

ГЕНЕРАТИВНАЯ СИМУЛЯЦИЯ ИГРОВОГО ОКРУЖЕНИЯ В РЕАЛЬНОМ ВРЕМЕНИ

Э. С. Большаков¹ [0000-0002-2208-9515], В. В. Кугуракова² [0000-0002-1552-4910]

^{1, 2}Институт информационных технологий и интеллектуальных систем
Казанского федерального университета

¹edward.bolshakov117@gmail.com, ²vlada.kugurakova@gmail.com

Аннотация

Рассмотрены возможности генеративных нейросетевых симуляций с фокусом на применении методов обучения с подкреплением и нейросетевых мировых моделей для создания интерактивных миров. Описаны ключевые достижения в области обучения агентов с использованием обучения с подкреплением. Особое внимание уделено нейросетевым моделям мира, а также генеративным моделям, таким как *Oasis*, *DIAMOND*, *Genie* и *GameNGen*, использующим диффузионные сети для создания реалистичных и интерактивных игровых миров. Рассмотрены возможности и ограничения моделей генеративных симуляций, такие как проблемы с аккумуляцией ошибки и ограничениями памяти, а также их влияние на качество генерации. В заключении названы темы дальнейших исследований.

Ключевые слова: видеоигры, игровое окружение, генеративная симуляция, обучение с подкреплением, генеративные нейросети, симуляция игрового процесса, мировые модели.

ВВЕДЕНИЕ

Развитие технологий обучения с подкреплением и нейросетевых мировых моделей открыло новые горизонты в области генеративных нейросетей. Особенно интересен прогресс в области видеоигровой индустрии, так как видеоигры являются отличным примером виртуального окружения, которое может использоваться для обучения агентов определенному поведению.

На текущий момент для разработки видеоигр в основном используют готовые игровые движки¹, такие как *Unreal Engine*², *Unity*³ или *Godot*⁴, которые уже стали стандартом игровой индустрии. Они берут на себя основные задачи по симуляции физики, просчету столкновений и отрисовке игровой графики, а основная логика разрабатывается традиционным программированием в рамках этих готовых инструментов.

Модели генеративных симуляций способны произвести революцию в устоявшемся подходе, заменив собой основные функции игровых движков. Использование этих моделей позволит генерировать интерактивные игровые миры так же просто, как это сейчас делают модели генерации изображений по текстовым описаниям. За последние несколько лет было представлено несколько моделей, способных полноценно симулировать игровой процесс, и они уже показывают многообещающие результаты.

В настоящей работе проанализированы перспективы использования генеративных нейросетевых симуляций. В первом разделе представлены обзор методов обучения с подкреплением, их эволюция и связь с генеративными симуляциями. Во втором разделе описаны нейросетевые модели мира как продолжение исследований в области обучения с подкреплением, применяемые для моделирования виртуальных миров. В третьем разделе рассмотрены генеративные модели, которые применяют диффузионные сети для разработки реалистичных и интерактивных игровых пространств. В четвёртом разделе показаны ограничения генеративных симуляционных моделей, в частности, отмечены проблемы накопления ошибок и ограничения по памяти, а также их влияние на качество генерации. В заключении выделены направления будущих исследований.

¹ Игровой движок (англ. game engine) – программное обеспечение, предназначенное для разработки и создания видеоигр и представляющее собой комплекс программных компонентов, обеспечивающих основные функциональные возможности игры: визуализацию графики, физическое моделирование, управление памятью, воспроизведение звука, работу скриптов, анимацию, искусственный интеллект, сетевой код, управление игровым процессом и др. Современные игровые движки, такие как Unreal Engine, Unity, CryEngine, являются интегрированными средами разработки, предоставляющими набор инструментов для создания интерактивного контента.

² <http://unrealengine.com/>

³ <https://unity.com/>

⁴ <https://godotengine.org/>

RL-АГЕНТЫ

Создание генеративных симуляций в первую очередь обязано прогрессу в области обучения с подкреплением (Reinforcement Learning, далее в статье RL) [1]. Это область машинного обучения, в которой агент обучается принимать решения, взаимодействуя с окружающей средой и получая обратную связь в виде вознаграждений или наказаний. Цель RL-агента – максимизировать суммарное вознаграждение, выполняя оптимальные действия в различных ситуациях.

С момента своего возникновения RL претерпело значительные изменения. Ранние методы, такие как Q-обучение, позволяли агентам обучаться на основе таблиц Q-значений. Однако с развитием глубоких нейронных сетей появились более мощные алгоритмы, такие как Deep Q-Networks (DQN), которые используют глубокие нейросети для приближения Q-функций [2]. Это позволило значительно улучшить производительность агентов в сложных задачах.

Последующее развитие привело к созданию алгоритмов, способных обучаться без использования человеческих данных. Например, *AlphaGo Zero*, разработанный DeepMind [3], обучался, играя против самого себя, и достиг уровня игры, превосходящего предыдущие версии, включая *AlphaGo*, которая для обучения использовала выборку человеческих партий [4].

Яркими примерами RL-агентов, обученных на видеоиграх, являются также *AlphaStar* от DeepMind – агент, обучившийся играть в *StarCraft II*⁵ на выборке из записей профессиональных игр [5], и *OpenAI Five* – агент, играющий в *Dota 2*, обученный компанией OpenAI на аналогичном датасете [6]. При этом вторая модель способна моделировать игру сразу 5 человек – именно столько человек находятся в одной из двух противостоящих команд в единичной игровой сессии *Dota 2*. Подобные обученные RL-агенты могут также использоваться в тестировании видеоигр, где они способны эффективно заменить обычных игроков в задачах тестирования игрового баланса [7, 8].

⁵ Все упомянутые видеоигры приведены в разделе Лудография.

Моделирование взаимодействия агента с окружающей средой

В задачах обучения с подкреплением для моделирования процессов обучения и взаимодействия агента с окружающей средой используется модель марковского процесса принятия решений – МППР (Markov Decision Process – MDP) [9].

Агент наблюдает определенное состояние (S) окружающей среды, затем совершает определенные действия (A), взаимодействуя с этой средой, и получает вознаграждение (R). В зависимости от действий агента состояние среды изменяется с вероятностью (P). Целью оптимизации для МППР является поиск оптимальной стратегии (π), которая представляет собой функцию зависимости действий агента от текущего состояния окружающей среды [10].

Большинство моделей, рассматриваемых в этой статье, использует обобщенную модель МППР – частично наблюдаемый марковский процесс принятия решений. В этом обобщении агент наблюдает не полноценное состояние окружающей среды, а использует сенсорную модель – распределение вероятностей наблюдений (O) для определенного состояния. Стратегия такого процесса уже сопоставляет последовательность наблюдений с действием агента [11].

Сбор данных с помощью RL-агентов

Для создания правдоподобной симуляции, с которой будет взаимодействовать игрок, обучение должно происходить на выборке записей игры реальных людей. Поскольку получить достаточно большой датасет таких записей может быть довольно проблематичным, можно с этой целью обучить RL-агента, приближенно имитирующего действия реального человека. Такой подход позволяет автоматизировать процесс сбора данных – записей геймплея одновременно с записями действий RL-агента [12].

В отличие от типичного набора параметров во время обучения с подкреплением, который направлен на максимизацию игрового результата, здесь цель обучения – собрать набор обучающих данных, которые напоминают игру человека и содержат достаточно много разнообразных примеров из различных игровых сценариев, чтобы повысить полноту обучающей выборки. Из-за этого функция вознаграждения для обучения RL-агента будет фиксированной для конкретной игры.

С другой стороны, можно использовать самого агента как основу для создания генеративных симуляций. Речь идёт о самом процессе изучения окружающей среды во время обучения. Агент может сам представлять собой модель, обучающуюся моделировать правила окружения, чтобы на основе этих правил выбирать следующее своё действие. Модель мира, которую агенты изучают в процессе, становится основой для генеративной симуляции.

Изучение модели мира в процессе обучения – это один из подходов (model-based) для обучения агентов, сформировавшийся в области обучения с подкреплением [13].

НЕЙРОСЕТЕВЫЕ МИРОВЫЕ МОДЕЛИ

В этом разделе будут рассмотрены ключевые работы, связанные с model-based подходом к обучению RL-агентов, в частности, в игровых средах. Основной фокус при рассмотрении работ сделан как раз на потенциальной возможности использовать изученные мировые модели для симуляции.

World Models

В одноимённой работе World Models авторы предлагают использовать подобные модели для обучения агентов методом обучения с подкреплением [14]. Нейросетевая модель сама представляет собой агента, обучающегося при взаимодействии с моделируемым миром, и состоит из трёх компонентов. В процессе обучения нейросеть запоминает временные и пространственные параметры моделируемого мира, в результате чего один из компонентов обученной модели можно использовать для обучения агентов без непосредственного взаимодействия с моделируемым миром. Авторы ссылаются на схожесть их подхода с механизмом человеческих сновидений.

Архитектура нейросети состоит из трех подсетей. Одна из них – это вариационный автокодировщик [15] (VAE – Variational AutoEncoder), преобразующий наблюдения агента при взаимодействии с моделируемым миром в вектора скрытого пространства признаков (V). Вторая – рекуррентная нейросеть (M), обучающаяся делать вероятностные прогнозы векторов этого пространства, которые ожидает получить от автокодировщика с целью распознавания динамики в статичных изображениях мира. Третья сеть – простая линейная однослойная

нейросеть (C), использующая результаты предсказаний для выбора действий агента (см. рис. 1).

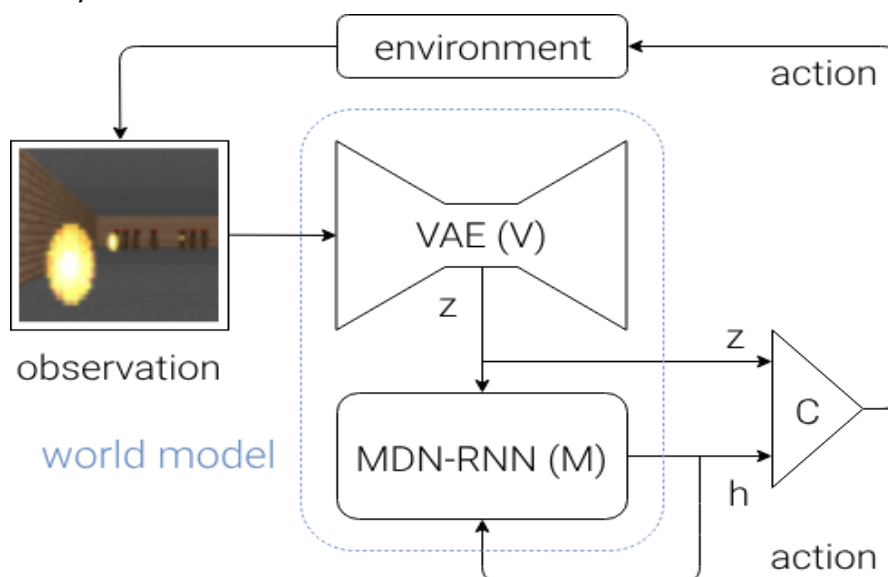


Рис. 1. Архитектура модели World Models [14]

Совокупность первых двух моделей и есть модель мира, сформированная в процессе обучения агента. Спрогнозированные моделью мира вектора скрытого пространства используются как входные данные для нового прогнозирования в той же модели. Таким образом нейросеть «воображает» сценарии из мира, в котором обучалась.

Авторы статьи предложили использовать этот подход для обучения других агентов методом обучения с подкреплением. Учитывая, что нейросетевая модель мира обучалась делать предсказания на своих собственных действиях, она способна заменить симулируемый мир при обучении других агентов. При этом результаты обучения на нейросетевой симуляции можно использовать уже для тестирования агента в обычной симуляции, а автокодировщик позволяет сжимать пространство признаков исходного мира в скрытое пространство меньшей размерности, упрощая процесс обучения, в результате чего этот подход стал применяться в робототехнике.

PlaNet

Исследователи Google AI совместно с DeepMind доработали систему обучения агентов с использованием нейросетевых симуляций. Изначальный подход предполагал использование изображений для прогнозирования дальнейших

действий агента. Использование нейросетевой симуляции подразумевало постоянное кодирование и декодирование этих изображений, которые затем формировали воображаемые нейросетью сценарии взаимодействия [16].

Архитектура *PlaNet* использует то же самое скрытое пространство, в котором в процессе обучения формируется пространственно-временная модель реального мира, полученная благодаря преобразованию наблюдений во время обучения агента. *PlaNet* прогнозирует значения скрытого пространства, а сценарии взаимодействия с миром моделируются нейросетью через эти значения, а не через изображения. Подобный подход позволяет агентам изучать более абстрактные представления, такие как положения и скорости объектов. Это упрощает прогнозирование, исключая необходимость генерации изображений в процессе обучения.

IRIS

Следующий шаг в области моделирования миров для генеративных симуляции – это работа 2022 года, в которой представлена модель *IRIS* – агент глубокого обучения, использующий трансформеры для эффективного моделирования мира [17]. В отличие от традиционных методов глубокого обучения, которые требуют значительных объемов данных для обучения, *IRIS* достигает высокой эффективности при ограниченном количестве взаимодействий с окружающей средой.

Для обучения и тестирования модели авторы выбрали бенчмарк *Atari 100k* [18], содержащий 26 различных игр. Архитектура модели состоит из тех же автокодировщика и трансформера [19] для обработки последовательности данных, то есть наблюдений и действий. Трансформер используется как авторегрессионная модель, делая новые предсказания на основе своих предыдущих предсказаний. Трансформеры позволяют модели работать с большими объёмами информации и учитывать контекст из разных частей последовательности. Механизм внимания позволяет модели фокусироваться на важных аспектах окружающей среды и игнорировать несущественные данные.

IRIS достаточно около двух часов игры для обучения. Модель смогла достичь результата 1.046 по метрике Mean Human-Normalized Score. Это не самый высокий результат среди других агентов, например, модель *EfficientZero* [20] пре-

восходит *IRIS* практически в два раза, однако здесь интересно именно само применение трансформеров для моделирования мира.

ГЕНЕРАТИВНЫЕ СИМУЛЯЦИИ

Успешное развитие разнообразных подходов к моделированию миров для обучения в них агентов породило новую ветку исследований в области генеративных симуляций. Одни из основных вопросов исследователей в этой области – можно ли использовать генеративные нейросетевые модели для создания уникальных и полностью интерактивных миров? Могут ли подобные модели симулировать взаимодействие с этими мирами? Решение этих задач может значительно изменить игровую индустрию в целом, оставив в прошлом привычную разработку игр в специализированном ПО – игровых движках. Далее в статье приведены примеры уже существующих нейросетевых моделей, способных, с некоторыми ограничениями и допущениями, но всё же генерировать и симулировать интерактивные миры.

DIAMOND

На основе результатов авторов *IRIS*, в 2024 году вышла работа, демонстрирующая результаты генерации уже другой модели – *DIAMOND* (Diffusion As a Model Of eNvironment Dreams) [21]. В ней авторы генерируют интерактивную симуляцию на базе диффузионных моделей для того же *Atari* бенчмарка.

Целью этой работы было показать, как сильно визуальные детали в мировых моделях влияют на последующее обучение агентов внутри них. По сравнению с *IRIS*, *DIAMOND* достигла оценки Mean Human-Normalized Score в 1.46, что уже значительно превышает результаты *IRIS*.

Чтобы модель могла работать в реальном времени, исследователи внесли изменения в архитектуру диффузионной модели. Распространённую DDPM [22] они заменили на модифицированную EDM [23], что позволило снизить количество итераций удаления шума до 3. Это также позволило избежать проблемы аккумуляции ошибки, которая присутствовала в DDPM-подходе, при малом количестве итераций удаления шума. Но наиболее интересным достижением авторов является эксперимент с симуляцией игрового процесса игры *CS:GO* (см. рис. 2).



Рис. 2. Возможности генеративной симуляции модели *DIAMOND* [24]

Цель эксперимента – изучить возможности генеративной симуляции модели уже в трёхмерном мире. В этот раз модель обучалась на специальном датасете, содержащем более 87 часов геймплея на одной из игровых локаций [24]. По результатам эксперимента модель оказалась способной достоверно воспроизводить окружение и реагировать на действия игрока на протяжении 100 шагов, но она испытывает трудности с последовательной генерацией, например, при приближении к стене, когда модель теряет контекст из-за ограничения памяти.

Genie

Чуть ранее в том же году Google DeepMind представила *Genie* – генеративное интерактивное окружение. Это модель, способная генерировать полноценные интерактивные игры жанра платформер только из пользовательского текстового описания или изображения [25].

Ключевое достижение исследователей из Google DeepMind – обучение модели только на визуальном наборе данных, без использования меток действий. Это позволило модели самостоятельно выявлять закономерности, что является

важным шагом к созданию универсальных агентов, способных адаптироваться к различным задачам.

Чтобы достичь подобного результата, была использована модель скрытых действий, кодирующая изменения между кадрами в определённое действие. Для этого использовался автокодировщик на основе сетей векторного квантования, что позволило сократить количество кодируемых действий до небольшого дискретного набора кодов (см. рис. 3). В основе модели всё так же лежит трансформер, но с альтернативной пространственно-временной архитектурой (ST-Transformer) [26]. Исследователи также упоминали об использовании другой модификации трансформера для эффективной обработки данных из изображений (Vision transformer) [27].

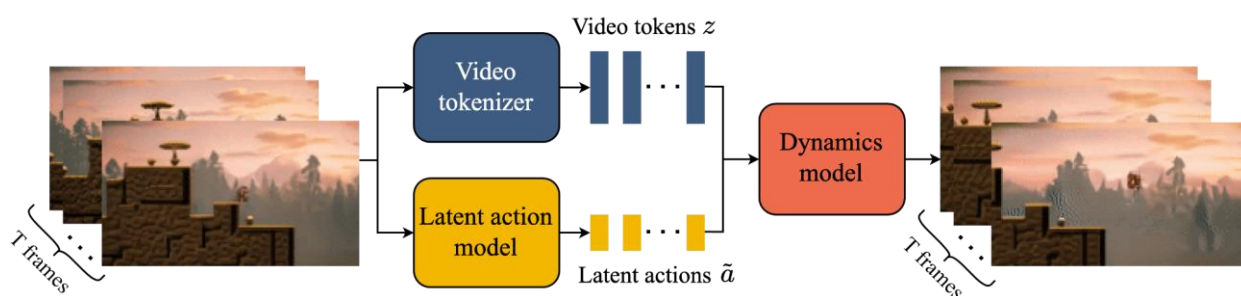


Рис. 3. Архитектура и процесс обучения модели *Genie* [25]

Модель обучалась на более чем 30 тысячах часов (около 6.8 млн. небольших клипов) видеозаписей геймплея из различных 2D-платформеров. Конечное количество параметров модели довольно внушительное – около 10 млрд., хотя и обучение производилось на кластере из 256 TPU⁶.

Помимо возможности воспроизводить игровой процесс игр платформеров, исследователи сообщают о некоторых эмерджентных способностях модели. Например, она научилась воспроизводить одну из характерных особенностей игр этого жанра – параллакс игрового фона, что говорит о способности модели воспринимать и трёхмерные пространства.

⁶ TPU (сокр. от англ. Tensor Processing Unit) – тензорный процессор, это особый тип процессоров, используемый для нейронных вычислений.

GameNGen

Параллельно с предыдущими работами исследователи из Google Research представили свою модель – *GameNGen* [28]. Это генеративная модель для симуляции игрового процесса игры *DOOM*, также созданная на базе диффузионной нейросетевой модели для генерации изображений *Stable Diffusion 1.4* [29]. Для обучения модели разработчики использовали выборку игр, предварительно записанных во время обучения отдельного нейросетевого агента в самой игре.

Диффузионные модели обучаются на тексте и текстовых встраиваниях [30]. Авторы *GameNGen* смогли модифицировать модель для обучения на последовательности предыдущих кадров, используя кодировщик. Он кодирует изображения в скрытом пространстве, сжимая их и уменьшая размерность исходных данных. Очевидно, одних лишь изображений недостаточно для интерактивной симуляции. В этом случае вместо текстовых встраиваний изучались встраивания действий. В игре – это нажатия определённых клавиш. Авторы также адаптировали механизм перекрёстного внимания для работы с закодированной последовательностью действий.

Симуляция игрового процесса происходит в реальном времени, с небольшой, но достаточной для стабильной игры частотой кадров – около 20 кадров в секунду. Предположительно, благодаря изначально низкому разрешению симулируемой игры и использованию предыдущих кадров для предсказания, авторы смогли сократить количество итераций удаления шума до 4, что позволило уменьшить количество вычислительных операций и получить практически такое же качественное изображение, как при 20 итерациях.

Хоть эта модель не является точной симуляцией игры, она способна симулировать сложные игровые механики, такие как подсчёт здоровья и боеприпасов, атаки врагов, повреждения объектов, открытие дверей и сохранение состояния игры на длительных траекториях (см. рис. 4). Интересно, что игровой интерфейс, в котором отображается основная информация о здоровье, боеприпасах и доступном вооружении, сохраняет свою согласованность с игровым процессом на протяжении всей симуляции. Учитывая этот факт, можно предположить, что подобная визуализация части информации о состоянии игрового мира в игровом интерфейсе помогает модели не только выявлять закономерности изменения

среды и интерфейса, но и использовать интерфейс как способ долгосрочного хранения информации.



Рис. 4. Фрагмент игрового процесса, сгенерированного *GameNGen* [30]

Для верификации, помимо отдельных метрик оценки качества генерации, таких как пиковое отношение сигнала к шуму (PSNR) или изученное воспринимаемое сходство участков изображения (LPIPS) [31], авторы также представили статистику человеческих оценок, собранную с помощью отдельного инструмента. Они дали 10-ти кандидатам возможность изучить около сотни небольших клипов длительностью от 1.5 до 3 секунд, сгенерированных моделью, каждый из которых был показан параллельно с таким же клипом из самой игры. Кандидаты должны были отличить, какой из клипов был симуляцией, а какой – фрагментом настоящего игрового процесса. По итогам тестирования оценщики смогли отличить симуляцию только в 58% случаях, но с увеличением длительности клипов результаты ухудшились до 60%.

Oasis

Модель *Oasis* представляет собой интерактивную нейросимуляцию с открытым миром, разработанную компанией Decart в сотрудничестве с Etched [32]. *Oasis* генерирует интерактивный мир в реальном времени. Симуляция основана на игре *Minecraft*, которая отличается высокой степенью агентивности игрока. Благодаря этому, каждое действие пользователя непосредственно влияет на происходящее в мире.

Симуляция способна воспроизводить почти все игровые механики исходной игры, от простого перемещения до взаимодействия с игровыми персонажами и изменения игрового окружения игроком. Модель также научилась симулировать взаимодействие с игровыми интерфейсами. Компания выложила демонстрационную версию модели в открытый доступ.

Oasis основана на диффузионных моделях, которые зарекомендовали себя как эффективные инструменты для генерации изображений и видео. В отличие от традиционных диффузионных моделей, *Oasis* использует трансформеры, что позволяет учитывать контекст предыдущих кадров. Архитектура включает дополнительные временные слои внимания, интегрированные между пространственными слоями внимания, что обеспечивает контекстуализацию на основе предыдущих кадров. Диффузия выполняется в скрытом пространстве, созданном с помощью вариационного автокодировщика Vision Transformer (ViT VAE) [33], что позволяет сосредоточиться на более высокоуровневых характеристиках изображения (см. рис. 5).

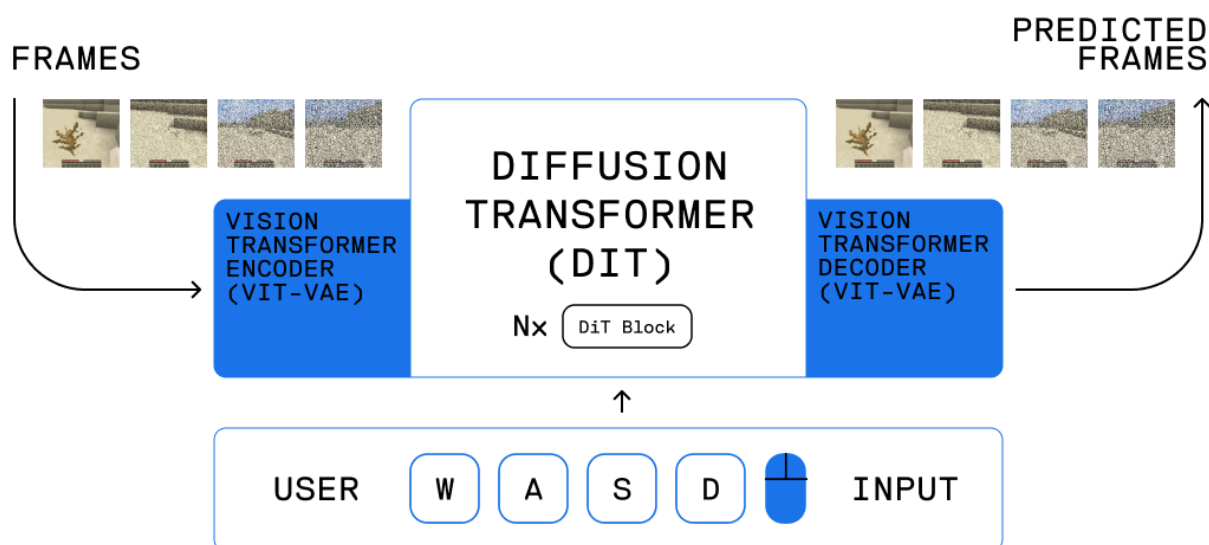


Рис. 5. Архитектура *Oasis* [33]

Подобная трансформерная архитектура выбрана не случайно. Компания хочет протестировать модель на своих интегральных схемах особого назначения (ASIC) Sohu⁷, предназначенных для работы с трансформерами. Авторы надеются

⁷ <https://etched.com/announcing-etched>

увеличить возможности генерации до 4K разрешения. На данный момент модель способна генерировать 20 кадров симуляции в секунду в разрешении 360 p и только на специальном тензорном процессоре NVIDIA H100⁸.

ОГРАНИЧЕНИЯ ГЕНЕРАТИВНЫХ СИМУЛЯЦИЙ

Несмотря на впечатляющие результаты, модели генеративных симуляций – это сравнительно новая область исследований. Эксперименты по выявлению эффективных архитектур моделей, подходов к обучению все еще продолжаются, а современные прототипы для симуляции игровых процессов очень далеки от популярных игровых движков, которые они и стремятся заменить. Как минимум, для обучения модели нужно симулировать игровой процесс, этот самый игровой процесс необходимо смоделировать или использовать уже готовую игру, как это было в большей части рассмотренных ранее работ.

Качество симуляции всех моделей страдает от одних и тех же проблем: аккумуляция ошибки и ограничение памяти. Большинство работ признает наличие этих проблем, а где-то исследователи уже демонстрируют свои решения, что мы и рассмотрим в этом разделе.

Аккумуляция ошибки

Из-за своей авторегрессионной природы модели для генеративных симуляций страдают от проблемы аккумуляции ошибки. Например, авторы *GameNGen* отмечают, что статичное изображение начинает деградировать на 20–30 шаге симуляции, если игрок не перемещается (рис. 6). Авторы *DIAMOND* отмечали, что этот эффект происходит, начиная с 50-го шага генерации уже динамичного изображения в двухмерных играх. *Oasis* демонстрирует нестабильную работу при статичной симуляции мира с некоторыми структурами и не всегда справляется с игровыми интерфейсами.

⁸ <https://nvidia.com/en-us/data-center/h100/>

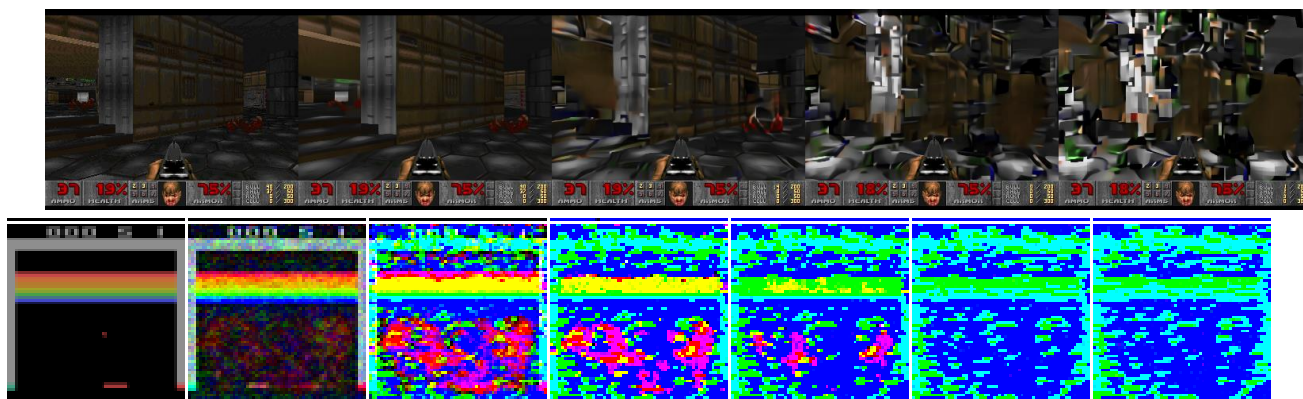


Рис. 6. Визуальные артефакты, появляющиеся в результате аккумуляции ошибки в моделях *GameNGen* (сверху) и *DIAMOND* (снизу)

Авторы каждой из упомянутых моделей по-своему решили проблему аккумуляции. *GameNGen* использует механизм увеличения шума. Определённое количество гауссовского шума добавляется к закодированным кадрам во время обучения, а уровень шума предоставляется модели в качестве входных данных [34]. Разработчики *GameNGen* сообщают, что подобный подход позволил им добиться стабильной продолжительной генерации без потерь качества изображения.

В *DIAMOND* с этой целью использована альтернативная диффузионная модель генерации изображений (EDM). Помимо стабилизации изображения на длительной дистанции, она также позволяет сократить число итераций удаления шума.

Oasis решили названную проблему, внедрив систему динамического шумоподавления, которая регулирует шум во время прогнозирования, добавляя шум при первых прямых проходах диффузии, чтобы уменьшить накопление ошибок, и постепенно удаляя шум при последующих проходах, чтобы модель могла находить и сохранять высокочастотные детали в предыдущих кадрах для повышения последовательной генерации.

Ограничение памяти

Вторая гораздо более крупная проблема подобных моделей – ограниченная память. К настоящему моменту результаты генераций всех подобных моделей визуально воспринимаются как сон. Детали на статичных изображениях по-

стоянно изменяются, динамичное изменение окружения сохраняет свою консистентность лишь в пределах поля зрения игрока. Ни одна из моделей не способна на последовательную генерацию, и ограничение памяти проявляется в симуляциях по-разному.

Модели, симулирующие трехмерное пространство (*GameNGen*, *DIAMOND*, *Oasis*), не способны долгосрочно запоминать сгенерированное окружение. В *Oasis*, как только часть окружения скрывается из поля зрения игрока, модель забывает его и генерирует новое, если игрок вернёт камеру в исходное положение. Чуть меньше эта проблема проявляется в *DIAMOND* и *GameNGen*. Там это происходит, когда игрок подходит вплотную к стене или смотрит в пол или в небо. Дело в том, что *GameNGen* и *DIAMOND* обучались на играх с фиксированным игровым окружением и обучились достоверно прогнозировать его по предыдущим кадрам, в то время как *Oasis* пытается симулировать открытый процедурно сгенерированный мир (см. рис. 7).

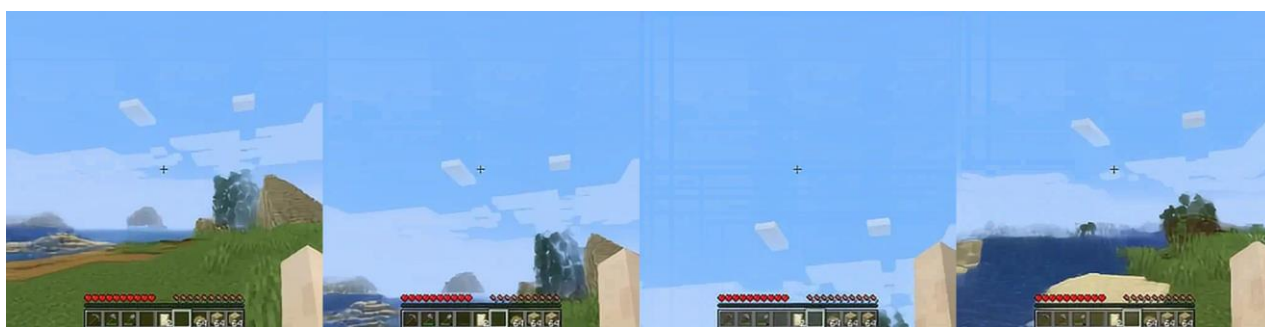


Рис. 7. Демонстрация ограничения памяти на примере *Oasis*: модель «забывает» сгенерированный мир, когда он покидает поле зрения камеры

В попытке решить проблему без кардинального изменения архитектуры Google DeepMind представила вторую версию модели *Genie*, обученную уже на трёхмерных играх различных жанров. Эта модель уже способна на последовательную симуляцию целой минуты игрового процесса [35]. По большей части это связано с увеличением размера самой модели, огромному объёму обучающей выборки и высокими вычислительными мощностями, доступным компании. Несмотря на это, в некоторых примерах игрового процесса, сгенерированного *Genie 2*, способность модели запоминать окружение не превосходит уровень предыдущих моделей.

ЗАКЛЮЧЕНИЕ

На основе рассмотренных работ можно отметить, что существуют следующие направления для дальнейших исследований:

- **Память.** Решение проблемы с ограничениями памяти модели может быть неплохой отправной точкой для стабильной генерации окружения. Вероятно, это потребует пересмотра самой архитектуры моделей и нейросетей, что уже создает большое поле для экспериментов.
- **Обобщение генеративной модели на другие игры.** Следующий шаг к полноценной генерации игрового окружения – это обучение обобщенных моделей, способных стимулировать игровые процессы не только конкретной видеоигры, но и целого отдельного класса видеоигр.
- **Тестирование.** Обучение с подкреплением можно использовать для автоматизации процесса тестирования видеоигр.

В ходе изучения возможностей существующих моделей генеративных симуляций мы выделили основную проблему существующих моделей – ограничение памяти. Для решения этой проблемы мы провели детальное сравнение описанных архитектур, отобрав наиболее важные компоненты каждой. Эти компоненты составят основу модели, на которой мы будем проводить дальнейшие эксперименты, с целью выявить архитектуру, позволяющую качественно улучшить память моделей генеративных симуляций.

Изучение возможностей генеративных нейросетевых симуляций продемонстрировало огромный потенциал для создания интерактивных и динамических виртуальных миров, что может радикально изменить не только игровую индустрию, но и подходы к моделированию реальных процессов. Модели на основе нейросетевых миров и генеративных сетей позволяют успешно имитировать сложные взаимодействия с окружающей средой.

Однако, несмотря на значительный прогресс, текущие технологии сталкиваются с рядом проблем, таких как аккумуляция ошибки и ограничения памяти, что ограничивает возможности их применения в долгосрочных и сложных симуляциях. Решение этих проблем в долгосрочной перспективе может сделать модели нейросетевых генеративных симуляций новым стандартом игровой разработки и даже вытеснить игровые движки.

ЛУДОГРАФИЯ⁹

Blizzard Entertainment. (2016) [2010.] *Starcraft II*. [видеоигра][загрузка][Windows, macOS]. Blizzard Entertainment.

Id Software. (1993) [1993.] *DOOM*. [видеоигра][загрузка, CD][DOS, Windows, macOS, Linux]. Id Software.

Mojang Studios. (2025) [2011.] *Minecraft* [видеоигра][загрузка][Windows, macOS, Linux, Android, iOS, iPadOS, Xbox 360, Raspberry Pi, Windows Phone, PlayStation 3, Fire OS, PlayStation 4, Xbox One, PlayStation Vita, Wii U, Apple TV, tvOS, Samsung Gear VR, Nintendo Switch, New Nintendo 3DS, PlayStation VR]. Mojang Studios, Xbox Game Studios, Sony Interactive Entertainment.

Valve Corporation. (2025) [2013.] *Dota 2* [видеоигра][загрузка][Windows, macOS, Linux] Valve Corporation.

Valve Corporation, Hidden Path Entertainment. (2023) [2012.] *Counter Strike: Global Offensive* [видеоигра][загрузка][Windows, macOS, Linux] Valve Corporation.

СПИСОК ЛИТЕРАТУРЫ

1. Fayaz S.A. et al. Machine learning: An introduction to reinforcement learning // Machine Learning and Data Science: Fundamentals and Applications. 2022. P. 1–22.
2. Mnih V. et al. Human-level control through deep reinforcement learning // Nature. 2015. Vol. 518. No. 7540. P. 529–533.
3. Silver D. et al. Mastering the game of go without human knowledge // Nature. 2017. Vol. 550. No. 7676. P. 354–359.
4. Silver D. et al. Mastering the game of Go with deep neural networks and tree search // Nature. 2016. Vol. 529. No. 7587. P. 484–489.
5. Vinyals O. et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning // Nature. 2019. Vol. 575. No. 7782. P. 350–354.
6. Berner C. et al. Dota 2 with large scale deep reinforcement learning // arXiv

⁹ Лудография (от лат. ludus – «игра» и греч. graphō (γράφω) – «пишу») – список игр, упоминаемых в тексте, который отделён от библиографии. Здесь описание каждой игры представлено в формате, предложенном в рекомендациях <http://semionerd.blogspot.com/p/how-to-write-ludography.html>.

preprint arXiv:1912.06680. 2019.

7. Сахибгареева Г.Ф., Кузурасова В.В., Большаков Э.С. Инструменты балансирования игр // Электронные библиотеки. 2023. Т. 26. No. 2. С. 225–251.
8. Rani G. et al. A deep reinforcement learning technique for bug detection in video games // International Journal of Information Technology. 2023. Т. 15. No. 1. С. 355–367.
9. Wiering M.A., Van Otterlo M. Reinforcement learning // Adaptation, learning, and optimization. 2012. Vol. 12. No. 3. P. 729.
10. Майн Х., Осаки С. Марковские процессы принятия решений. 1977.
11. Spaan M.T.J. Partially observable Markov decision processes. // Reinforcement learning: State-of-the-art. Berlin, Heidelberg: Springer Berlin Heidelberg. 2012. P. 387–414.
12. Cai Q. et al. A survey on deep reinforcement learning for data processing and analytics // IEEE Transactions on Knowledge and Data Engineering. 2022. Vol. 35. No. 5. P. 4446–4465.
13. Moerland T.M. et al. Model-based reinforcement learning: A survey // Foundations and Trends in Machine Learning. 2023. Vol. 16. No. 1. P. 1–118.
14. Ha D., Schmidhuber J. World models // arXiv preprint arXiv:1803.10122. 2018.
15. Pinheiro Cinelli L. et al. Variational autoencoder // Variational methods for machine learning with applications to deep networks. Cham: Springer International Publishing. 2021. P. 111–149.
16. Hafner D. et al. Learning latent dynamics for planning from pixels // International conference on machine learning. PMLR. 2019. P. 2555–2565.
17. Micheli V., Alonso E., Fleuret F. Transformers are sample-efficient world models // arXiv preprint arXiv:2209.00588. 2022.
18. Kaiser L. et al. Model-based reinforcement learning for Atari // arXiv preprint arXiv:1903.00374. 2019.
19. Vaswani A. et al. Attention is all you need // Advances in neural information processing systems. 2017. Vol. 30.
20. Ye W. et al. Mastering Atari games with limited data // Advances in neural information processing systems. 2021. Vol. 34. P. 25476–25488.

21. *Alonso E. et al.* Diffusion for world modeling: Visual details matter in Atari // Advances in Neural Information Processing Systems. 2024. Vol. 37. P. 58757–58791.
22. *Ho J., Jain A., Abbeel P.* Denoising diffusion probabilistic models // Advances in neural information processing systems. 2020. Vol. 33. P. 6840–6851.
23. *Karras T. et al.* Elucidating the design space of diffusion-based generative models // Advances in neural information processing systems. 2022. Vol. 35. P. 26565–26577.
24. *Pearce T., Zhu J.* Counter-strike deathmatch with large-scale behavioural cloning // 2022 IEEE Conference on Games (CoG). IEEE, 2022. P. 104–111.
25. *Bruce J. et al.* Genie: Generative interactive environments // Forty-first International Conference on Machine Learning. 2024.
26. *Xu M. et al.* Spatial-temporal transformer networks for traffic flow forecasting // arXiv preprint arXiv:2001.02908. 2020.
27. *Dosovitskiy A. et al.* An image is worth 16x16 words: Transformers for image recognition at scale // arXiv preprint arXiv:2010.11929. 2020.
28. *Valevski D. et al.* Diffusion models are real-time game engines // The Thirteenth International Conference on Learning Representations. 2024.
29. *Rombach R. et al.* High-resolution image synthesis with latent diffusion models // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022. P. 10684–10695.
30. *Yu H. et al.* Uncovering the text embedding in text-to-image diffusion models // arXiv preprint arXiv:2404.01154. 2024.
31. *Zhang R. et al.* The unreasonable effectiveness of deep features as a perceptual metric // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. P. 586–595.
32. *Decart E. et al.* Oasis: A universe in a transformer.
URL: <https://oasis-model.github.io/>
33. *Choi B., Jeong J.* ViV-Ano: Anomaly detection and localization combining vision transformer and variational autoencoder in the manufacturing process // Electronics. 2022. Vol. 11. No. 15. P. 2306.
34. *Pasini M. et al.* Continuous autoregressive models with noise augmentation avoid error accumulation // arXiv preprint arXiv:2411.18447. 2024.

35. Parker-Holder J. et al. Genie 2: A Large-Scale Foundation World Model. URL: <https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/>

REAL-TIME GENERATIVE SIMULATION OF GAME ENVIRONMENT

E. S. Bolshakov¹ [0000-0002-2208-9515], V. V. Kugurakova² [0000-0002-1552-4910]

^{1, 2}*Institute of Information Technologies and Intelligent Systems, Kazan (Volga Region) Federal University, ul. Kremlyovskaya, 35, Kazan, 420008*

¹edward.bolshakov117@gmail.com, ²vlada.kugurakova@gmail.com

Abstract

This paper explores the potential of generative neural network simulations, focusing on the application of reinforcement learning methods and neural world models for creating interactive worlds. Key achievements in agent training using reinforcement learning are discussed. Special attention is given to neural world models, as well as generative models such as Oasis, DIAMOND, Genie, and GameNGen, which employ diffusion networks to generate realistic and interactive game worlds. The opportunities and limitations of generative simulation models are examined, including issues related to error accumulation and memory constraints, and their impact on the quality of generation. The conclusion presents suggestions for future research directions.

Keywords: *video games, game environment, generative simulation, reinforcement learning, generative neural networks, gameplay simulation, world models.*

LUDOGRAPHY

Blizzard Entertainment. (2016) [2010.] *Starcraft II*. [видеоигра][загрузка][Windows, macOS]. Blizzard Entertainment.

Id Software. (1993) [1993.] *DOOM*. [видеоигра][загрузка, CD][DOS, Windows, macOS, Linux]. Id Software.

Mojang Studios. (2025) [2011.] *Minecraft* [видеоигра][загрузка][Windows, macOS, Linux, Android, iOS, iPadOS, Xbox 360, Raspberry Pi, Windows Phone, PlayStation 3, Fire OS, PlayStation 4, Xbox One, PlayStation Vita, Wii U, Apple TV, tvOS, Samsung Gear VR, Nintendo Switch, New Nintendo 3DS, PlayStation VR]. Mojang Studios, Xbox Game Studios, Sony Interactive Entertainment.

Valve Corporation. (2025) [2013.] *Dota 2* [видеоигра][загрузка][Windows, macOS, Linux] Valve Corporation.

Valve Corporation, Hidden Path Entertainment. (2023) [2012.] *Counter Strike: Global Offensive* [видеоигра][загрузка][Windows, macOS, Linux] Valve Corporation.

REFERENCES

1. Fayaz S.A. et al. Machine learning: An introduction to reinforcement learning // Machine Learning and Data Science: Fundamentals and Applications. 2022. P. 1–22.
2. Mnih V. et al. Human-level control through deep reinforcement learning // Nature. 2015. Vol. 518. No. 7540. P. 529–533.
3. Silver D. et al. Mastering the game of go without human knowledge // Nature. 2017. Vol. 550. No. 7676. P. 354–359.
4. Silver D. et al. Mastering the game of Go with deep neural networks and tree search // Nature. 2016. Vol. 529. No. 7587. P. 484–489.
5. Vinyals O. et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning // Nature. 2019. Vol. 575. No. 7782. P. 350–354.
6. Berner C. et al. Dota 2 with large scale deep reinforcement learning // arXiv preprint arXiv:1912.06680. 2019.
7. Sahibgareeva G.F., Kugurakova V.V., Bolshakov E.S. Game balancing tools [Instrumenti balansirovaniya igr] // Russian Digital Library Journal. 2023. Vol. 26. No. 2. P. 225–251 (in Russian).
8. Rani G. et al. A deep reinforcement learning technique for bug detection in video games // International Journal of Information Technology. 2023. Vol. 15. No. 1. P. 355–367.
9. Wiering M.A., Van Otterlo M. Reinforcement learning // Adaptation, learning, and optimization. 2012. Vol. 12. No. 3. P. 729.
10. Main H., Osaki S. Markovskie processi prinatiya reshenii. 1977.
11. Spaan M.T.J. Partially observable Markov decision processes // Reinforcement learning: State-of-the-art. Springer Berlin Heidelberg. 2012. P. 387–414.
12. Cai Q. et al. A survey on deep reinforcement learning for data processing and analytics // IEEE Transactions on Knowledge and Data Engineering. 2022. Vol. 35. No. 5. P. 4446–4465.

13. *Moerland T.M. et al.* Model-based reinforcement learning: A survey // Foundations and Trends in Machine Learning. 2023. Vol. 16. No. 1. P. 1–118.
14. *Ha D., Schmidhuber J.* World models // arXiv preprint arXiv:1803.10122. 2018.
15. *Pinheiro Cinelli L. et al.* Variational autoencoder // Variational methods for machine learning with applications to deep networks. Cham: Springer International Publishing. 2021. P. 111–149.
16. *Hafner D. et al.* Learning latent dynamics for planning from pixels // International conference on machine learning. PMLR. 2019. P. 2555–2565.
17. *Micheli V., Alonso E., Fleuret F.* Transformers are sample-efficient world models // arXiv preprint arXiv:2209.00588. 2022.
18. *Kaiser L. et al.* Model-based reinforcement learning for Atari // arXiv preprint arXiv:1903.00374. 2019.
19. *Vaswani A. et al.* Attention is all you need // Advances in neural information processing systems. 2017. Vol. 30.
20. *Ye W. et al.* Mastering Atari games with limited data // Advances in neural information processing systems. 2021. Vol. 34. P. 25476–25488.
21. *Alonso E. et al.* Diffusion for world modeling: Visual details matter in Atari // Advances in Neural Information Processing Systems. 2024. Vol. 37. P. 58757–58791.
22. *Ho J., Jain A., Abbeel P.* Denoising diffusion probabilistic models // Advances in neural information processing systems. 2020. Vol. 33. P. 6840–6851.
23. *Karras T. et al.* Elucidating the design space of diffusion-based generative models // Advances in neural information processing systems. 2022. Vol. 35. P. 26565–26577.
24. *Pearce T., Zhu J.* Counter-strike deathmatch with large-scale behavioural cloning // 2022 IEEE Conference on Games (CoG). IEEE, 2022. P. 104–111.
25. *Bruce J. et al.* Genie: Generative interactive environments // Forty-first International Conference on Machine Learning. 2024.
26. *Xu M. et al.* Spatial-temporal transformer networks for traffic flow forecasting // arXiv preprint arXiv:2001.02908. 2020.
27. *Dosovitskiy A. et al.* An image is worth 16x16 words: Transformers for image recognition at scale // arXiv preprint arXiv:2010.11929. 2020.

28. *Valevski D. et al.* Diffusion models are real-time game engines // The Thirteenth International Conference on Learning Representations. 2024.
29. *Rombach R. et al.* High-resolution image synthesis with latent diffusion models // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022. P. 10684–10695.
30. *Yu H. et al.* Uncovering the text embedding in text-to-image diffusion models // arXiv preprint arXiv:2404.01154. 2024.
31. *Zhang R. et al.* The unreasonable effectiveness of deep features as a perceptual metric // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. P. 586–595.
32. *Decart E. et al.* Oasis: A universe in a transformer.
URL: <https://oasis-model.github.io/>
33. *Choi B., Jeong J.* ViV-Ano: Anomaly detection and localization combining vision transformer and variational autoencoder in the manufacturing process // Electronics. 2022. Vol. 11. No. 15. P. 2306.
34. *Pasini M. et al.* Continuous autoregressive models with noise augmentation avoid error accumulation // arXiv preprint arXiv:2411.18447. 2024.
35. *Parker-Holder J. et al.* Genie 2: A Large-Scale Foundation World Model.
URL: <https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/>

СВЕДЕНИЯ ОБ АВТОРАХ



БОЛЬШАКОВ Эдуард Сергеевич – ассистент кафедры индустрии разработки игр института ИТИС. Сфера научных интересов – игровая сценаристика, игровой дизайн, применение нейросетей в разработке видеоигр.

Eduard Sergeevich BOLSHAKOV – assistant of the Game Development Industry Department at the ITIS Institute. Research interests – game storytelling, game design, application of neural networks in video game development.

email: edward.bolshakov117@gmail.com

ORCID: 0000-0002-2208-9515



КУГУРАКОВА Влада Владимировна — доцент, к. т. н., зав. кафедрой индустрии разработки видеоигр ИТИС КФУ. Сфера научных интересов — различные аспекты проектирования игр.

Vlada Vladimirovna KUGURAKOVA — Associate Professor, Head of the Video Game Development Industry Chair of ITIS KFU. Research interests include various aspects of game design.

email: vlada.kugurakova@gmail.com

ORCID: 0000-0002-1552-4910

Материал поступил в редакцию 28 января 2025 года