”, аналогично механизмам внимания, но вычисляемых рекуррентно. Каждый временной шаг обновляет скрытое состояние с помощью линейного внимания: вычисляются взвешенные суммы “значений” с весами по сходству “ключей”, однако в отличие от классического self-attention, это выполняется в рекуррентной форме, без явного квадратичного сравнения всех пар.

1.3.3 Структурированные когнитивные графы (CSCG)

Графовые модели последовательной памяти представлены не только цепочками Маркова, но и более сложными структурами. Структурированный когнитивный граф (Clone-Structured Cognitive Graph, CSCG) [27] – сравнительно новая модель, предложенная для описания формирования когнитивных карт в гиппокампе. CSCG представляет последовательности наблюдений в виде графа, где узлы – это “клоны” наблюдений, соответствующие различным контекстам, а рёбра – переходы (последовательные связи) между узлами. Идея в том, что одно и то же наблюдение o в разных контекстах (предшествующие события) представляется разными узлами $o_1, o_2, \ldots$ (клонами), что позволяет избежать проблем с ситуациями, когда одно наблюдение может означать разные ситуации. Граф высокой размерности с такими дублированными узлами способен одновременно отражать обобщение (узнавая одинаковые наблюдения) и дифференциацию по контексту.

Формально, CSCG можно рассматривать как расширение скрытой марковской модели, где каждому возможному наблюдению x соответствует набор скрытых состояний $z : \text{emit} = x$ – “клонов” этого наблюдения. Переходы возможны только между клонами, соответствующими последовательным наблюдениям. Обучение CSCG эквивалентно обучению клонированной HMM (CHMM) –

модификации алгоритма Баум–Уэлша с разреженной матрицей переходов. В методе предложено использовать ожидание-максимизацию (EM) для подгонки вероятностей переходов между клоновыми состояниями, группируя состояния по эмиссиям. Благодаря введению клонов модель эффективно разделяет "путающиеся" наблюдения по разным траекториям, что усиливает способности к разделению паттернов и обобщению.

CSCG изначально мотивирован задачами навигации: в лабиринтах животные формируют когнитивную карту, где одно и то же место может играть разные роли в разных маршрутах. Модель объясняет феномены **splitter cells** в гиппокампе – нейронов, которые избирательно активны для одного и того же места в зависимости от того, откуда пришло животное или куда направляется. Также CSCG объясняет наблюдаемую перекодировку **place-cell** при изменении контекста. С вычислительной точки зрения, CSCG – это вероятностный граф, решающий задачу последовательной памяти через построение графа состояний с мультипликативным ростом (число узлов = число наблюдений $\times$ число контекстных вариаций). Для обучения и вывода используются методы message-passing на графе, обеспечивающие устойчивость к шуму. Недостатком CSCG является потенциально высокая сложность (граф может стать очень большим) и необходимость заранее задавать или адаптивно выбирать число клонов на наблюдение. Тем не менее, CSCG представляет интересный био-вдохновленный подход, объединяющий статистический HMM-подход с иерархическим графовым представлением опыта.

1.3.4 Echo State Networks, Liquid State Machines & State Space Models

Альтернативный подход к обучению временной памяти предложен в концепции резервуарных вычислений. К ней относятся модели эхо-состояний (Echo State Network, ESN) [35] и жидкостных состояний (Liquid State Machine, LSM) [36]. Их общая черта – фиксированная рекуррентная "резервуарная" сеть, внутри которой вводится входной сигнал, возбуждающий динамическое "эхо", и лишь выходные веса обучаются для считывания состояния резервуара. В Echo

State Network скрытый слой состоит из большого числа нейронов с случайными рекуррентными связями, обычно разреженными. Весовые коэффициенты внутри резервуара фиксируются при инициализации (с соблюдением “эхо-условия”: спектральный радиус матрицы весов W_{hh} должен быть < 1 для устойчивости динамики). Входные веса W_{in} также фиксируются (случайно). Когда подаётся входная последовательность $\mathbf{x}_t$, нейроны резервуара обновляют состояния:

$$h_t = \tanh(W_{\text{in}}\, x_t + W_{\text{hh}}\, h_{t-1}) \tag{1.2}$$

, генерируя богатое нелинейное динамическое эхо недавней истории входов. Задача обучения сводится к нахождению выходной матрицы W_{out}, такой что выход $\mathbf{y}_t = W_{out}\mathbf{h}_t$ приближает желаемую последовательность (или целевое преобразование входа).

Liquid State Machine (LSM) – биологически вдохновленный вариант резервуарного подхода, предложенный для спайковых нейронов. В LSM резервуар – это случайная рекуррентная сеть спайковых нейронов (часто модель integrate-and-fire), а выход формируется через обучаемый «читающий» нейрон или слой, который интегрирует выходы всех нейронов резервуара (например, через линейную комбинацию с обучаемыми коэффициентами). Фиксированные случайные связи резервуара и динамика спайков создают “жидкое” множество временных признаков – т.е. множество возможных откликов на разные входные стимулы.

Отличие ESN и LSM: ESN обычно оперирует непрерывными значениями (активностями), LSM – спайковыми событиями; но математически обе можно рассматривать как динамические системы

$$h_t = F(h_{t-1}, x_t) \tag{1.3}$$

с фиксированной нелинейной функцией F (задана случайно) и обучаемым лишь линейным считыванием. Их преимущество – скорость обучения и простота, недостаток – необходимость большого резервуара и отсутствие адаптации резервуарных весов под задачу (хотя существуют расширения, допускающие небольшое обучения внутри резервуара).

Модели пространств состояний (**State Space Models, SSM**) – общий термин, но в контексте современных нейросетевых архитектур он обозначает

класс последовательных моделей, использующих линейно-дифференциальные (или разностные) уравнения состояния для представления долгих зависимостей. Классический SSM в теории управления описывается системой:

$$x_{t+1} = Ax_t + Bu_t, \quad y_t = Cx_t \tag{1.4}$$

где $\mathbf{x}_t$ – вектор скрытого состояния, $\mathbf{u}_t$ – вход, $\mathbf{y}_t$ – выход, A, B, C – матрицы. Собственные значения матрицы A определяют динамику (обычно затухающие колебания или экспоненциальные хвосты), что позволяет хранить информацию о прошлом в компоненте $\mathbf{x}_t$ продолжительное время. В нейросетевых SSM предлагается обучать параметры A, B, C (а иногда и обобщать до нелинейных $A(\mathbf{x})$), чтобы аппроксимировать нужную функцию от последовательности входов.

Конкретным примером современной SSM является модель **S4 (Structured State Spaces for Sequence)** [37]. В S4 непрерывное уравнение $d\mathbf{x}/dt = A\mathbf{x} + B\mathbf{u}(t)$ с определённой структурой A (блок-диагональная матрица с линейно расположенными экспоненциальными корнями) решается аналитически, и полученное дискретное свёрточное ядро используется для эффективной обработки последовательностей. Это даёт возможность моделировать очень длинные последовательности с субквадратичной сложностью. Интуитивно, SSM-архитектуры пытаются имитировать фильтры с долговременной памятью (память в динамических системах), но параметры этих фильтров обучаются, как нейросеть, под задачу.

SSM представляют собой современное развитие идей RNN, где рекуррентная связь заменена на решаемую линейную систему, обучаемую под данные. Биологически аналогии SSM можно усмотреть в линейно-временных откликах нейронов, например, дифференцировании и интегрировании входных сигналов дендритами, однако прямая связь с нейрофизиологией пока условна.

1.4 Иерархическая временная память

1.4.1 Биологическая модель

Одной из моделей, пытающихся соединить нейробиологические принципы и алгоритмы машинного обучения для задачи памяти последовательностей, является Hierarchical Temporal Memory (HTM) [38–40] – иерархическая временная память. HTM основывается на строении неокортекса. В частности, модель HTM пытается воспроизвести работу неокортексовых **колонок** и свойств пирамидальных нейронов (многочисленные синапсы на апикальных и базальных дендритах).

HTM строится из элементарных модулей – регионов, которые можно представить аналогом области коры. Каждый регион разделён на множество миниколонок, а в каждой миниколонке содержится несколько (10-100) пирамидальных нейронов. Эти нейроны имеют сегментированые дендриты: проксимальные сегменты получают вход от предыдущего уровня или сенсорных данных, базальные сегменты – от нейронов того же уровня (контекстные связи), апикальные сегменты – от вышележащих уровней. Такая структура близко соответствует организации реальных нейронов. Проксимальный дендрит определяет реакцию нейрона на текущий вход (его рецептивное поле), а базальные и апикальные отвечают за прогнозирование – они могут деполяризовать нейрон, не вызывая спайка, если распознают паттерн активности соседних (или вышележащих) нейронов, предвосхищающий активацию данного нейрона.

Для описания HTM-алгоритма введём некоторые обозначения. Пусть M – число миниколонок, n – число клеток на колонку. Состояние системы в момент t задаётся множеством активных клеток $A_t \subset 1,\ldots,M \times 1,\ldots,n$ и множеством предсказывающих клеток P_t. Предсказывающее состояние определяется на основании локальных совпадений на базальных дендритах: у каждой клетки есть множество потенциальных синаптических связей от других клеток, и каждый базальный дендрит функциирует как детектор совпадения. Формально, пусть у клетки c имеется набор дендритов $D_{c1}, D_{c2}, \ldots$, и каждый дендрит D_{cj} имеет

множество входных синапсов $S(D_{cj})$ от некоторых клеток. Определим бинарную функцию активности дендрита:

$$\text{act}(D_{cj}, t) = \begin{cases} 1, & \text{если } |\{x \in S(D_{cj}) \mid x \in A_t\}| \geq \theta \\ 0, & \text{иначе} \end{cases} \tag{1.5}$$

где θ – порог срабатывания (минимальное число активных входов, чтобы дендрит D_{cj} возбудился). Клетка c переходит в предсказывающее состояние на шаге $t+1$ (т.е. $c \in P_{t+1}$), если хотя бы один её базальный дендрит возбудился на шаге t:

$$c \in P_{t+1} \iff \exists j \text{ таких, что } \text{act}(D_{cj}, t) = 1 \tag{1.6}$$

1.4.2 Обучение HTM

Правила обучения в HTM реализуют принцип хеббовского обучения на уровне синапсов дендритов. Если клетка c оказалась активна *и предсказана* (т.е. была в P_t и перешла в A_t), то укрепляются синапсы тех дендритов, которые и вызвали эту успешную предикцию. Если же не было предсказанной клетки, то для одной из активных клеток (обычно выбирают первую в колонке или случайную) создаётся новый дендритный сегмент или укрепляются синапсы существующего сегмента так, чтобы связать текущий контекст A_{t-1} с данной клеткой.

Таким образом, HTM запоминает последовательности путём формирования привязанных паттернов активности: каждый шаг последовательности представлен уникальным сочетанием активных клеток (контекстно-зависимым), и переход $X \to Y$ запечатлевается в синапсах от паттерна X к клеткам, составляющим паттерн Y. Интересно, что предсказание может быть многозначным: если после одного контекста A возможны два разных продолжения (B или C), то при A деполяризуются клетки и для B, и для C. Тогда когда придёт следующий вход, совпадающий с одним из них, соответствующие нейроны активируются, а другие останутся лишь деполяризованными (не породив спайк). HTM естественным образом обрабатывает “развилки” последовательностей таким способом и

даже может хранить несколько гипотез вперёд, пока не придёт разрешающий вход (аналогично ветвлению прогнозов).

HTM использует чисто локальное обучение без необходимости обратного распространения ошибки. Это близко к биологической правдоподобности [41; 42]: правило усиления синапсов зависит только от активности пресинаптической и постсинаптической клеток (что соответствует гипотезе обучения в мозге через локальные изменения). Более того, механизм предсказания через деполяризацию напоминает поведение реальных нейронов, обнаруженное экспериментально: пресинаптические потенциалы, не приводящие к спайку, тем не менее влияют на вероятность будущего возбуждения нейрона. HTM предполагает, что мозг постоянно держит множество нейронов в таком "ожидающем" состоянии, что подтверждено нахождением NMDA-спайков на дендритах в отсутствии непосредственного возбуждения нейрона. Применения HTM были продемонстрированы в задачах предсказания временных рядов, обнаружения аномалий, а также в качестве компонента для обработки сенсорных последовательностей. Модель способна *онлайн* учиться последовательностям в потоке данных – за счёт локального обучения она непрерывно обновляет память по мере поступления информации, без необходимости офлайн-итераций по данным. Это ценно для встроенных систем и обработки сигналов с временной структурой (например, прогнозирование показателей, детекция нехарактерных событий).

Ограничения HTM заключаются в основном в отсутствии глобальной оптимизации: модель не "выводит" абстрактные признаки через глубокие представления, как это делают обученные end-to-end нейросети. Её иерархическая составляющая остаётся в значительной мере теоретической: хотя концептуально регионы могут образовывать иерархию, практически приложения чаще ограничиваются одним уровнем памяти. Кроме того, HTM в текущем виде оперирует дискретными бинарными активациями (спайками) и фиксированной пороговой логикой, что затрудняет использование при шумных непрерывных данных (необходим этап кодирования входов в SDR). Тем не менее, HTM внесла значительный вклад, предложив конкретный алгоритм последовательной памяти, выведенный из нейробиологических принципов: разреженности, дендритных совпадений, локального обучения и иерархичности колончатой организации коры.

1.5 Предсказательное кодирование и сети Хопфилда

Теория предсказательного кодирования (Predictive Coding) [22; 23; 29] и ассоциативные памяти Хопфилда [43] – два направления, предложенные для объяснения памяти и восприятия в нейронных системах. Несмотря на разный исторический контекст, современные исследования выявляют глубокую связь между ними. Рассмотрим классические и новые модели в этих областях, а затем их взаимосвязь.

1.5.1 Классические сети Хопфилда и их обобщения

Сеть Хопфилда – один из первых формальных примеров автоассоциативной памяти. Она представляет собой полностью связанный рекуррентный нейронный слой, где каждый узел (нейрон) s_i биполярный (значение ± 1) и симметрично соединён с другими [43]. Весовые коэффициенты $w_{ij} = w_{ji}$ задаются так, чтобы определённые бинарные вектора ξ^μ (паттерны памяти) стали **аттракторами** динамики сети.

Динамика обновления: на каждом шаге выбирается нейрон i, и его состояние задаётся по правилу:

$$s_i(t+1) = \operatorname{sgn}\left(\sum_j w_{ij} s_j(t)\right), \tag{1.7}$$

(возможны как синхронные, так и асинхронные обновления всех нейронов).

Хопфилд показал, что при симметричных весах существует скалярная функция энергии:

$$E = -\frac{1}{2}\sum_{i,j} w_{ij} s_i s_j, \tag{1.8}$$

которая не возрастает при каждом обновлении состояния сети. Таким образом, динамика сходится к одному из локальных минимумов E, что и будет устойчивым шаблоном (аттрактором).

Если веса обучены на паттерны ξ^μ:

$$w_{ij} = \frac{1}{N} \sum_\mu \xi_i^\mu \xi_j^\mu, \tag{1.9}$$

то эти паттерны становятся аттракторами.

Расширения сети Хопфилда направлены на увеличение ёмкости и на снятие ограничений симметричности. **Ассиметричная сеть Хопфилда** (**AHN**) [44; 45] – это вариант, где $w_{ij} \neq w_{ji}$. В классической постановке нарушение симметрии убирает гарантию сходимости: энергия не определена, динамика может начать осциллировать или хаотически кружить. Тем не менее, асимметричные сети могут функционировать как ассоциативная память, если рассматривать их динамику как последовательное воспроизведение паттернов (циклы) вместо статических аттракторов. В частности, можно построить цикл из нескольких состояний – асимметричные веса будут поочерёдно переключать сеть через последовательность. Это можно трактовать как временную память: одно состояние сети вызывает другое. Однако отсутствует строгая теория стабильности для общего случая. Некоторые работы вводят ограничения (например, разделение матрицы весов на симметричную + кососимметричную часть) для обеспечения частичной функций энергии. Другой прогресс – современные непрерывные сети Хопфилда с высокой ёмкостью. В работах последних лет показано, что если разрешить нейронам принимать непрерывные значения (не только ± 1) и ввести новую энергию, например с ненелинейной (экспоненциальной или другой) функцией схожести, то сеть может хранить экспоненциально много паттернов. Например, в работах [44; 46] предложена энергия, эквивалентная **механизму внимания** (softmax по схожести ключей и запросов), и показано соответствие такой сети Хопфилда и трансформера. Эти Modern Hopfield Networks (MHN) по-сути снимают прежнее ограничение ёмкости за счёт непрерывности и другой динамики (градиентного спуска по новой энергии). Они менее био-подобны, но концептуально расширяют идею ассоциативной памяти к мощным моделям, связанным с глубоким обучением.

1.5.2 Предсказательное кодирование: классические и современные модели

Предсказательное кодирование [22; 23; 29] (PC) – теория, согласно которой основной функцией нейронных систем является генерация предсказаний сенсорных входов и обработка ошибок предсказания. В классической модели PC рассматривалась двухслойная зрительная система: верхний слой формирует предсказанные активность нижнего (например, V1) через обратные связи, а нижний вычисляет разницу между реальным входом и предсказанным (ошибку), которую передаёт наверх. Таким образом, в сети имеются два типа нейронов: предикторы и нейроны ошибки. Предикторы посылают вниз ожидаемые признаки, нейроны ошибки сигналят расхождение, корректируя верхний уровень. Алгебраически это выражается как попытка минимизировать локальную ошибку $E = \frac{1}{2}|x - \hat{x}|^2$ – квадратичное отклонение между реальным входом x и реконструкцией $\hat{x}$, полученной из активности верхних нейронов r через матрицу W: $\hat{x} = Wr$.

Градиентный спуск по r (с учётом регуляризации) приводит к правилу обновления:

$$\Delta r \propto W^T(x - \hat{x}) - \lambda r, \tag{1.10}$$

что можно интерпретировать как: «активность верхнего нейрона растёт пропорционально ошибке предсказания в связанных нижних нейронах» (первый член) и затухает пропорционально своей величине (второй член — приор).

В непрерывном варианте получаются дифференциальные уравнения динамики:

$$\frac{dr}{dt} = -\frac{\partial E}{\partial r}. \tag{1.11}$$

Аналогичные уравнения можно записать для обновления весов W при обучении модели представления.

В более общем виде, Predictive Coding Network (PCN) — иерархическая сеть, где каждый уровень пытается предсказать следующий нижележащий.

Обозначим:

– r^l — активности на уровне l,

– $W^{l \to l-1}$ — веса, генерирующие предсказание вниз,

– $W^{l-1\to l}$ — веса, поднимающие ошибки вверх.

Для простоты предполагается, что $W^{l-1\to l} \propto (W^{l\to l-1})^T$, для согласованности кодирования.

Тогда ошибка на уровне $l-1$:

$$\varepsilon^{l-1} = r^{l-1} - \hat{r}^{l-1}, \tag{1.12}$$

$$\hat{r}^{l-1} = W^{l\to l-1} r^l. \tag{1.13}$$

Динамика активности уровня l описывается уравнением:

$$\tau \frac{dr^l}{dt} = -\frac{\partial E}{\partial r^l} = -(W^{l\to l-1})^T \varepsilon^{l-1} + \varepsilon^l, \tag{1.14}$$

где $\varepsilon^L = 0$ на верхнем уровне L (нет предсказателя выше).

Это уравнение означает, что активность на уровне l обновляется пропорционально:

– ошибке на нижнем уровне ε^{l-1}, трансформированной вверх,

– минус ошибке собственного уровня ε^l, так как этот уровень сам предсказывается вышестоящим.

В результате система стремится к состоянию, где все ошибки ε малы, то есть каждый уровень достаточно хорошо объясняет нижележащий.

Обучение весов происходит медленнее, по накопленным корреляциям между r^l и ε^{l-1}, аналогично правилу Хебба или перцептрону:

$$\Delta W^{l\to l-1} \propto r^l \cdot (\varepsilon^{l-1})^T. \tag{1.15}$$

Предсказательное кодирование объясняет ряд нейрофизиологических фактов, например, наличие в коре нейронов, кодирующих ошибки (некоторые интернейроны могут соответствовать этому) и феномены адаптации к ожидаемым стимулам. С точки зрения памяти, PCN представляет динамический процесс, где сохранение последовательности реализуется как постоянное поддержание предсказаний: ранее увиденное указывает сети, что ожидать далее, и любая рассинхронизация вызывает сигнал ошибки. Однако, классические PCN в основном исследовались на статических или стационарных паттернах (картинках, устойчивых признаках). Расширение на последовательности требует механизма

для удержания прошлых состояний – например, рекуррентных связей или временного контекста.

Главное сходство – и PCN, и Hopfield-память можно описать как системы, минимизирующие некий энергетический или целевой функционал. Для сети Хопфилда это функция энергии $E_{\text{Hopf}}(\mathbf{s})$, для PCN – суммарная ошибка предсказания $E_{\text{PC}}(r,\varepsilon)$. Оказалось, что подходящим выбором переменных можно преобразовать E_{PC} к виду, близкому к энергии Hopfield, но с некоторыми асимметричными компонентами.

В недавней работе [47] также отмечено, что энергетическая функция Hopfield-сети может быть интерпретирована как функция "предсказание + ошибка": минимум энергии соответствует состоянию, где все нейроны согласованы со взвешенными суммами. Это философски пересекается с принципом **free-energy minimization** Фристона [15], общим случаем которого является predictive coding.

Сети Хопфилда изначально не претендовали на строгую нейрофизиологическую реалистичность: бинарные нейроны, симметричные синапсы – скорее математическая идеализация. Однако идеи ассоциативной памяти были чрезвычайно влиятельны в нейрокогнитивной науке. Предложена интерпретация гиппокампа как автоассоциативной памяти (CA3 область как сеть Хопфилда для быстрого запоминания эпизодов). Что касается симметрии, биологические синапсы, конечно, не симметричны; но двунаправленные связи между областями мозга могут приближаться к этому (например, корково-кортикальные петли). Асимметричные Hopfield-типа сети (с оговорками по стабильности) ближе к реальности, и как описано выше, могут быть реализованы через механизмы предсказательного кодирования.

Предсказательное кодирование, напротив, разрабатывалось именно как биоправдоподобная теория. Нейроны ошибки можно сопоставить, например, с определёнными интернейронами, а нейроны предсказания – с пирамидальными клетками. Обмен сигналами "вверх ошибка / вниз предсказание" согласуется с известной архитектурой "feedforward/feedback" в сенсорных системах. Также PC предлагает объяснение, почему в мозге так много обратных связей: они несут предсказания. Локальное обучение на основе ошибок реализуемо через стандартные механизмы (правило Хебба с модуляцией от нейромодуляторов,

представляющих глобальную ошибку, или STDP, чувствительное к рассинхронизации активности).

Глава 2. Распределенная хеббовская временная память (DHTM)

2.1 Алгоритм распространения доверия и факторная скрытая марковская модель

Существует множество разработок, основанных на идеях иерархической временной памяти (HTM), которая обсуждалась ранее. Расширения архитектуры HTM могут включать добавление нейронов с механизмом GRU [48]. Это позволило улучшить производительность модели при работе с долгосрочными зависимостями в анализе временных рядов [39]. В рамках обучения без учителя можно объединить Extreme Learning Machine [49] и HTM в одной модели [40]. HTM также может использоваться в алгоритмах обучения с подкреплением на основе модели мира [50].

Модель о которой пойдет речь в этой главе — DHTM — [51] основана на HTM и интегрирует её с HMM, используя вероятностную модель сегментов, образуя гибридный алгоритм обучения по последовательностям. DHTM преподносит идею разреженных распределённых представлений (Sparse Distributed Representations) на категориальные скрытые переменные состояния, дендритные сегменты нейронов — на компоненты матрицы переходов частично связанной HMM [25], а кортикальные колонки — на функцию вероятности эмиссии, аналогично подходу в CSCG [52].

DHTM основана на алгоритме передачи сообщений в графе и обучения основанного на распространении доверия (belief propagation). Алгоритм распространения доверия представляет собой механизм передачи сообщений для вывода в графовых вероятностных моделях. Он часто используется в вероятностных моделях для вычисления маргинальных распределений и наиболее вероятных значений переменных.

Для вычисления маргинальных вероятностей узел X_i в графе отправляет сообщение узлу X_j по ребру (i, j) с использованием следующего уравнения:

$$m_{i \to j}(x_j) = \sum_{x_i} \psi_{ij}(x_i, x_j) \prod_{k \neq j} m_{k \to i}(x_i), \tag{2.1}$$

где $\psi_{ij}(x_i, x_j)$ — функция, представляющая взаимосвязь между узлами X_i и X_j, а $m_{k\to i}(x_i)$ — сообщение от узла X_k к X_i. Итерации продолжаются до сходимости [53].

Графовая структура, обучаемая предлагаемым методом, относится к классу факторных скрытых марковских моделей (Factorial Hidden Markov Models), которые расширяют концепцию HMM, позволяя нескольким скрытым переменным состояния действовать одновременно. Вероятности переходов между скрытыми состояниями в факторной HMM описываются следующим уравнением:

$$P(H_{t+1}|H_t) = \prod_{i=1}^{N} A_i(H_{t,i}, H_{t+1,i}), \tag{2.2}$$

где N — количество скрытых переменных состояния, а $A_i(H_{t,i}, H_{t+1,i})$ — матрица переходов для каждой скрытой переменной. Более общая формулировка факторной HMM также позволяет учитывать зависимости между различными скрытыми переменными [25].

Модель DHTM вдохновлена этими подходами и предоставляет мощные средства для вероятностного моделирования последовательных данных.

2.2 Классическая модель

Здесь кратко описан алгоритм распределённой хеббовской временной памяти (DHTM), усовершенствование которого является предметом научной новизны. Для более подробной информации см. [51].

DHTM реализует обучение и вывод в фактор-графе, показанном на Рисунке 2.1-A, для одной наблюдаемой переменной. Все случайные переменные имеют категориальный характер. Модель DHTM может включать несколько наблюдаемых переменных O_t^k, обрабатываемых параллельно, где t — текущий временной шаг, а k — номер переменной. Каждая скрытая переменная H_t^k формирует модель объяснения соответствующей наблюдаемой переменной O_t^k и рассматривается независимо от других скрытых случайных переменных для текущего временного шага. Фактор контекста F_c^k описывает зависимость между подмножеством скрытых переменных предыдущего временного шага и текущей скры-

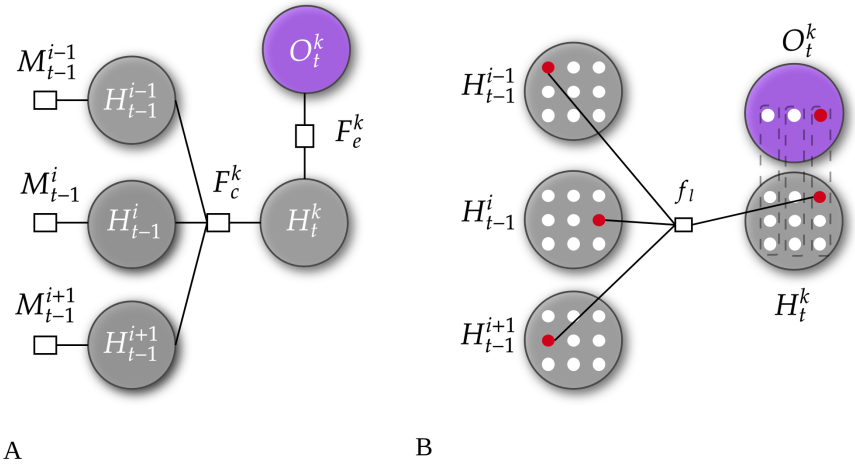

Рисунок 2.1 — A. Фактор-граф DHTM. B. Внутренняя структура переменных и факторов. Каждый сегмент соответствует значению фактора контекста f_l для комбинации состояний (красные круги). Фактор эмиссии фиксирован таким образом, что каждое наблюдаемое состояние соответствует группе скрытых состояний (столбец), которые могут быть активированы с равной вероятностью при активном наблюдаемом состоянии.

той переменной H_t^k для k-й переменной. Сообщения с предыдущих временных шагов представлены факторами M_{t-1}^i. Фактор эмиссии F_e^k фиксирован и задаёт правило обновления апостериорного распределения скрытой переменной (Рисунок 2.1-B).

Одной из ключевых особенностей DHTM является то, что она обрабатывает функции фактора контекста в виде набора вычислительных единиц, называемых сегментами, которые являются аналогами биологических дендритных сегментов пирамидальных нейронов. Каждый сегмент соответствует значению фактора для комбинации состояний переменных, являющихся аргументами функции фактора. Поскольку в модели используется только небольшая часть возможных комбинаций, этот подход аналогичен используемому для эффективной обработки разреженных матриц [54]. Однако, как будет показано в следующих разделах, не каждая комбинация требует отдельного сегмента, и некоторые из них могут быть объединены, что дополнительно улучшает вычислительную эффективность алгоритма.

Вычисления сегмента в DHTM изображены на Рисунке 2.2-A. Сегмент можно рассматривать как случайную величину с распределением Бернулли, вероятность активации P_l которой зависит от значения фактора f_l, эффектив-

ности синапсов w_u и вероятностей активации предсинаптических клеток (состояний переменных) m_u. Предсинаптические клетки формируют рецептивное поле сегмента $rec(l)$, которое соответствует комбинациям состояний скрытых переменных предыдущего временного шага. Вычисления происходят на этапе распространения доверия и представляют собой модифицированную версию алгоритма суммы-произведения, которая приблизительно учитывает зависимость между переменными H^i_{t-1}, не учитываемую самой моделью.

Количество используемых сегментов определяется на этапе роста сегментов, который следует за этапом распространения доверия. Состояния переменных сэмплируются из соответствующих распределений, и определяется перекрытие между активными клетками и рецептивными полями сегментов. Если нет сегментов, соответствующих данной комбинации состояний, создаётся новый сегмент.

Таким образом, в DHTM новый сегмент создаётся для каждой новой комбинации выбранных состояний. Однако в шумных средах это приводит к избыточному количеству сегментов, которые редко используются и могут даже негативно влиять на качество модели.

Обычно в таких случаях применяется обрезка для удаления шумных сегментов на основе частоты их активации. Однако такой подход может удалить важные зависимости и ухудшить производительность временной памяти. Более того, это требует настройки гиперпараметров, таких, как ограничение на максимальное количество сегментов и доля обрезаемых сегментов, которые зависят от среды и не могут быть заранее определены. В предложенном новом методе разработан универсальный способ повышения устойчивости DHTM к шуму путём модификации вычислений и роста сегментов.

2.3 Объединение сегментов

Ключевая идея заключается в объединении сегментов с существенно перекрывающимися рецептивными полями. Для этого на этапе роста сегментов устанавливается порог активации перекрытия сегмента ниже размера рецептивного поля. Иными словами, при обнаружении новой комбинации состояний

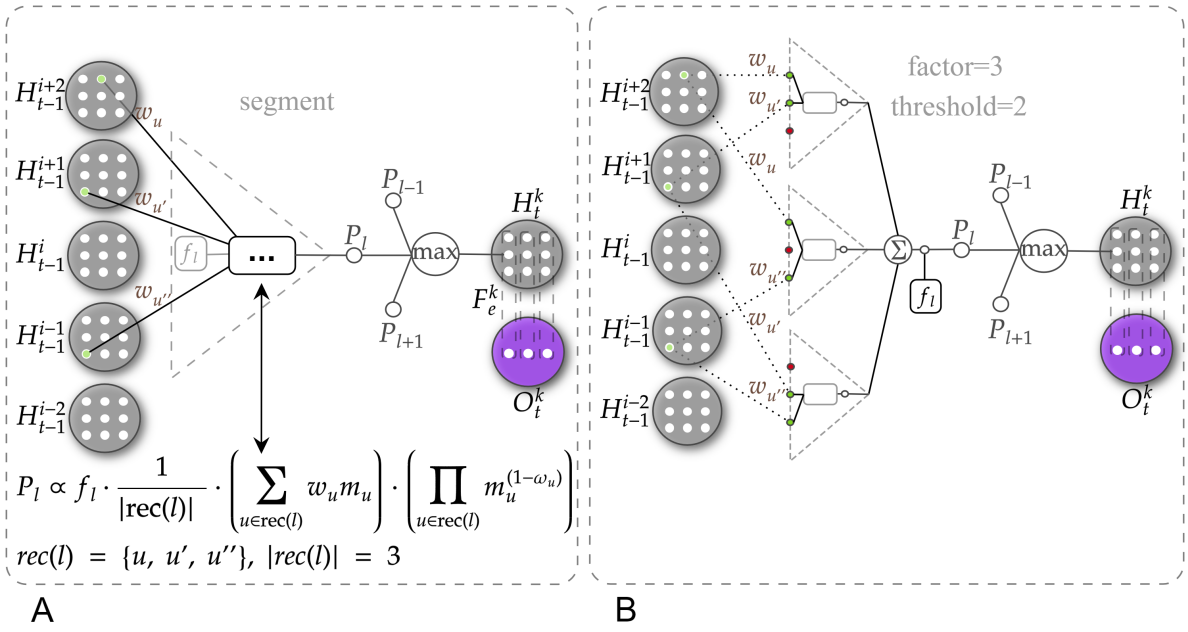

Рисунок 2.2 — A. Вычисление вероятности активации сегмента с использованием значения фактора f_l для расчёта эффективного возбуждения клетки сегментом l. B. Демонстрация учёта комбинаций подсегментов, пример для размера фактора = 3 и порога = 2.

сначала выполняется поиск сегмента с похожим рецептивным полем. Однако такой подход требует модификации этапа распространения доверия, так как сегмент может быть активирован несколькими способами.

Основная концепция заключается в суммировании вероятностей активации сегмента по всем возможным комбинациям состояний в рецептивном поле. Каждую такую комбинацию мы называем подсегментом. Однако мы исключаем комбинации, превышающие заданный порог (Рисунок 2.2-B). Это делается по нескольким причинам:

– Вклад в сумму обычно уменьшается с ростом переменных в произведении, когда речь идет о вероятностях;
– Это значительно сокращает количество операций.

В данном случае количество учитываемых комбинаций подсегментов составляет C^{trsh}_{factor}. Тогда уравнение для вероятности активации сегмента принимает следующую форму:

$$P_l \propto f_l \cdot \sum_{\substack{\mathrm{rec}^i(l) \\ |\mathrm{rec}^i(l)|=\mathrm{trsh}}} \left(\frac{1}{|\mathrm{rec}^i(l)|} \cdot \sum_{u \in \mathrm{rec}^i(l)} w_u m_u \right) \cdot \left(\prod_{u \in \mathrm{rec}^i(l)} m_u^{(1-\omega_u)} \right) \tag{2.3}$$

где $\mathrm{rec}^i(l)$ — подмножество клеток в рецептивном поле, соответствующее i-му подсегменту, а $|\mathrm{rec}^i(l)|$ — количество клеток в подмножестве.

2.4 Адаптивное пороговое значение

Как уже упоминалось, суть сегментов заключается в том, что они представляют отображение комбинаций состояний переменных в значения факторов. Совокупность сегментов формирует фактор-функцию. Установив порог активации ниже размера фактора, мы просто отображаем все комбинации состояний определённого размера, найденные в рецептивном поле, в одно значение фактора, тем самым сокращая количество параметров, определяющих фактор-функцию. Однако это может быть слишком жёстким ограничением, и наши эксперименты показывают, что качество модели может быть улучшено, если порог адаптивно настраивается для каждого сегмента.

Порог на этапе роста сегментов определяет активные сегменты. Чем выше порог, тем меньше активных сегментов, и, следовательно, тем выше вероятность создания нового сегмента, и наоборот.

Подход основывается на идее, что чем меньше значение фактора, тем менее полезен сегмент, поскольку оно пропорционально частоте активации соответствующей комбинации состояний. Это означает, что если все сегменты имеют низкие значения факторов, стоит сформировать новый сегмент в надежде, что он захватит более удачную комбинацию состояний. Увеличение порога сегмента обратно пропорционально его значению фактора позволяет достичь этой цели: оно увеличивает вероятность роста нового сегмента в случае, если ни один сегмент с высоким значением фактора не был активирован. Начальное значение для новых сегментов устанавливается на максимум, а порог, соответственно,

минимален. Если в процессе обучения значение фактора уменьшается, порог увеличивается, снижая вероятность активации сегмента.

Глава 3. Временное предсказательное кодирование (TPC)

3.1 Предсказательное кодирование и временные представления в RL

Как отмечалось ранее, предсказательное кодирование, вдохновлённое нейронаукой, предполагает, что мозг и искусственные системы обрабатывают информацию путём непрерывного предсказания сенсорных входов и минимизации ошибок предсказания [22;23]. В машинном обучении этот принцип лежит в основе класса моделей, где каждый слой или временной шаг стремится предсказать следующий, используя возникающую ошибку для обновления внутренних представлений. Эти модели естественным образом распространяются на последовательные данные и показали успех в обучении временным структурам, например, в задаче предсказания видео [55] или контрастивного обучения латентной динамики [56].

Временное предсказательное кодирование (TPC) развивает эту идею, поэтапно предсказывая будущие входы и используя только локальные обновления на основе ошибок [26]. Это приводит к компактным латентным представлениям, которые хранят информацию во времени, фактически функционируя как форма предсказательной памяти. Теоретический анализ связывает TPC с асимметричными сетями Хопфилда с выбеливанием (whitening) латентных представлений, что обеспечивает большую ёмкость и устойчивость для последовательного воспроизведения [26]. Эти свойства делают TPC перспективным кандидатом для представления состояний (belief states) в частично наблюдаемых средах.

В обучении с подкреплением это согласуется с предсказательными представлениями состояния [57], где будущие наблюдения определяют внутреннее состояние. Основываясь на этом, предложенно интегрировать модель на основе TPC в качестве внутреннего модуля памяти для RL-агентов, который обучается с помощью минимизации потерь. Это стимулирует латентные представления отслеживать скрытые переменные среды, поддерживая более эффективное планирование и принятие решений в условиях неопределённости.

Предсказательные представления состояния (PSR) неявно кодируют важную информацию о состоянии через предсказание будущих наблюдений [57]. Основываясь на этом, наш подход явно интегрирует модель мира на основе TPC в RL-агента. Обучение с целью предсказательного кодирования наряду со стандартной оптимизацией RL обеспечивает надёжные представления убеждения, позволяя агенту эффективно справляться с частичной наблюдаемостью за счёт фокусировки на релевантной динамике и фильтрации отвлекающих факторов. Детальная архитектура и экспериментальная оценка приведены в следующих разделах.

3.2 Временное предсказательное кодирование в RL-сценарии

Мы используем TPC, расширяя классические сети предсказательного кодирования для учёта последовательных задач, встречающихся в сценариях обучения с подкреплением. Рисунок 3.1 иллюстрирует двухслойную реализацию TPC, адаптированную для обучения с подкреплением.

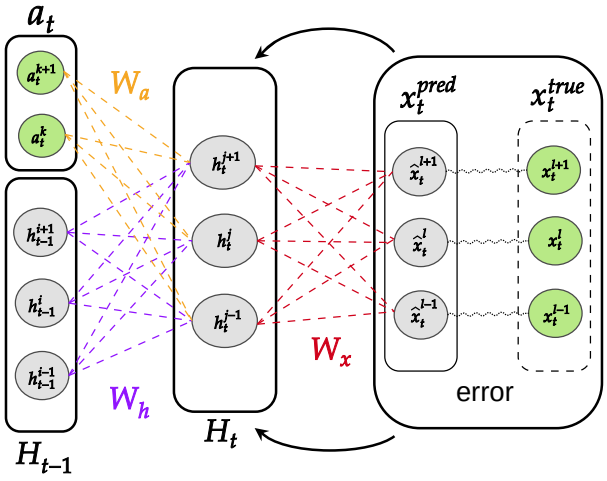

Рисунок 3.1 — Архитектура двухслойной сети временного предсказательного кодирования (TPC).

Сеть получает на вход последовательности наблюдений $x_1, \ldots, x_T$ вместе с соответствующими действиями агента $a_1, \ldots, a_T$. Латентные представления на каждом временном шаге t обозначаются как H_t. На каждом шаге времени

модель минимизирует следующую функцию потерь:

$$L_t = \|x_t - \hat{x}_t\|_2^2 + \left\|H_t - \hat{H}_t\right\|_2^2 = \\ = \|x_t - f(W_x H_t)\|_2^2 + \left\|H_t - h(W_h \hat{H}_{t-1} + W_a a_t)\right\|_2^2, \tag{3.1}$$

где f и h обозначают нелинейные функции активации, а W_a, W_h и W_x — обучаемые матрицы весов, используемые для генерации предсказаний. Ошибки предсказания на каждом шаге определяются как:

$$\epsilon_t^x := x_t - f(W_x H_t), \quad \epsilon_t^H := H_t - h(W_h \hat{H}_{t-1} + W_a a_t). \tag{3.2}$$

Процедура обучения состоит из двух последовательных этапов. Сначала латентные представления H_t оптимизируются с помощью градиентного спуска:

$$\Delta H_t \propto -\nabla_{H_t} L_t = -\epsilon_t^H + W_x^\top f'(W_x H_t)\epsilon_t^x. \tag{3.3}$$

После оптимизации латентных представлений H_t они присваиваются рекуррентным состояниям $\hat{H}_t$ для использования в предсказании на следующем временном шаге. Затем веса модели обновляются следующим образом:

$$\Delta W_x \propto f'(W_x H_t)\epsilon_t^x H_t^\top, \;\; \Delta W_h \propto h'(\tilde{H}_t)\epsilon_t^H \hat{H}_{t-1}^\top, \;\; \Delta W_a \propto h'(\tilde{H}_t)\epsilon_t^H a_t^\top, \tag{3.4}$$

где f' и h' обозначают якобианы нелинейных функций f и h соответственно, а $\tilde{H}_t = W_h \hat{H}_{t-1} + W_a a_t$.

Во время валидации (фазы воспроизведения) модель работает с фиксированными весами W_h, W_a, и W_x, и только латентные состояния H_t динамически обновляются для минимизации ошибки предсказания на каждом шаге. Когда истинное наблюдение x_t доступно, модель итеративно обновляет H_t по градиентному правилу из Eq. 3.3 и корректирует предсказанное наблюдение $\hat{x}_t$ пропорционально сенсорной ошибке предсказания ϵ_t^x (Eq. 3.2), тем самым согласуя внутренние представления с наблюдаемыми данными. Напротив, когда наблюдения недоступны — например, при генерации последующих признаков (successor features) во время RL-планирования — модель выполняет одношаговое прямое предсказание, используя предыдущее латентное состояние $\hat{H}_{t-1}$ и текущее действие a_t. Этот предсказательный механизм позволяет агенту симу-

лировать будущие траектории на основе своей внутренней модели, давая возможность принимать решения в частично наблюдаемых средах.

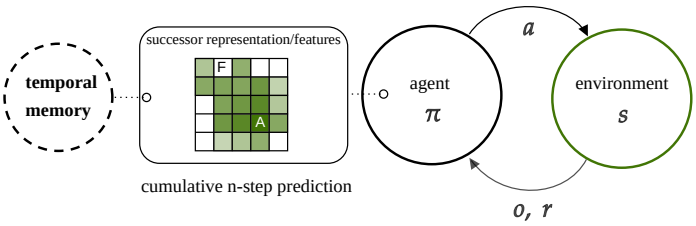

Рисунок 3.2 — Архитектура агента.

Архитектура агента представляется как универсальная система, состоящая из двух ключевых компонентов: модели мира и механизма принятия решений, основанного на её предсказаниях (см. рисунок 3.2). Модель мира — например TPC — играет центральную роль, поддерживая компактное латентное представление окружающей среды и позволяя предсказывать её развитие во времени.

На каждом шаге агент получает новое наблюдение и передаёт его в модель мира, которая обновляет своё внутреннее латентное состояние. Это состояние отражает текущее представление агента о ситуации — с учётом как предыдущих наблюдений, так и последних действий. Используя изученную динамику, модель может прогнозировать, как будет развиваться ситуация в будущем для каждого возможного действия. Эти предсказания формируют внутренние «прогоны» — воображаемые сценарии, описывающие, как может меняться состояние системы. Такие сценарии можно рассматривать как аналоги successor representation — обобщённого прогноза будущих состояний при заданной стратегии поведения.

Вместо того чтобы оценивать действия напрямую по исходным наблюдениям, агент преобразует предсказанные латентные состояния в ожидаемое вознаграждение с помощью обученной модели. Каждый элемент латентного пространства вносит свой вклад в общее вознаграждение, и эти вклады суммируются вдоль всей предсказанной траектории, чтобы получить ожидаемую итоговую награду. Такой подход позволяет агенту быстро адаптироваться к новым задачам или изменяющимся условиям, не переобучая модель мира.

Чтобы выбрать действие, агент оценивает каждую возможную опцию, предсказывает её результат через внутреннюю симуляцию, ранжирует действия

по ожидаемой награде и делает выбор, применяя softmax-функцию. Этот процесс планирования повторяется на каждом шаге, благодаря чему агент может гибко адаптироваться и учитывать долгосрочные последствия своих решений, опираясь на предсказательную память.

TPC особенно хорошо подходит для такой памяти: оно вдохновлено биологическими механизмами и эффективно с вычислительной точки зрения. Его латентные состояния обучаются через минимизацию локальной ошибки предсказания и естественным образом отражают временные зависимости. Благодаря этому TPC отлично справляется с ролью «внутреннего симулятора» и помогает агенту принимать обоснованные решения в условиях частичной доступности информации.

Глава 4. Эксперименты

4.1 Распределенная Хеббовская временная память (DHTM)

4.1.1 Среды и сценарий обучения

DHTM разработана для обучения в реальном времени. Это особенно полезно для алгоритмов в задачах обучения с подкреплением, где адаптация имеет решающее значение. Поэтому последовательности для тестирования алгоритма сегментов были собраны в средах RL, тех же, которые использовались для тестирования DHTM: Pinball и AnimalAI.

Pinball — это двумерная среда, созданная с использованием игрового движка Godot [58] (Рис. 4.1-A). Она состоит из шара, поля, окружённого стенами, и круговой зоны в центре, способной отклонять шар в случайных направлениях, добавляя стохастичность в среду. В начале каждого эпизода шар получает начальный импульс и продолжает свободно перемещаться, взаимодействуя со средой.

Другой набор наблюдений был получен из среды AnimalAI [59] (Рис. 4.1-B) с использованием фиксированной стратегии. AnimalAI — это настраиваемая трёхмерная среда, специально созданная для реализации различных психологических тестов для животных.

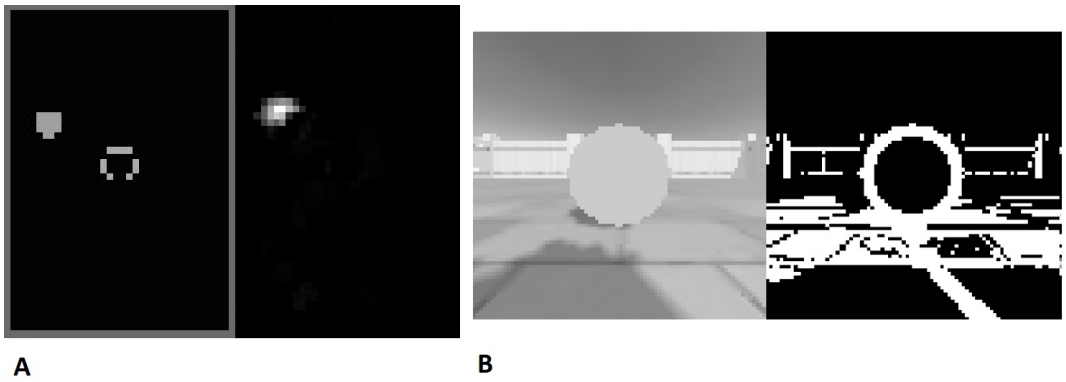

Рисунок 4.1 — Примеры исходных выходных данных и их обработанная версия (справа): A. среда Pinball, B. AnimalAI.

Исходный выходной сигнал каждого шага эпизода в этих средах представляет собой RGB-изображение, которое предварительно обрабатывается таким образом, чтобы моделировать выходной сигнал на основе событий. Для тестирования устойчивости модели также добавляется гауссовский шум в исходное изображение. Этот шум вводит случайность наблюдений, моделируя реальные условия, при которых данные могут содержать некоторый уровень шума. Обработанное изображение затем передаётся в тот же кодировщик с механизмом "победитель берёт всё"(WTA), который использовался в экспериментах DHTM. Выход кодировщика представляет собой 5 и 6 категориальных случайных переменных для сред Pinball и AnimalAI соответственно. Эти переменные затем передаются во временную память.

4.1.2 Результаты в средах Pinball и AnimalAI

Для каждой среды был определен оптимальный размер фактора и порог для сравнения качества новой вычислительной модели сегментов с лучшей старой моделью. В качестве метрик производительности использовались отрицательное логарифмическое правдоподобие наблюдений (surprise) и количество сегментов [60]. Низкие значения surprise и количества сегментов соответствуют лучшей и более эффективной модели.

Мы сравнили три сценария при разном уровне шума:

(1) Оптимально определённый размер фактора и порогов для исходных сегментов DHTM — красная линия.

(2) Максимальный размер фактора и уменьшенные пороги, т.е. проверка идеи из раздела 2.3 — синяя линия.

(3) Максимальный размер фактора, уменьшенный порог на этапе распространения убеждений (раздел 2.3) и адаптивный порог, предложенный в разделе 2.4 для этапа роста сегментов — зелёная кривая.

Результаты для сред Pinball и AnimalAI представлены на рисунках 4.2 и 4.3 соответственно.

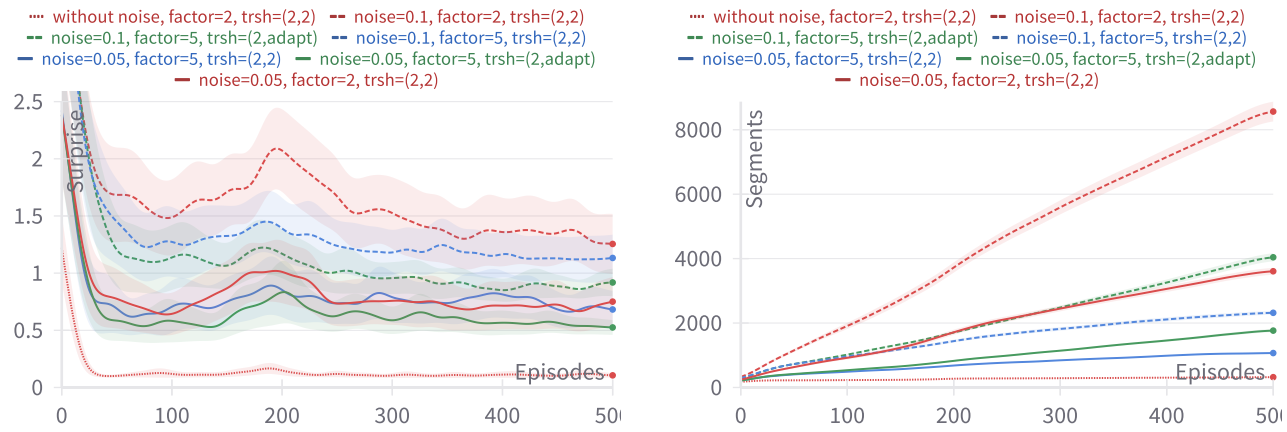

Рисунок 4.2 — Среда Pinball. Разные цвета соответствуют различным вариантам размера фактора и порогов: $factor = 2, trsh = (2,2)$ — оптимальный фактор для исходной версии DHTM; $factor = 5, trsh = (2,2)$ — фиксированный порог (раздел 2.3); $factor = 5, trsh = (2, adapt)$ — адаптивный порог (раздел 2.4). Различия в стиле линии показывают уровни шума согласно легенде.

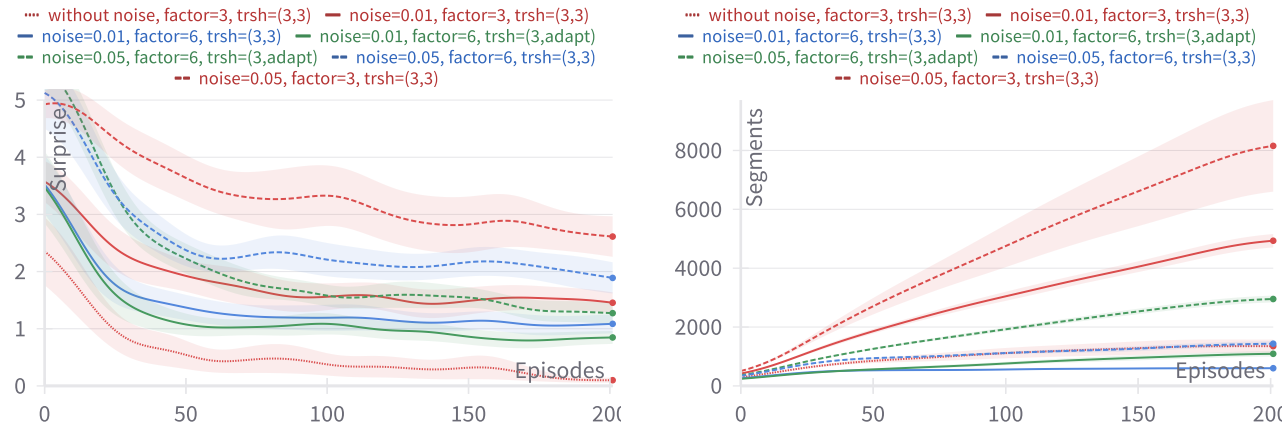

Рисунок 4.3 — Среда AnimalAI. Различие в цвете соответствует вариантам размера фактора и порогов, аналогично среде Pinball. Стиль линий обозначает различные уровни шума.

4.1.3 Сравнение с другими методами

Более того, было проведено сравнение исходной DHTM (1 - красная линия), адаптивной версии (3 - зелёная линия) и исходной DHTM с применением

обрезки сегментов (жёлтая линия). Обрезка сегментов основывалась на максимальном количестве сегментов, установленном в соответствии с числом сегментов на последнем измеренном эпизоде для адаптивной версии. Сегменты для обрезки выбирались на основе частоты их использования. Соответствующие результаты представлены на рисунках 4.4 и 4.5. Эти результаты демонстрируют принципиальное различие между предлагаемым методом и обрезкой. Достигается не только сокращение числа сегментов, но и улучшение качества, тогда как обрезка только ухудшает его.

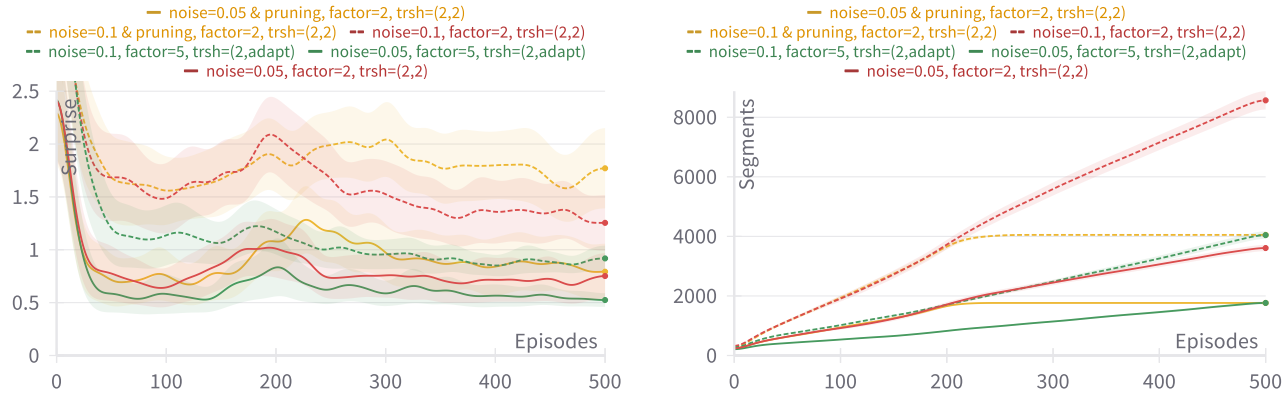

Рисунок 4.4 — Среда Pinball. Сравнение исходной DHTM (красная линия), адаптивной версии (зелёная линия) и исходной DHTM с обрезкой сегментов (жёлтая линия). Стиль линий соответствует различным уровням шума.

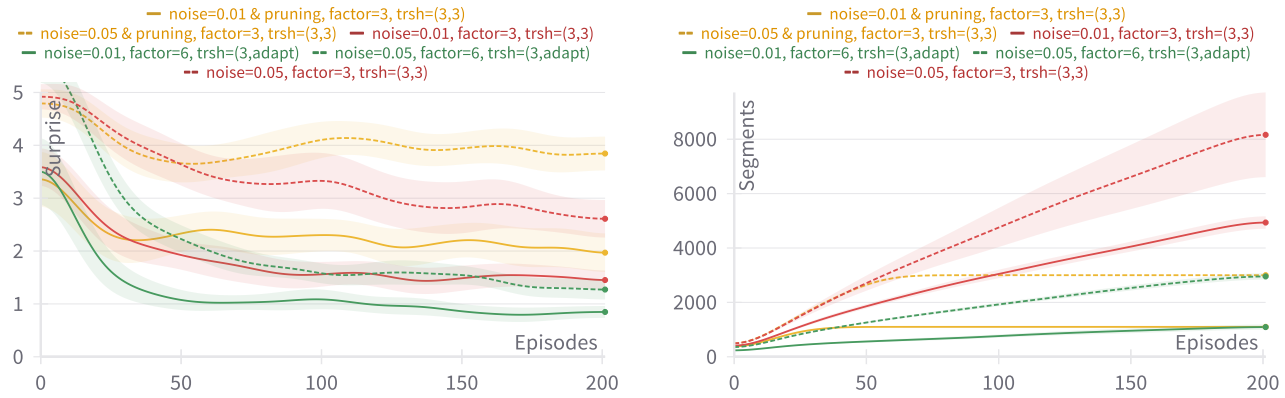

Рисунок 4.5 — Среда AnimalAI. Сравнение исходной DHTM, адаптивной версии и исходной DHTM с обрезкой сегментов. Стиль линий соответствует различным уровням шума.

Таким образом, удалось значительно сократить количество сегментов, используя пороги ниже размера фактора в условиях шума, что также улучшило

качество модели. Это означает, что использование большего размера фактора с более низким порогом повышает устойчивость модели к шуму. Графики также показывают, что адаптивный порог приводит к более устойчивой модели, хотя слегка увеличивает количество выращиваемых сегментов по сравнению с фиксированным порогом.

4.2 Временное предсказательное кодирование (TPC)

В данной главе была исследована эффективность TPC в качестве метода формирования латентных представлений для агентов в задачах обучения с подкреплением (RL) в частично наблюдаемых средах. Эксперименты проводились в специально разработанных средах типа "grid-world"(сетчатый мир) с разными уровнями наблюдаемой неоднозначности.

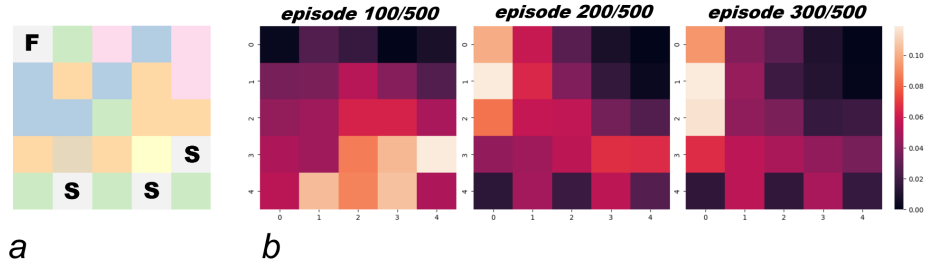

Рисунок 4.6 — (a) Среда RL с случайными начальными позициями (S) и случайной целью (F); (b) Карты частоты посещения агентом клеток на эпизодах 100, 200 и 300.

Были исследованы два экспериментальных сценария:

Случайные траектории (фиксированная стратегия): Проводились случайные блуждания в мире размером 5×5. Каждой клетке случайным образом присваивался один из нескольких цветов, при этом цветов было меньше, чем клеток, что создавало неоднозначность наблюдений. Для сбора траекторий использовалась равномерная случайная стратегия. Такая стационарная стратегия, являясь более простой по сравнению с полноценным обучением RL, позволяет изолированно изучить способность TPC предсказывать следующее наблюдение агента только на основе исторического контекста, без влияния сигналов вознаграждения и смещения, вызванного процессом сбора траекторий.

RL-сценарий (выученная стратегия): Использовалась аналогичная среда 5×5, содержащая одну случайную целевую клетку (дающую вознаграждение) и три случайно выбранные стартовые клетки, обозначенные одинаковым цветом. Остальные клетки были случайно окрашены в один из шести возможных цветов. В отличие от первого сценария, здесь агент обучался стратегии с помощью RL, стремясь пройти от случайной стартовой позиции к цели, максимизируя суммарное вознаграждение.

Пример такой среды представлен на рис. 4.6(а).

4.2.1 Точность предсказаний в случайных траекториях

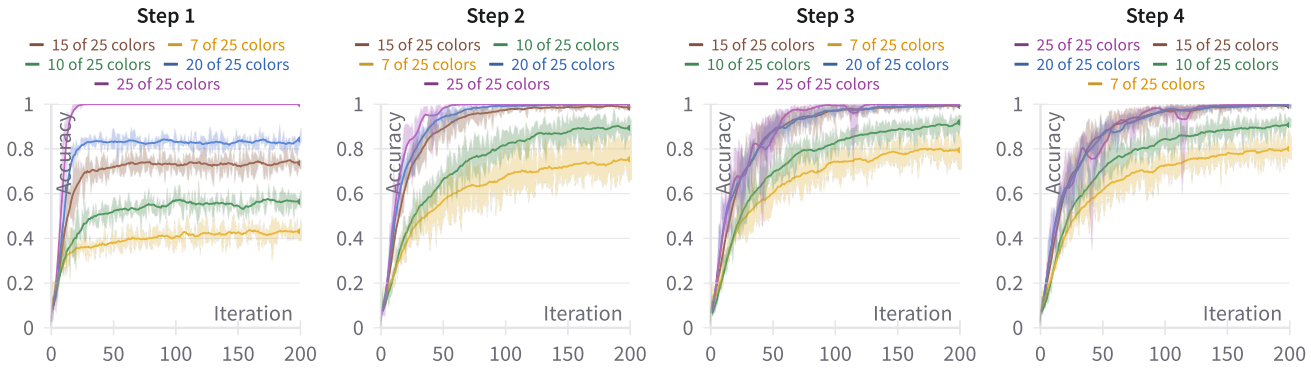

Рисунок 4.7 — Точность предсказаний в случайных блужданиях при разном количестве цветов (25, 20, 15, 10, 7). Точность увеличивается с каждым шагом в рамках траектории.

Сначала была проанализирована точность предсказаний TPC в сценарии с случайными траекториями. На рис. 4.7 представлена точность предсказаний на первых четырёх шагах переходов эпизода для разных уровней неоднозначности (с использованием 25, 20, 15, 10 и 7 цветов). Наблюдается явное улучшение точности с каждым последующим шагом, что свидетельствует о том, что TPC эффективно использует предыдущие наблюдения для устранения неоднозначности текущего состояния.

4.2.2 Точность предсказаний в RL-сценарии

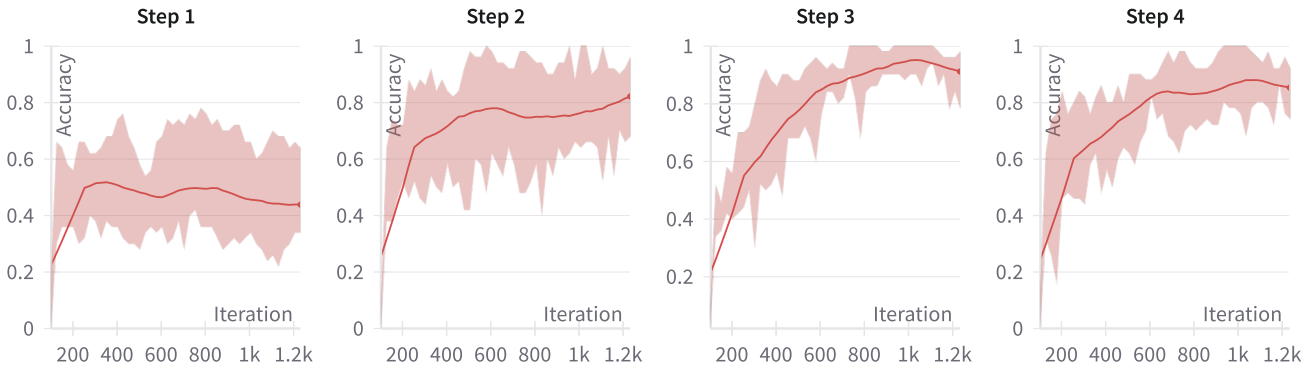

Рисунок 4.8 — Точность предсказаний в RL-сценарии на первых 4 шагах. Точность возрастает с каждым шагом.

Теперь рассмотрим RL-сценарий, где агент активно перемещался к цели. На рис. 4.8 представлена точность предсказаний на первых четырёх шагах каждого эпизода. Снова наблюдается быстрое улучшение точности по мере продвижения агента, что подтверждает успешное обучение представлениям состояний, важным для навигации в частично наблюдаемой среде.

4.2.3 Сравнение с другими моделями в RL

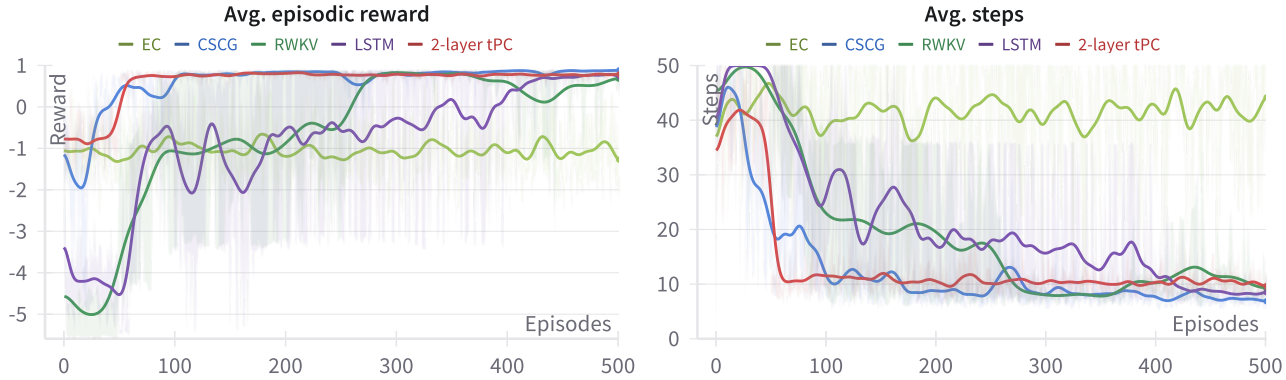

Рисунок 4.9 — Сравнение агентов на основе TPC, LSTM, RWKV, CSCG и эпизодического контроля. Слева: среднее вознаграждение за эпизод. Справа: среднее количество шагов до цели.

Были сравнены агентов на основе TPC с базовыми моделями — LSTM, RWKV, CSCG и эпизодическим контролем — по суммарному вознаграждению и числу шагов до цели. Рис. 4.9 показывает, что TPC стабильно демонстрирует сравнимые или лучшие результаты.

Эти эксперименты подтверждают потенциал моделей TPC как эффективного метода представлений в RL.

4.2.4 Кластеризация латентных представлений и обобщающая способность

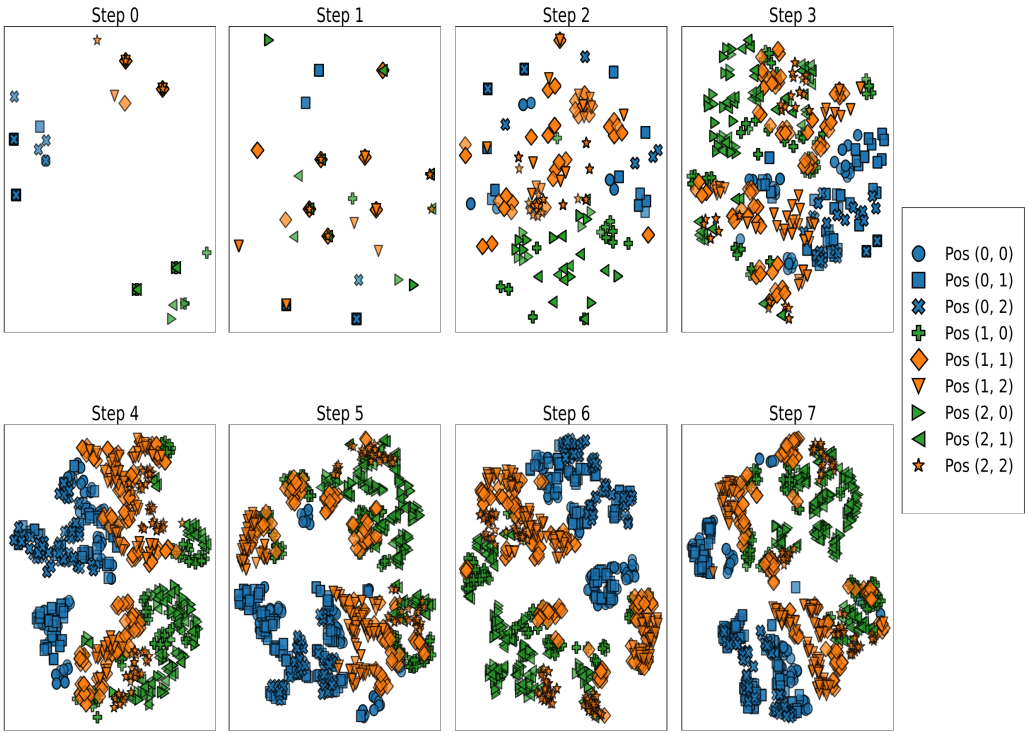

Рисунок 4.10 — Визуализация латентных состояний методом t-SNE на шагах с 0 по 8 в сетке 3×3. Форма маркера показывает истинную позицию; цвет обозначает наблюдаемый цвет клетки. Латентные состояния постепенно формируют кластеры, соответствующие истинным позициям.

Были проведены дополнительные эксперименты в меньшей среде 3×3 с тремя цветами, где агент выполнял случайные блуждания и пытался предсказать свои последующие наблюдения. Затем было проанализировано, насколько

латентные состояния, изученные моделью TPC, эффективно представляют истинные, скрытые от агента позиции в сетке.

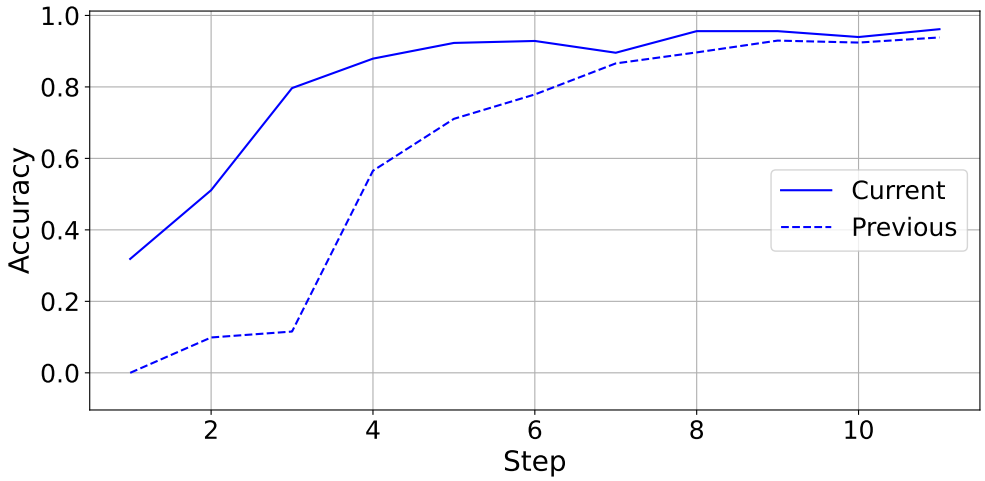

Рисунок 4.11 — Точность обученной логистической регрессии при предсказании истинной позиции по латентному состоянию на данных с временных шагов в рамках траекторий. Обе линии представляют модели, обученные на данных текущего шага: сплошная линия показывает точность на том же шаге, пунктирная линия — на предыдущем.

На рис. 4.10 показана кластеризация методом t-SNE примерно 5000 латентных состояний, полученных на шагах траекторий с 0 по 8. Каждая уникальная форма маркера соответствует истинной позиции агента на сетке, а цвет показывает наблюдаемый цвет клетки. Со временем латентные состояния всё более чётко группируются в кластеры, соответствующие истинным позициям на сетке, демонстрируя способность TPC внутренне различать и представлять состояния, несмотря на частичную наблюдаемость.

На рис. 4.11 представлены результаты обучения классификаторов на основе логистической регрессии на латентных состояниях с целью предсказания истинной позиции на сетке. Сплошная линия демонстрирует точность классификации, когда обучение и тестирование проводятся на одном и том же временном шаге траектории (шаги от 1 до 11), приближаясь к практически идеальной точности. Пунктирная линия показывает точность, когда классификатор, обученный на шаге n, тестируются на шаге $n-1$. Сокращение разрыва между этими кривыми со временем указывает на то, что латентные представления становятся всё более обобщёнными и устойчивыми, отражая стабильные структуры состояний, выходящие за рамки текущего временного контекста.

Эти результаты подчёркивают потенциал модели TPC в качестве высокоэффективного метода внутренних представлений для обучения с подкреплением в частично наблюдаемых средах, мотивируя дальнейшие исследования в более сложных условиях.

Заключение

В настоящей работе исследованы перспективные подходы к моделированию временной памяти и обработке последовательной информации в условиях частичной наблюдаемости, являющихся ключевыми задачами в области искусственного интеллекта и когнитивных наук. Основное внимание уделено двум оригинальным подходам: распределённой хеббовской временной памяти (DHTM) и временному предсказательному кодированию (TPC).

Модель DHTM, являясь улучшенной версией вероятностной эпизодической памяти, сочетает графовые модели, аналогичные факторным скрытым марковским моделям, с нейронной архитектурой, схожей с иерархической временной памятью (HTM). В рамках диссертации предложены и исследованы новые методы объединения и адаптивного роста сегментов, позволяющие существенно повысить устойчивость модели к шумам и уменьшить избыточность сегментов без потери качества предсказаний. Эксперименты в средах Pinball и AnimalAI подтвердили эффективность адаптивного подхода в сравнении с классическими методами, показав не только улучшение качества модели, но и повышение её устойчивости к шумовым воздействиям.

Временное предсказательное кодирование (TPC) интегрирует принципы нейробиологически мотивированного предсказательного кодирования с задачами обучения с подкреплением (RL). В работе была исследована возможность использования TPC как метода формирования латентных представлений для RL-агентов в частично наблюдаемых средах. Эксперименты, проведённые в специально разработанных средах типа grid-world с разной степенью наблюдаемой неоднозначности, показано, что TPC эффективно формирует устойчивые и обобщённые латентные представления, позволяющие агенту успешно решать задачи навигации и планирования. При сравнении с другими подходами, такими как LSTM, RWKV, CSCG и EC-agent, агент на основе TPC демонстрировал стабильно высокие результаты, подтверждая превосходство или, по меньшей мере, сопоставимость с лучшими из существующих моделей.

Дополнительный анализ латентных представлений с помощью методов визуализации (t-SNE) и классификации подтвердил способность TPC эффективно дифференцировать состояния среды даже при наличии сильной наблюдаемой

неоднозначности. Латентные представления постепенно формировали устойчивые и генерализуемые структуры, соответствующие реальным состояниям агента, что является ключевым фактором для успешного применения метода в реальных задачах.

Таким образом, результаты проведённых исследований подтверждают значительный потенциал предложенных моделей DHTM и TPC в качестве мощных и эффективных подходов к обработке последовательной информации и представлению состояний в частично наблюдаемых средах. Дальнейшие исследования могут быть направлены на адаптацию и интеграцию этих методов в более сложные и реалистичные сценарии, а также на изучение дополнительных механизмов повышения их производительности и био-правдоподобности.

Список литературы

1. Neuroscience-inspired artificial intelligence / Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, Matthew Botvinick // *Neuron*. — 2017. — Vol. 95, no. 2. — Pp. 245–258.

2. *Chowdhary KR1442, Chowdhary KR.* Natural language processing // *Fundamentals of artificial intelligence*. — 2020. — Pp. 603–649.

3. *Lara-Benítez Pedro, Carranza-García Manuel, Riquelme José C.* An experimental review on deep learning architectures for time series forecasting // *International journal of neural systems*. — 2021. — Vol. 31, no. 03. — P. 2130001.

4. Model-based reinforcement learning: A survey / Thomas M Moerland, Joost Broekens, Aske Plaat et al. // *Foundations and Trends® in Machine Learning*. — 2023. — Vol. 16, no. 1. — Pp. 1–118.

5. Building machines that learn and think like people / Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, Samuel J Gershman // *Behavioral and brain sciences*. — 2017. — Vol. 40. — P. e253.

6. Recent advances in recurrent neural networks / Hojjat Salehinejad, Sharan Sankar, Joseph Barfett et al. // *arXiv preprint arXiv:1801.01078*. — 2017.

7. *Hochreiter Sepp, Schmidhuber Jürgen.* Long short-term memory // *Neural computation*. — 1997. — Vol. 9, no. 8. — Pp. 1735–1780.

8. An overview and comparative analysis of recurrent neural networks for short term load forecasting / Filippo Maria Bianchi, Enrico Maiorino, Michael C Kampffmeyer et al. // *arXiv preprint arXiv:1705.04378*. — 2017.

9. Rwkv: Reinventing rnns for the transformer era / Bo Peng, Eric Alcaide, Quentin Anthony et al. // *arXiv preprint arXiv:2305.13048*. — 2023.

10. *Eddy Sean R.* What is a hidden Markov model? // *Nature biotechnology*. — 2004. — Vol. 22, no. 10. — Pp. 1315–1316.

11. *Kaelbling Leslie Pack, Littman Michael L, Cassandra Anthony R.* Planning and acting in partially observable stochastic domains // *Artificial intelligence.* — 1998. — Vol. 101, no. 1-2. — Pp. 99–134.

12. Point-based value iteration: An anytime algorithm for POMDPs / Joelle Pineau, Geoff Gordon, Sebastian Thrun et al. // Ijcai. — Vol. 3. — 2003. — Pp. 1025–1032.

13. *Hausknecht Matthew J, Stone Peter.* Deep Recurrent Q-Learning for Partially Observable MDPs. // AAAI fall symposia. — Vol. 45. — 2015. — P. 141.

14. *French Robert M.* Catastrophic forgetting in connectionist networks // *Trends in cognitive sciences.* — 1999. — Vol. 3, no. 4. — Pp. 128–135.

15. *Friston Karl.* The free-energy principle: a unified brain theory? // *Nature reviews neuroscience.* — 2010. — Vol. 11, no. 2. — Pp. 127–138.

16. *Bi Guo-qiang, Poo Mu-ming.* Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type // *Journal of neuroscience.* — 1998. — Vol. 18, no. 24. — Pp. 10464–10472.

17. *Caporale Natalia, Dan Yang.* Spike timing–dependent plasticity: a Hebbian learning rule // *Annu. Rev. Neurosci.* — 2008. — Vol. 31, no. 1. — Pp. 25–46.

18. *Olshausen Bruno A, Field David J.* Sparse coding with an overcomplete basis set: A strategy employed by V1? // *Vision research.* — 1997. — Vol. 37, no. 23. — Pp. 3311–3325.

19. *Urbanczik Robert, Senn Walter.* Learning by the dendritic prediction of somatic spiking // *Neuron.* — 2014. — Vol. 81, no. 3. — Pp. 521–528.

20. A deep learning framework for neuroscience / Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin et al. // *Nature neuroscience.* — 2019. — Vol. 22, no. 11. — Pp. 1761–1770.

21. Hierarchical temporal memory based on spin-neurons and resistive memory for energy-efficient brain-inspired computing / Deliang Fan, Mrigank Sharad, Abhronil Sengupta, Kaushik Roy // *IEEE transactions on neural networks and learning systems.* — 2015. — Vol. 27, no. 9. — Pp. 1907–1919.

22. *Rao Rajesh PN, Ballard Dana H.* Predictive Coding in the Visual Cortex: A Functional Interpretation of Some Extra-classical Receptive-field Effects // *Nature Neuroscience.* — 1999. — Vol. 2, no. 1. — Pp. 79–87.

23. *Friston Karl.* A Theory of Cortical Responses // *Philosophical Transactions of the Royal Society B: Biological Sciences.* — 2005. — Vol. 360, no. 1456. — Pp. 815–836.

24. *Dzhivelikian Evgenii, Kuderov Petr, Panov Aleksandr I.* Learning Hidden Markov Model of Stochastic Environment with Bio-Inspired Probabilistic Temporal Memory // Biologically Inspired Cognitive Architectures 2023. Studies in Computational Intelligence. — Vol. 1130. — Pp. 330–339.

25. *Ghahramani Zoubin, Jordan Michael.* Factorial hidden Markov models // *Advances in neural information processing systems.* — 1995. — Vol. 8.

26. *Tang Edwin, Mattar Marcelo G., Daw Nathaniel D.* Sequential Memory with Temporal Predictive Coding // *arXiv preprint arXiv:2306.09852.* — 2023.

27. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps / Dileep George, Rajeev V Rikhye, Nishad Gothoskar et al. // *Nature communications.* — 2021. — Vol. 12, no. 1. — P. 2392.

28. Neural episodic control / Alexander Pritzel, Benigno Uria, Sriram Srinivasan et al. // International conference on machine learning / PMLR. — 2017. — Pp. 2827–2836.

29. *Friston Karl.* A theory of cortical responses // *Philosophical transactions of the Royal Society B: Biological sciences.* — 2005. — Vol. 360, no. 1456. — Pp. 815–836.

30. *Lisman John, Redish A David.* Prediction, sequences and the hippocampus // *Philosophical Transactions of the Royal Society B: Biological Sciences.* — 2009. — Vol. 364, no. 1521. — Pp. 1193–1201.

31. Advancing neuromorphic computing with loihi: A survey of results and outlook / Mike Davies, Andreas Wild, Garrick Orchard et al. // *Proceedings of the IEEE.* — 2021. — Vol. 109, no. 5. — Pp. 911–934.

32. *Eichenbaum Howard.* Time cells in the hippocampus: a new dimension for mapping memories // *Nature Reviews Neuroscience.* — 2014. — Vol. 15, no. 11. — Pp. 732–744.

33. *Rabiner Lawrence, Juang Biinghwang.* An introduction to hidden Markov models // *ieee assp magazine.* — 1986. — Vol. 3, no. 1. — Pp. 4–16.

34. Empirical evaluation of gated recurrent neural networks on sequence modeling / Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio // *arXiv preprint arXiv:1412.3555.* — 2014.

35. Robust echo state network with sparse online learning / Cuili Yang, Kaizhe Nie, Junfei Qiao, Danlei Wang // *Information Sciences.* — 2022. — Vol. 594. — Pp. 95–117.

36. A digital liquid state machine with biologically inspired learning and its application to speech recognition / Yong Zhang, Peng Li, Yingyezhe Jin, Yoonsuck Choe // *IEEE transactions on neural networks and learning systems.* — 2015. — Vol. 26, no. 11. — Pp. 2635–2649.

37. S4nd: Modeling images and videos as multidimensional signals with state spaces / Eric Nguyen, Karan Goel, Albert Gu et al. // *Advances in neural information processing systems.* — 2022. — Vol. 35. — Pp. 2846–2861.

38. Unsupervised real-time anomaly detection for streaming data / Subutai Ahmad, Alexander Lavin, Scott Purdy, Zuha Agha // *Neurocomputing.* — 2017. — Vol. 262. — Pp. 134–147. — Online Real-Time Learning Strategies for Data Streams. URL: https://www.sciencedirect.com/science/article/pii/S0925231217309864.

39. Improved Hierarchical Temporal Memory for Online Prediction of Ocean Time Series Data / Tianao Qin, Ruixin Chen, Rufu Qin, Yang Yu // *Journal of Marine Science and Engineering.* — 2024. — Vol. 12, no. 4. — P. 574.

40. ELM-HTM guided bio-inspired unsupervised learning for anomalous trajectory classification / Arif Ahmed Sekh, Debi Prosad Dogra, Samarjit Kar et al. // *Cognitive Systems Research.* — 2020. — Vol. 63. — Pp. 30–41.

41. *Bengio Yoshua, Lee Dong-Hyun, Bornschein Jorg et al.* Towards Biologically Plausible Deep Learning. — 2016.

42. Avoiding Catastrophe: Active Dendrites Enable Multi-Task Learning in Dynamic Environments / Abhiram Iyer, Karan Grewal, Akash Velu et al. // *Frontiers in Neurorobotics*. — 2022. — Vol. 16. — URL: https://www.frontiersin.org/articles/10.3389/fnbot.2022.846219.

43. *Hopfield John J.* Neural networks and physical systems with emergent collective computational abilities. // *Proceedings of the national academy of sciences*. — 1982. — Vol. 79, no. 8. — Pp. 2554–2558.

44. *Krotov Dmitry, Hopfield John J.* Dense associative memory for pattern recognition // *Advances in neural information processing systems*. — 2016. — Vol. 29.

45. Cloob: Modern hopfield networks with infoloob outperform clip / Andreas Fürst, Elisabeth Rumetshofer, Johannes Lehner et al. // *Advances in neural information processing systems*. — 2022. — Vol. 35. — Pp. 20450–20468.

46. Hopfield networks is all you need / Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner et al. // *arXiv preprint arXiv:2008.02217*. — 2020.

47. Associative memories via predictive coding / Tommaso Salvatori, Yuhang Song, Yujian Hong et al. // *Advances in neural information processing systems*. — 2021. — Vol. 34. — Pp. 3874–3886.

48. Graph attention networks / Petar Veličković, Guillem Cucurull, Arantxa Casanova et al. // *arXiv preprint arXiv:1710.10903*. — 2017.

49. *Ding Shifei, Xu Xinzheng, Nie Ru.* Extreme learning machine and its applications // *Neural Computing and Applications*. — 2014. — Vol. 25. — Pp. 549–556.

50. Hierarchical intrinsically motivated agent planning behavior with dreaming in grid environments / Evgenii Dzhivelikian, Artem Latyshev, Petr Kuderov, Aleksandr I Panov // *Brain Informatics*. — 2022. — Vol. 9, no. 1. — P. 8.

51. *Dzhivelikian Evgenii, Kuderov Petr, Panov Aleksandr I.* Learning Successor Representations with Distributed Hebbian Temporal Memory // *arXiv preprint arXiv:2310.13391*. — 2023.

52. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps / Dileep George, Rajeev V. Rikhye,

Nishad Gothoskar et al. // *Nature Communications*. — 2021. — Apr. — Vol. 12, no. 1. — P. 2392. — URL: https://doi.org/10.1038/s41467-021-22559-5.

53. Understanding belief propagation and its generalizations / Jonathan S Yedidia, William T Freeman, Yair Weiss et al. // *Exploring artificial intelligence in the new millennium*. — 2003. — Vol. 8, no. 236-239. — Pp. 0018–9448.

54. *Duff Iain S, Erisman Albert Maurice, Reid John Ker*. Direct methods for sparse matrices. — Oxford University Press, 2017.

55. *Lotter William, Kreiman Gabriel, Cox David*. Deep predictive coding networks for video prediction and unsupervised learning // *arXiv preprint arXiv:1605.08104*. — 2016.

56. *van den Oord Aaron, Li Yazhe, Vinyals Oriol*. Representation Learning with Contrastive Predictive Coding // NeurIPS. — 2018. — Pp. 1–12.

57. Learning predictive state representations / Satinder P Singh, Michael L Littman, Nicholas K Jong et al. // Proceedings of the 20th International Conference on Machine Learning (ICML-03). — 2003. — Pp. 712–719.

58. Godot reinforcement learning agents / Edward Beeching, Jilles Debangoye, Olivier Simonin, Christian Wolf // *arXiv preprint arXiv:2112.03636*. — 2021.

59. The animal-AI testbed and competition / Matthew Crosby, Benjamin Beyret, Murray Shanahan et al. // Neurips 2019 competition and demonstration track / PMLR. — 2020. — Pp. 164–176.

60. *Rodkin Ivan, Kuderov Petr, Panov Aleksandr I.* Stability and Similarity Detection for the Biologically Inspired Temporal Pooler Algorithms. — Vol. 213. — Pp. 570–579.