УДК 004.83 DOI 10.18522/2311-3103-2026-1-250-270

В.Б. Савинов, Н.Н. Шушарина

ОБЗОР МЕТОДОВ УЛУЧШЕНИЯ РАССУЖДЕНИЙ В БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЯХ

Появление больших языковых моделей стало важным этапом в области обработки естественного языка, поскольку такие модели демонстрируют впечатляющие результаты в генерации, трансформации и анализе текстов, а также в решении широкого спектра прикладных задач. Однако, несмотря на значительные успехи успехи практического применения, большие языковые модели обладают ограниченными способностями к рассуждению. Эти ограничения проявляются в трудностях обобщения знаний за пределами обучающего распределения, проблемах переноса знаний в новые контексты, а также в снижении точности выполнения многошаговых логических и математических операций. Целью данной работы является изучение методов повышения способности больших языковых моделей к рассуждению, где под рассуждением понимается процесс формирования и оценки выводов из существующей информации. В работе рассматриваются основные типы рассуждений, релевантные для больших языковых моделей: математическое, логическое и рассуждение на основе здравого смысла. Приводится список наиболее часто используемых контрольных тестов, применяемых для оценки качества рассуждений языковых моделей. Проводится обзор методов улучшения рассуждений, применяемых в больших языковых моделях на 2025 год. В зависимости от этапа применения (на этапе обучения или на этапе использования языковой модели) - рассматриваются подходы к подготовке данных для обучения, изменения в архитектуре языковых моделей, процедуры обучения и дообучения моделей (в том числе с использованием специально подготовленных синтетических наборах данных), обучение с подкреплением, различные варианты построения смысловых цепочек, механизмы интеграции внешних инструментов и многоагентный подход. Также рассматриваются существующие ограничения больших языковых моделей, которые проявляются в отсутствии понимания концепций, плохой обобщающей способности вне распределения обучающих данных и снижении эффективности при росте сложности задач. Выделяются наиболее перспективные методы, направленные на повышение качества и надежности рассуждений больших языковых моделей.

Большая языковая модель; методы рассуждений; искусственный интеллект.

V.B. Savinov, N.N. Shusharina

A REVIEW OF METHODS FOR IMPROVING REASONING IN LARGE LANGUAGE MODELS

The emergence of large language models has become an important milestone in the field of natural language processing, as such models demonstrate impressive results in text generation, transformation, and analysis, as well as in solving a wide range of applied tasks. However, despite significant practical success, large language models possess limited reasoning capabilities. These limitations manifest in difficulties with generalizing knowledge beyond the training distribution, challenges in transferring knowledge to new contexts, and reduced accuracy when performing multi-step logical and mathematical operations. The goal of this work is to examine methods for improving the reasoning abilities of large language models, where reasoning is understood as the process of forming and evaluating inferences based on existing information. The paper discusses the main types of reasoning relevant to large language models: mathematical, logical, and commonsense reasoning. It provides a list of the most commonly used benchmarks applied to assess the reasoning quality of language models. An overview is presented of the methods used to enhance reasoning in large language models at 2025. Depending on the stage of application (during training or during model usage), the work examines approaches to training data preparation, architectural modifications of language models, training and finetuning procedures (including those using specially constructed synthetic datasets), reinforcement learning, various chain-of-thought construction techniques, mechanisms for integrating external tools, and multi-agent approaches. The paper also discusses existing limitations of large language models, which include the lack of conceptual understanding, poor out-of-distribution generalization, and reduced effectiveness as task complexity increases. Finally, the most promising methods aimed at improving the quality and reliability of reasoning in large language models are highlighted.

Large language model; reasoning methods; artificial intelligence.

Введение. Появление больших языковых моделей (БЯМ) вызвало настоящий прорыв в области обработки естественного языка (англ. Natural Language Processing, NLP). Лежащая в основе БЯМ архитектура трансформер демонстрирует впечатляющую производительность в прогнозировании следующего токена [1] и кроме задач, связанных с обработкой текста, успешно применяется в задачах компьютерного зрения, обработки мультимодальных данных, системах управления роботами и т.д. Однако БЯМ обладают ограниченными способностями к рассуждению, что обусловлено их конструкцией и процедурой обучения, которая делает акцент на статистическом моделировании языка, а не на понимании [2]. Это поднимает вопрос о возможностях БЯМ к проявлению способности к рассуждению, что является одним из важнейших аспектов систем искусственного интеллекта (ИИ).

Целью данной работы является изучение методов, применяемых для улучшения рассуждений БЯМ. В первой части описывается, что понимается под рассуждением в контексте БЯМ. Во второй части перечисляются контрольные тесты, используемые для оценки рассуждений. Третья часть рассматривает методы используемые для улучшения рассуждений БЯМ, исходя из возможных этапов применения при разработке БЯМ: подготовка данных, выбор архитектуры, обучение, дообучение и использование. Приводятся существующие у БЯМ ограничения к рассуждению. В заключении делаются выводы о наиболее перспективных методах для улучшения качества рассуждений БЯМ.

1. Рассуждение в больших языковых моделях. Большая языковая модель (от англ. large language model, LLM) – это предварительно обученная статистическая языковая модель (рассматривает текст как последовательность слов и выполняет оценку вероятности слов), основанная на искусственных нейронных сетях (ИНС) с большим числом параметров [3]. В основном, БЯМ построены на архитектуре трансформер [1] и содержат от единиц до сотен миллиардов параметров [3].

Критики БЯМ отмечают их способность совершать глупые ошибки (фактические и логические), непоследовательность и ограниченность рассуждений, отсутствие здравого смысла и планирования ответа, невозможность исправления их авторегрессионной сути. БЯМ демонстрируют неспособность обобщения информации [4] и существенное снижение точности ответов в случае нестандартного представления задач [5].

Однако области применения БЯМ регулярно расширяются и кроме задач обработки естественного языка они уже применяются для решения самых разных задач: решение математических проблем, управление роботами и автономными автомобилями, реализация различных автономных агентов [6], социологические исследования и т.д.

Рассуждение – это процесс формирования и оценки выводов из существующей информации [7]. Традиционно, рассуждения разделяются на дедуктивное, индуктивное и абдуктивное, но применительно к базовым моделям ИИ, рассуждение разделяют по задачам применения: рассуждения на основе здравого смысла, математические рассуждения, логические рассуждения, математические рассуждения, причинно-следственные рассуждения, визуальные рассуждения, аудио рассуждения, мультимодальные рассуждения, воплощенные рассуждения и т.д. В случае БЯМ, рассуждение разделяют на математическое, логическое, рассуждение на основе здравого смысла и рассуждение специфичное для предметной области [8].

Несмотря на все успехи БЯМ в генерации и обработки текста, требуется различать формальную и функциональную языковые компетенции [9]. Где под формальной языковой компетенцией понимается знание правил и статистических закономерностей языка, а под функциональной языковой компетенцией – способность к языковому мышлению. Учитывая это различие, можно отметить, что хотя БЯМ владеют формальной языковой компетенцией на уровне человека, с функциональной языковой компетенцией отмечаются проблемы [9]. При этом, если с увеличением количества обучающих данных формальная языковая компетенция у БЯМ также улучшается, то с функциональной языковой компетенцией эффект масштаба выражен менее явно, что вынуждает разработчиков достигать интересующих видов поведения БЯМ при помощи специализированных методов (обучение с подкреплением или использование внешних инструментов).

Таким образом, несмотря на существующие проблемы и критику – появляются новые области и способы применения БЯМ, а представления о возможностях БЯМ постоянно меняются. Применительно к БЯМ, основными видами рассуждения являются математическое, логическое и рассуждение на основе здравого смысла. Стандартные БЯМ обладают ограниченной способностью к рассуждению и хотя они способны решать задачи, похожие на образцы, присутствовавшие в обучающей выборке, но в случае новых проблем, требующих рассуждений - у БЯМ возникают трудности.

2. Тесты для оценки производительности. Для контроля прогресса развития БЯМ используются специальные контрольные тесты (бенчмарки) [10].

Для оценки рассуждений на основе здравого смысла используются:

- HellaSwag – выбор из нескольких вариантов завершения предложения.
- BoolQ – вопрос по короткому тексту из Википедии с вариантами ответа Да/Нет.
- PIQA – вопросы на физический здравый смысл.
- SIQA (SocialIQA – Social Interaction QA) – описание социальных ситуаций и возможных вариантов ответов.
- WinoGrande – решение неоднозначных задач заполнения пропусков в предложениях, которые предлагают двоичные варианты (стал ответом на достижение высоких результатов на Winograd Schema Challenge).
- ARC (AI2 Reasoning Challenge) – вопросы по естественным наукам начальной школы (разделен на Сложный и Простой наборы вопросов).

Для оценки математических рассуждений БЯМ применяются:

- MATH – решение математических задач.
- GSM8K – тестирование математических рассуждений. Набор состоит из порядка 8000 вопросов и ответов по математике для начальной школы (7473 обучающих и 1319 тестовых задач).
- MGSM (англ. Multilingual Grade School Math) – это 250 задач из GSM8K, переведенные на 10 языков (Испанский, Французский, Немецкий, Русский, Китайский, Японский, Тайский, Суахили, Бенгальский, Телугу).
- SVAMP (англ. Simple Variations on Arithmetic Math word Problems) – набор элементарных математических текстовых задач (состоят из короткого текстового описания и вопросов о неизвестных значениях).
- AIME (англ. American Invitational Mathematics Examination) – набор математических олимпиадных задач для американских школьников (например, AIME24 иAIME25 – 30 задач).

Для оценки логических рассуждений используются:

- ProntoQA – синтетический набор данных для анализа цепочки мыслей БЯМ.
- LogiQA – набор из 8678 вопросов для проверки понимания прочитанного и оценки способности к логическому мышлению (данные: 7376 для тренировки, 651 для оценки и 651 для тестирования) на Английском и Китайском языках).

Другие контрольные тесты, используемые для проверки БЯМ:

- GPQA (Graduate-Level Google-Proof Q&A Benchmark) – сложный набор из 448 вопросов (от экспертов по биологии, физики и химии) с несколькими вариантами ответов.
- BIG-bench (Beyond the Imitation Game benchmark) – набор для оценки способностей БЯМ. Состоит из 204 разнообразных заданий (включают: лингвистику, детское развитие, математику, здравый смысл, биологию, физикц, социальные предубеждения, разработку программного обеспечения).
- BIG-Bench Hard (BBH) – набор из 23 сложных задач BIG-bench, требующих многоэтапных рассуждений (включают в себя логическую дедукцию, арифметику, рассуждения на основе здравого смысла).
- DROP (Discrete Reasoning Over the content of Paragraphs) – набор для оценки возможностей по пониманию и рассуждению. Набор содержит 96 тысяч вопросов, которые были получены с помощью краудсорсинга (открывки из Википедии и сложные вопросы к ним).

♦ AlpacaEval – метод для оценки способности следовать инструкциям пользователя. Набор данных содержит 805 пользовательских инструкций с эталонными ответами от GPT-4. Для уменьшения предвзятости, связанной с предпочтением более длинных ответов используется AlpacaEval 2.

♦ Arena-Hard – метод для оценки ответов на запросы пользователя. Набор данных постоянно обновляется на основе данных с краудсорсинговой платформы Chatbot Arena, содержит 500 пользовательских запросов и использует GPT-4-Turbo в качестве судьи для сравнения с ответами базовой модели (GPT-4).

♦ ARC-AGI (англ. Abstraction and Reasoning Corpus for Artificial General Intelligence) – тест на интеллект общего назначения, нацеленный на способности обобщения и адаптации к новым задачам (публичные данные: 400 тренировочных и 400 тестовых задач, приватные данные: 200 задач для валидации).

Для оценки способностей БЯМ к рассуждению предлагаются и более простые тесты: например, подсчет букв в слове (человек может выполнить эту задачу с точностью в 100%, а от БЯМ потребуется базовая способность к рассуждениям).

Однако при использовании контрольных тестов необходимо учитывать следующие проблемы: насыщение (на некоторых бенчмарках современные БЯМ достигают почти идеальной производительности), утечки тестовых данных (тестовые данные бенчмарков могли попасть в тренировочную выборку БЯМ, что может приводить к переобучению модели), качество самих тестов (бенчмарки различаются по качеству и могут иметь сложности с воспроизводимостью). Для преодоления этих сложностей появляются новые более сложные бенчмарки, разрабатываются динамические или закрытые версии контрольных тестов.

Таким образом, для тестирования БЯМ существует большое количество самых разнообразных бенчмарков. Для оценки способности БЯМ к рассуждению наиболее часто используются ARC AI2, GSM8K (или MGSM), MATH, BBH.

3. Методы улучшения рассуждений. Для улучшения способности БЯМ к рассуждениям возможно использовать различные подходы. Рассмотрим их исходя из возможных этапов применения: на этапах обучения (подготовка данных, архитектура, дообучение) или на этапе использования БЯМ (рис. 1).

Рис. 1. Методы улучшения рассуждений БЯМ.

Данные. Качественные и разнообразные данные играют важнейшую роль при обучении языковых моделей. При создании открытого набора данных FineWeb-Edu было показано, что фильтрация интернет-страниц из наборов данных CommonCrawl и вычленение из них только обучающих материалов – позволило повысить показатели в тестах оценки знаний и рассуждений [11]. Тщательная фильтрация и вычленение страниц с ма-

тематическими данными из набора CommonCrawl и данные программного кода позволили обучить модель DeepSeekMath 7B, которая продемонстрировала 51.7% на математическом бенчмарке MATH [12]. Включение в обучающую выборку данных с программным кодом повышает не только способность БЯМ генерировать код, но также улучшает способность к рассуждению и знания о мире [13].

Распространенным подходом улучшения рассуждений является обучение на синтетических данных.

Используя качественные образцы и принципы символьной логики, были получены образцы логических рассуждений в виде набора данных Formal Logic Deduction Diverse (FLD) (состоит из примеров многоступенчатых дедукций с неизвестными фактами, разными правилами рассуждений), что позволило улучшить способность БЯМ к логическим рассуждениям [14].

В процессе обучения модели phi-4 (14 миллиардов параметров) в основном использовались синтетические данные, что позволило показать производительность на уровне моделей большего размера, особенно в контрольных тестах на рассуждение [15]. Для создания наборов синтетических данных в качестве основы можно использовать предварительно отобранные данные из сети Интернет, которые затем можно использовать для создания задач на рассуждение [15].

При обучении модели MiniMax-M1, отказались от использования синтетических данных, а отдали приоритет парам вопрос-ответ и до 70% увеличили долю данных, соответствующих науке, технологии, инженерии и математике (англ. Science, Technology, Engineering, Mathematics, STEM) [16].

Таким образом, использование при обучении БЯМ качественных данных позволяет значительно улучшить производительность моделей, без необходимости масштабирования их размеров. Важным методом является использование синтетических данных, но требуется внимательно контролировать их качество и разнообразие.

Архитектура. Рассмотрим какие изменения можно внести в архитектуру трансформера, чтобы добиться улучшения качества рассуждений.

В основе архитектуры трансформера лежит механизм внимания, использующий многопеременную логистическую функцию (softmax), которая применяется для расчета итоговых весов внимания, переводя итоговый вектор значений в вероятностное распределение. При работе обученной модели с данными, чье распределение отличается от того, которое использовалось при обучении, softmax размывается. Чтобы исправить ситуацию и добиться более резкого распределения, предлагается использовать адаптивную температуру [17]. Также механизм внимания может быть модифицирован для формирования паттернов ассоциации и манипуляции, что важно для выполнения рассуждений, включающих связывание и преобразование понятий [18].

Изменения в самой архитектуре БЯМ, включающие добавление обучаемого механизма связей между слоями, могут способствовать улучшению математических рассуждений [19].

Учитывая, что исследования нейронаук показывают, что человеческий мозг (как носитель обобщенного интеллекта) имеет разные области, отвечающие за обработку языка и решение логических и математических задач, то и для построения будущих интеллектуальных систем потребуется не единственная модель, а модульная архитектура, сочетающая языковые модели с моделями, которые представляют абстрактные знания и поддерживают сложные рассуждения [9].

Исследование показывает, что рассуждения можно проводить не выходя в пространство токенов [20]. У БЯМ может быть реализована работа в двух режимах: латентном – когда внутреннее представление БЯМ используется для последующей генерации, и языковом – стандартный режим работы БЯМ с генерацией токенов. Процесс обучения подобной модели начинается с обычной генерации текста, но на последующих этапах – первые шаги рассуждения заменяются внутренним представлением (выделяются специальными токенами <bot> и <eot>). Подобный подход позволяет модели выполнять процесс рассуждения без языковых ограничений, что может интерпретироваться в виде поиска по дереву [20].

Высказывается предположение, что для осуществления когнитивных процедур, связанных с обработкой информации (в частности – рассуждений) мозг опирается на пространственные представления (характеризующие не только мир вещей, но и субъективный внутренний мир мыслей), приобретаемые при взаимодействии с окружающим миром [21]. Для объединения БЯМ с моделями, осуществляющими обработку сенсорных данных или робототехнических данных - может использоваться подход смешения мнений экспертов (англ. Mixture of experts, MoE).

К архитектурным изменениям также относится изменения токенизатора (инструмент, преобразующий текст в данные (токены)). Подход «Байтовый латентный трансформер» (англ. Byte Latent Transformer, BLT) демонстрирует отказ от токенизатора с фиксированным словарем и переход на динамическую токенизацию уровне байтов (входной поток разбивается на наборы байтов, чьи границы определяются по энтропии следующего байта) [22]. Данный подход не только демонстрирует эффективность обучения и вывода за счет динамического выбора длины наборов байт (если данные предсказуемы), но и улучшение в рассуждениях и обобщении [22].

Таким образом, для улучшения способностей и качества рассуждений можно рассматривать различные варианты изменения архитектуры БЯМ: изменение токенизатора, механизма внимания, добавление возможности обработки сенсорной информации, использование архитектуры MoE.

Обучение. Для улучшения качества рассуждений, БЯМ обучаются в несколько этапов, которые характеризуются увеличением относительного количества высококачественных данных с примерами рассуждений (математические задачи, код) [23].

Так предварительное обучение моделей Qwen3 и MiMo-7B [23] проходило в три этапа:

1. Начальный этап обучения, на котором модель получает предварительные знания. Используется небольшое контекстное окно (для моделей Qwen3 и MiMo-7B: 4096 и 8192 токенов, соответственно).

2. Повышение способности к рассуждению, за счет увеличения доли качественных (научных, математических, программного кода, примеров рассуждения) данных. Длина контекстного окна не меняется.

3. Расширение возможностей решения сложных задач за счет дополнительного включения высококачественных данных, синтетических ответов на задачи по математике и программированию. Расширение длины контекста (для моделей Qwen3 и MiMo-7B: до 32768 токенов).

Для предварительного обучения модели MiniMax-M1 использовались 7.5T токенов данных: на первых 2.5T токенах данных обучение велось с постоянной скоростью обучения 8e-5, а затем постепенно уменьшалась до 8e-6 [16]. Расширение длины контекста происходило постепенно в 4 этапа (позволяет избегать взрыва градиентов), начиная от 32 килобайт и увеличивая до 1 миллиона токенов.

Таким образом, для улучшения у БЯМ способности к рассуждению, процедура предварительного обучения проводится в несколько этапов, с постепенным увеличением количества качественных данных, относящихся к математическим задачам, примерам программного кода, научным и техническим материалам.

Дообучение. Для улучшения рассуждений БЯМ можно использовать дообучение модели на специально подготовленных данных.

Метод самообучения рассуждению (англ. Self-Taught Reasoner, STaR) использует БЯМ, чтобы генерировать рассуждения для ответа на вопросы, используя несколько первоначальных примеров рассуждений в качестве образца [24]. В случае если сгенерированный ответ неверен - рассуждение генерируется снова (учитывая правильный ответ), а полученные рассуждения (давшие правильный ответ) используются в качестве данных для дообучения модели, и далее цикл повторяется. Данный подход позволил на основе небольшого набора первоначальных подсказок (с несколькими примерами) создать большой набор данных рассуждений, что позволило дообученной модели улучшить результаты на контрольных тестах CommonsenseQA и GSM8K [24]. Однако одной из клю-

чевых проблем подобного подхода является возможность использования неправильных обоснований, которые в конце приводят к правильным ответам. Подобные рассуждения будут попадать в обучающую выборку, что приведет к снижению качеств модели. Для преодоления данной проблемы – необходимо проверять не только ответ, но и все промежуточные шаги рассуждения.

Для обучения БЯМ способности генерировать внутренние размышления перед формированием ответа была предложена методика оптимизации мыслительных предпочтений (англ. Thought Preference Optimization, TPO), которая заключается в итеративной генерации БЯМ различных вариантов размышлений и ответов, оценке их качества с помощью отдельной модели-судьи и последующей оптимизации БЯМ с помощью прямой оптимизации предпочтений (англ. Direct Preference Optimization, DPO) на основе этих предпочтений [25]. Модель (Llama-3-8B-Instruct), обученная с помощью TPO, продемонстрировала улучшение результатов не только в задачах, требующих рассуждения, но и в общих знаниях, что позволяет предположить универсальность подхода. Однако, хотя тесты на AlpacaEval 2 и Arena-Hard показали улучшение – результаты на GSM8K стали хуже, что может быть связано с отсутствием достаточного числа математических инструкций во время обучения [25].

Для решения задач ARC-AGI, используя комбинированный подход индукции (выведение промежуточных функций (программный код на Python)) и трансдукции (непосредственное предсказание результатов, используя только имеющиеся примеры), БЯМ обучили на большом корпусе (сотни тысяч) синтетических данных программного кода, генерирующего варианты решения задач, что позволило приблизиться к уровню производительности человека [26].

Метод самокоррекции с помощью обучения с подкреплением (англ. Self-Correction via Reinforcement Learning, SCoRe) улучшает способность БЯМ самостоятельно исправлять свои ошибки, используя многоэтапный процесс обучения с подреплением, используя данные, сгенерированные самой моделью [27]. Метод SCoRe повысил точность базовой модели на бенчмарках MATH (+15.6%) и HumanEval (+9.1%) [27].

Метод обратного мышления (англ. Reverse Thinking, RevThink), вдохновляется человеческой способностью рассуждать как в прямом, так и в обратном направлениях и позволяет улучшить способность БЯМ к рассуждению путем дообучения модели (Mistral-7B-Instruct-v0.3, с помощью LoRA (англ. Low-Rank Adaptation)) на данных, представляющих собой сгенерированные решения сразу трех задач: прямое рассуждение по вопросу, формулировка обратного вопроса и обратные рассуждения в ответ на обратный вопрос [28]. Метод RevThink не только показал лучшие результаты на бенчмарках: GSM8K(60.9%), MATH (15.3%), ARC (78.5%), но и продемонстрировал способность эффективно обобщать знания на новые, ранее не встречавшиеся наборы данных [28].

Дообучение на образцах рассуждения методом контролируемой тонкой настройки (англ. Supervised fine-tuning, SFT) даже на небольшом количестве математических примеров позволяет научить БЯМ генерировать смысловые цепочки [29, 30].

В [29] из начального набора в 59 тысяч математических вопросов, был отобран небольшой набор данных s1K, содержащий всего1000 примеров. Каждый пример состоит из исходного вопроса, примера рассуждения и решения (сгенерированных Google Gemini Flash Thinking API). Процесс отбора этой тысячи примеров основывался на принципах качества, сложности и разнообразия:

- ♦ игнорировались некачественные и плохо отформатированные примеры,
- ♦ выбирались вопросы, которые не могли решить базовые БЯМ (Qwen2.5-7B-Instruct и Qwen2.5-32B-Instruct) и требовали для решения длительного рассуждения,
- ♦ для классификации вопросов по предметным областям использовалась отдельная БЯМ (Claude 3.5 Sonnet).

На полученным наборе данных s1K была дообучена (SFT) модель Qwen2.5-32B-Instruct, в результате чего была получена модель s1-32B, которая продемонстрировала значительное улучшение производительности (по сравнению с базовой моделью) на контрольных тестах AIME24, MATH500 и GPQA Diamond. Также был представлен простой

метод, принудительного ограничения бюджета ответа, позволяющий контролировать объем вычислений (токенов размышления), используемых во время генерации [29]. Данный метод либо принудительно завершает процесс рассуждений модели путем принудительного добавления фразы завершения рассуждений («Final Answer:»), либо стимулирования более продолжительных рассуждений путем удаления токена завершения рассуждений и добавления фразы «Wait», если модель пытается преждевременно остановиться. Подобный подход позволяет способствовать коррекции рассуждений и ответов модели, что было продемонстрировано увеличением точности на AIME24 по мере выделения большего количества токенов на размышление [29].

В [30] из миллионов математических задач было отобрано для последующего дообучения всего 817 примеров. При этом, основная гипотеза исследования состояла в том, что БЯМ уже имеют предварительные знания, скрытые в пространстве параметров модели и достаточно небольшого количества примеров, демонстрирующих образцы решения проблем, чтобы активировать способность БЯМ к сложным рассуждениям. Для отбора данных использовались критерии уровня сложности, общности и разнообразия:

- приоритет сложным задачам, способствующим развитию сложных цепочек рассуждений (сначала исключались задачи, которые за несколько попыток могла решить Qwen2.5-Math-7B-Instruct и оставляли только те задачи, которые рассуждающие модели DeepSeek-R1 и DeepSeek-R1-Distill-Qwen32B могли решить с вероятностью ниже заданного порога);
- выбор задач, которые стимулируют новые подходы к рассуждениям;
- должны охватывать широкий спектр математических задач.

Кроме отбора самих задач, осуществлялся отбор решений, качество которых оценивалось по структурной организации, эффективной реализации и надежности ответа:

- решение должно иметь четкую структуру и форматирование, избегая излишней многословности в объяснении простых шагов;
- поступательное решение с четким изложением ключевых идей;
- процесс рассуждения должен содержать этапы проверки (промежуточных результатов, предположений, логической последовательности выводов.

В результате дообучения (SFT) модели Qwen2.5-32B-Instruct на полученном наборе данных была получена модель LIMO, которая продемонстрировала улучшение производительности не только на математических контрольных тестах (AIME24, MATH500 и AMC23 (American Mathematics Competition)), но и на задачах, относящихся к областям вне пределов распределения тренировочных данных (англ. Out-of-Distribution, OOD).

Таким образом, дообучение БЯМ на специально подготовленных данных позволяет обучить БЯМ генерировать смысловые цепочки и улучшить качество рассуждений.

Обучение с подкреплением. Для выполнения сложных рассуждений, модель o1 от компания OpenAI была обучена с помощью обучения с подкреплением (англ. Reinforcement Learning, RL) [31]. Так как технические подробности о процедуре обучения модели o1 отсутствуют – возможно только рассмотреть возможные подходы по воспроизведению возможностей модели OpenAI o1 применительно к обучению с подкреплением [32]. Ключевыми компонентами являются: инициализация политики (предварительное обучение модели для развития навыков рассуждения), дизайн вознаграждения (создание эффективных сигналов для формирования вознаграждений), поиск (система генерации качественных решений на этапах обучения и тестирования) и само обучение (используя данные, полученные в процессе поиска, для улучшения политики) [32].

Появление открытой модели DeepSeek-R1 [33] продемонстрировало как именно обучение с подкреплением позволяет создавать большие рассуждающие модели (БРМ) (англ. large reasoning model, LRM). В основе DeepSeek-R1 лежит MoE-модель DeepSeek-V3 (671 миллиардов параметров), на базе которой была обучена модель DeepSeek-R1-Zero. При этом, использовалось только обучение с подкреплением, где в качестве алгоритма применялся подход оптимизации групповой относительной политики [12] (англ. Group Relative Policy Optimization, GRPO). Главной особенностью данного алгоритма является отказ от использования отдельной оценочной модели и выполнение оценки по группе сгенерированных ответов:

1. Формирование группы ответов на запрос.

2. Определение оценки для каждого ответа в группе.

3. Вычисление преимущества ответов относительно среднего значения оценок в группе.

Ответы с положительным преимуществом усиливают генерацию.

Оценка ответов модели осуществлялась при помощи проверок правильности ответа, корректности форматирования и выполнения проверочного кода.

Примечательно, что во время обучения у модели случился то, что авторы назвали «момент озарения» («Aha Moment»), когда в цепочке рассуждений модель написала: “Wait, wait. Wait. That’s an aha moment I can flag here.” и пересмотрела первоначальный подход к решению задачи [33].

Однако цепочки рассуждений DeepSeek-R1-Zero обладали ограниченной читабельностью, поэтому итоговая модель DeepSeek-R1 была обучена в несколько этапов:

1. Начальное дообучение базовой модели DeepSeek-V3-Base на специально подготовленном наборе данных, который получил название «Холодный старт» (англ. Cold Start). Данные были подготовлены в формате удобном для чтения человеком в виде: «блок рассуждения+итоговый ответ». Данный этап позволил дать более качественную начальную точку для последующего обучения.

2. Этап обучения с подкреплением, аналогичный DeepSeek-R1-Zero. В ответах модели наблюдалось смешение языков, поэтому было дополнительно добавлено вознаграждение за языковую согласованность (приводит к небольшому снижению производительности, но делает ответ более читаемым).

3. Дообучение (SFT) модели DeepSeek-V3-Base в течении 2 на специально подготовленном синтетическом наборе данных: 600 тысяч на рассуждение и 200 тысяч без рассуждений (ответы на вопросы, перевод).

4. Финальное обучение с подкреплением, аналогичное методологии обучения DeepSeek-R1-Zero, но в этом случае уже не только на математических и программных задачах, но и на общих запросах, где для оценки используются модели-оценщики, оценивающие полезность и безвредность ответа.

Итоговая модель DeepSeek-R1 не только продемонстрировала производительность на уровне передовой модели OpenAI o1, но и генерирует открытые результаты рассуждения.

Обучение с подкреплением превосходит контролируемую тонкую настройку в переносимости на другие задачи: модели, обученные с помощью RL на математических задачах также демонстрируют улучшение в задачах из других областей [34]. Наряду с RLHF, обучение с подкреплением и проверяемыми вознаграждениями (англ. Reinforcement Learning with Verifiable Rewards, RLVR) стали использоваться для дообучения моделей решению сложных задач. И если в случае RLHF используются данные о предпочтениях людей, то в RLVR – правильные результаты математических задач и выполнения инструкций (например, выполнения тестов для кода), что ограничивает возможности применения данного подхода в областях, где подобная проверка невозможна. Для преодоления данной проблемы, существует возможность использования собственной уверенности модели в качестве вознаграждения.

Таким образом, использование обучения с подкреплением позволяет значительно улучшить качество рассуждений БЯМ, однако его использование нуждается в дополнительном изучении.

Использование. БЯМ обучаются предсказывать токены (лексемы) и, фактически, они не предназначены для явных рассуждений, но используя определенные пользовательские инструкции можно решить эту проблему, направляя модель на выполнение необходимых шагов рассуждения [3]. К таким инструкциям относятся: цепочка и дерево мыслей, самосогласованность, рефлексия.

Цепочка мыслей (англ. Chain of Thought, CoT) является одним из самых распространенных методов улучшить способность БЯМ к рассуждению, используя только пользовательскую инструкцию [35]. Подход смысловой цепочки основывается на том, чтобы неявный процесс рассуждения стал явным: в инструкции описываются шаги рассуждения, необходимые чтобы БЯМ смогла прийти к логическому и обоснованному результату [3].

Существуют две основных формы подсказок цепочек мыслей (CoT):

1. С примерами (англ. Few-shot). В инструкции предоставляются несколько примеров пошаговых рассуждений, которые используются в качестве шаблонов для БЯМ [35].

2. Без примеров (англ. Zero-shot). В инструкцию добавляется фраза «Давайте размышлять шаг за шагом» («Let's think step by step»), что побуждает БЯМ самостоятельно формулировать этапы рассуждения [36].

Форма инструкции с примерами показывает большую эффективность, чем инструкция без примеров, но ее эффективность зависит от выбора подходящих и разнообразных примеров рассуждений [3]. Ручное конструирование подобных примеров является сложной и подверженной ошибкам задачей, что может быть решено автоматической CoT .

Дерево мыслей (англ. Tree of Thought, ToT) – это метод, основанный на идее рассмотрения альтернативных решений, представляя их в виде «дерева рассуждений», в котором каждая ветвь является отдельной линией рассуждений [37]. Данный метод позволяет исследовать различные возможные гипотезы, оценивая их обоснованность и релевантность исходной задаче [3]. ToT являет расширением CoT и может быть полезен для решения сложных задач, где одной линии рассуждений может быть недостаточно (неоднозначность , учет нюансов).

Граф мыслей (англ. Graph of Thoughts, GoT) – метод моделирования информации, генерируемой БЯМ в виде графа, где вершины – это информационные единицы («мысли»), а ребра — зависимости между ними [38]. GoT является развитием таких подходов как CoT и ToT: при решении сложной задачи человек не только следует цепочке мыслей (как в CoT) или перебирает разные мысли (как в ToT), а формирует сложную сеть мыслей по которым может следовать, возвращаться и объединять между собой [38].

Самосогласованность (англ. Self-Consistency) – метод, где БЯМ генерирует несколько вариантов ответа (используя CoT) и согласованность этих ответов между собой, служит признаком правильности результата [39]. Данный метод основан на предположении, что в задачах, требующих рассуждения, возможно несколько различных путей решения, приводящих к одному правильному ответу. Самосогласованность не требует дополнительных данных, обучения, вспомогательных моделей и работает используя только одну БЯМ, но при этом повышает точность языковых моделей на контрольных тестах, требующих рассуждения (GSM8K, SVAMP, ARC, CommonsenseQA) [39]. Метод можно использовать для проверки фактов с целью обеспечения точности информации [3].

Самостоятельное уточнение (англ. Self-Refine) – метод улучшения первоначального ответа БЯМ посредством итеративной обратной связи и самокоррекции [40]. Особенность (и основное ограничение) метода состоит в том, что одна модель выполняет все функции: сначала генерирует первоначальный ответ, затем его оценивает (предоставляет обратную связь) и потом использует этот отзыв для улучшение своего ответа.

Подобный метод может быть реализован в виде самопроверки (англ. SelfCheck), когда БЯМ изучает и оценивает логические шаги (сгенерированные на предыдущих этапах), сравнивает результаты и исправляет ошибки [41].

Рефлексия – метод в котором БЯМ используется для оценки собственного ответа [42]. Метод является итеративным: после генерации первоначального ответа, модели предлагается оценить свой ответ, что позволяет модели выявить потенциальные ошибки (предложить улучшения) и уточнить свои результаты, повышая качество своих ответов [3].

Генерация ответа, дополненная результатами поиска (англ. Retrieval-augmented generation, RAG [43]) – метод сочетания внутренних знаний БЯМ с внешней информацией. Метод RAG-Star использует генерацию ответа, дополненную результатами поиска для улучшения процесса рассуждения [44]. В основе метода лежит использование поиска по дереву Монте-Карло (англ. Monte Carlo Tree Search, MCTS), используемого для итеративного планирования промежуточных подзапросов и ответов, необходимых для решения задачи (используются внутренние знания БЯМ), а дальнейший поиск по дереву выполняется с использованием RAG, чтобы оценивать рассуждения используя внешние данные (статьи из Википедии) [44].

Адаптация инструкций. Для решения сложных задач, обычной цепочки мыслей (CoT) может оказаться недостаточно. Метод SELF-DISCOVER предлагает подход по объединению различных техник рассуждений, когда БЯМ самостоятельно выбирает варианты, подходящие для конкретной задачи [45].

Сам метод состоит из трех шагов [45]:

1. Выбор. БЯМ выбирает несколько (наиболее подходящих для решения задачи) вариантов рассуждений из 39 шаблонов рассуждений (разбиение на подзадачи, креативное, критическое и системное мышление, рассуждение шаг за шагом и другие).

2. Адаптация. БЯМ предлагается адаптировать (перефразировать) выбранные варианты рассуждений для решения данной задачи.

3. Реализация. БЯМ использует адаптированную схему рассуждений для генерации пошагового плана рассуждений для получения финального ответа.

Подход SELF-DISCOVER показал улучшение результатов на контрольных тестах BIG-Bench Hard, Thinking for Doing и MATH, по сравнению с обычным CoT [45].

Обучение во время тестирования (англ. test-time training, TTT) – это расширение классического подхода (предварительное обучение модели и ее последующее использование без изменения параметров) на адаптацию модели для решения конкретной задачи [46].

Метод TTT отличается от обычного дообучения модели, малым количеством данных доступных для обучения, поэтому ключевыми компонентами успешного TTT являются: предварительное дообучение модели на схожих задачах, использование вспомогательных форматов задач и их аугментаций, обучение на каждом экземпляре отдельно (для обучения используется LoRA (для ускорения могут использоваться квантованные адаптеры – QLoRA)) и использование самосогласованности [46]. Использование TTT продемонстрировало значительное повышение точности на задачах ARC-AGI (улучшение в 6 раз по сравнению с базовыми моделями) [46].

Вычисления во время тестирования (англ. test-time compute) – это подход, суть которого состоит в увеличении объемов вычислений, используемых для вывода [47]. Исходя из предпосылки, что для решения сложных задач люди думают дольше, а БЯМ, используя дополнительные вычисления улучшают свои результаты в задачах, требующих рассуждения [35, 42] – было показано, что оптимальное распределение вычислительных ресурсов на этапе вывода может быть более эффективным, чем простое увеличение размера модели (особенно в задачах, требующих сложного рассуждения) [47].

Наиболее яркой демонстрацией успешности данного подхода стало появление осенью 2024 года модели GPT-o1 от компания OpenAI [31]. Модель GPT-o1 была обучена (с помощью обучения с подкреплением) на выполнение сложных рассуждений. Основным нововведением, которое отличает GPT-o1 от прошлых версий GPT стало явное включение цепочки рассуждений (CoT) во время выполнения процесса вывода, что позволяет модели создавать внутреннюю цепочку рассуждений и тем самым решать более сложные задачи [48]. Во время работы, модель выполняет несколько итераций генерации ответа в которых используются токены рассуждений (на следующей итерации генерации они отбрасываются, а в контекст добавляется только ответ, полученный на предыдущем шаге рассуждений).

Производительность модели o1 последовательно увеличивалась не только во время обучения (англ. train-time compute), но и при увеличении времени, затраченного на «размышления» модели (вычисления во время тестирования) [31]. В качестве контрольных тестов (вместо MATH и GSM8K) использовались результаты по AIME (олимпиада по математике для американских школьников) и GPQA. На всех бенчмарках (требующих рассуждения) модель o1 значительно превзошла GPT-4o [31].

Впечатляющие результаты, демонстрируемые моделью o1 от OpenAI, вдохновили и другие исследовательские команды на работы в направлении использования CoT и увеличении вычислений во время тестирования. Появились модели DeepSeek-R1-Lite-Preview и QwQ-32B-Preview.

Исследователи из Alibaba создали модель Marco-o1, в которой используется дообучение на цепочку мыслей, поиск по дереву Монте-Карло (MCTS) и механизм рефлексии [49]. В качестве базовой модели Marco-o1 используется Qwen2-7B-Instruct. Для дообуче-

ния модели (по всем параметрам) использовались наборы данных с примерами цепочек размышлений (Open-O1 CoT dataset (открытый), Marco-o1 CoT dataset (синтетический)). Для проверки модели использовались английский и китайский варианты из набора данных MGSM.

Исследователи HuggingFace также отметили тенденцию, когда вместо масштабирования вычислений во время обучения моделей — используется динамические стратегии вывода, которая позволяет моделям «дольше думать» [50]. Опираясь на исследования, проведенные DeepMind [47], которые показывают, что рост вычислений во время тестирования можно увеличивать для итеративного самосовершенствования или использования отдельной модели, оценивающей ответ для последующего поиска в пространстве решений – был рассмотрен вариант генерации большего числа решений и их последующая оценка отдельной моделью-оценщиком [50]. В качестве основной модели использовались Llama-3.2-Instruct (версии с 1 и 3 миллиардами параметров), а для оценки использовалась модель Llama3.1-8B-PRM-Deepseek-Data (модель с 8 миллиардами параметров, которая была специально обучена выдавать обратную связь на каждый этап процесса рассуждений). Первая модель генерирует решения (до 256 вариантов), а вторая – выполняет их оценку. Для оценки использовался набор данных MATH-500, который является подмножеством из набора данных MATH.

По результатам проведенного исследования, делаются выводы о важной роли правильной оценки шагов рассуждений (которые, как показывает контрольный тест ProcessBench, в настоящее время ограничены [51]), важности развития самопроверки [27], возможности улучшить рассуждения и принятие решений за счет включения промежуточных шагов рассуждений в процесс генерации, использование методов поиска для генерации обучающих наборов данных, адаптация данных методов на другие области рассуждений [50].

Развивая успех o1, OpenAI выпустила модель o3 (в конце декабря 2024), которая набрала 75.7% на оценочном наборе ARC-AGI, а высокопроизводительная конфигурация o3 набрала 87,5% (модели были предварительно обучены на тренировочном наборе), что означает значительное повышение уровня обобщения и адаптации.

Таким образом, среди существующих различных методов, позволяющих улучшить качество рассуждений у БЯМ, можно выделить следующие: подготовка качественных данных для обучения модели (включая программный код и синтетические данные), внесение изменений в архитектуру трансформера (например, использование MoE), обучение с подкреплением, дообучение на данных, содержащих различные варианты размышлений, и применение специальных методов использования БЯМ (CoT, ToT, GoT, самосогласованность, рефлексия, RAG, адаптация инструкций, обучение модели во время тестирования при помощи LoRA). Особо необходимо отметить подход вычисления во время тестирования, который демонстрирует наиболее впечатляющие результаты.

Применение внешних инструментов. БЯМ может использовать внешние инструменты, получая к ним доступ через вызовы программного интерфейса (англ. Application Programming Interface, API), что позволяет преодолевать присущие БЯМ ограничения в выполнении арифметических операций или получения актуальной информации из поисковых систем [52]. Также внешние инструменты могут использоваться для улучшения качества рассуждений.

Цепочка кода (англ. Chain of Code, CoC) – метод, предлагающий БЯМ использовать код для структурирования своих решений [53]. Ключевая идея CoC состоит в использовании БЯМ для представления рассуждений в виде подзадач (представленных программным псевдокодом), которые далее обрабатываются программным интерпретатором, а в случае неопределенного поведения интерпретатора – псевдокод обрабатывается БЯМ (выступающей в роли эмулятора) [53]. На бенчмарке BBH, CoC достигает 84%, что на 12% больше, чем у CoT (превосходит результаты людей в 18 из 23 задач) [53].

Программа мыслей (англ. Program of Thoughts, PoT) – метод в котором БЯМ генерирует программу (на языке программирования Python), которая описывает процесс рассуждения и далее выполняется внешним интерпретатором [54]. Сравнение PoT и CoT

показало рост на математических бенчмарках: GSM8K (+8.5% (71.6%)), SVAMP (+8.8% (85.2%)) [54]. Сочетание PoT с методом самосогласования позволяет добиться еще более лучших результатов: GSM8K (80.0%), SVAMP (89.1%), но они сильно зависят от используемой модели (приведенные результаты получены на модели OpenAI Codex) и зависят от примеров, использованных в инструкции БЯМ [54].

Аналогичный подход используется в *методе программно-управляемых языковых моделей* (англ. ProgramAided Language models, PAL), где БЯМ используется для преобразования задач, сформулированных на естественном языке, в программный код, который затем выполняются с помощью интерпретатора Python [55]. На математических бенчмарках PAL показало улучшение результатов, по сравнению с CoT: GSM8K (+6.4% (72.0%)), SVAMP (+4.6% (79.4%)) [55].

Таким образом, интеграция языковых моделей с внешними инструментами позволяет эффективно сочетать способности БЯМ в понимании и интерпретации задач с возможностями специализированных инструментов.

Агенты. Автономные агенты (АА) (система, взаимодействующая с внешней средой, способная воспринимать эту среду и действовать в ней, выполняя заданные задачи) – являются важным направлением исследований [6]. Системы на базе АА являются многообещающим подходом к созданию искусственного интеллекта общего назначения (англ. Artificial General Intelligence, AGI), который сможет выполнять задачи посредством самостоятельного планирования и действий.

Учитывая, что АА необходимо выполнять определенную задачу (роль), воспринимать окружающую среду и взаимодействовать с ней – их структура формализуется до модулей: профиля, памяти, планирования и действий [6]. Модуль профиля (составляется вручную или генерируется) определяет роль агента (программист, исследователь, редактор и т.п.), который записывается в инструкцию поступающую на вход БЯМ. Модуль памяти позволяет АА хранить информацию, полученную из окружающей среды и использовать эти данные для выполнения будущих действий. Модуль планирования позволяет АА планировать будущие действия (используя смысловые цепочки [35] или другие виды рассуждения), учитывая обратную связь от среды [56], человека или других моделей [40]. Модуль действий отвечает за взаимодействие с окружающей средой (преобразование решений агента в конкретные данные) посредством вызова внешних API, взаимодействия с интерпретаторами программного кода [54], базами данных или знаний и другими моделями.

Кроме различных примеров использования АА для социального моделирования, проведения экспериментов и различных видов автоматизации [6] – подход АА применяется и для улучшения рассуждений.

Метод рассуждения и действия (англ. Reasoning and Acting, ReAct) — метод предлагает БЯМ генерировать последовательность чередующихся рассуждений и действий (взаимодействие с внешними источниками для получения дополнительной информации), что позволяет модели контролировать и корректировать свой план действий [56].

Метод инструментально-интегрированных агентов рассуждений (англ. Tool-integrated Reasoning Agents, ToRA) применяет несколько агентов (использующих внешние инструменты (вычислительные библиотеки и символьные системы решения) для решения математических задач [57]. Для обучения ToRA использовался набор данных ToRA-Corpus (16 тысяч инструкций использования инструментов, сгенерированных при помощи GPT-4 на математических наборах данных GSM8K и MATH) [57]. В результате дообучения LLaMA-2 70B на ToRA-Corpus, полученная модель достигла следующих результатов: GSM8K (84.3%), MATH (49.7%), SVAMP (82.7%) [57].

Система мультиагентного обучения БЯМ (англ. Multiagent LLM training, MALT) предлагает структуру из трех специализированных агентов (генератор, проверяющий и улучшатель), которые последовательно взаимодействуют для решения поставленной задачи [58]. Базовая модель (Llama 3.1 8B) дообучается на синтетических данных траекторий рассуждения, используя систему распределения вознаграждений, основанную на совместном результате, что позволяет эффективно обучать каждую модель в системе и достичь следующих результатов: GSM8K (90.3%), MATH (56.5%) [58].

При этом, систематические эксперименты показывают, что одиночные БЯМ с качественными инструкциями способны достигать такого же уровня производительности, как и многоагентные системы, но многоагентные системы обсуждения превосходят одиночные модели в случаях, когда в инструкциях отсутствуют демонстрации [59].

Таким образом, использование агентов позволяет строить сложные системы, комбинирующие БЯМ и внешние инструменты, что позволяет превосходить одиночные БЯМ.

Ограничения. Несмотря на прогресс в развитии БЯМ и появления БРМ, они имеют значительные ограничения, особенно при решении задач, требующих рассуждений.

Важным недостатком БРМ является феномен чрезмерного обдумывания (англ. overthinking), который проявляется в генерации избыточных цепочкам рассуждений, что приводит к увеличению вычислительных затрат и снижению эффективности моделей [60]. Наиболее явно данный эффект проявляется при решении простых задач [61]. Для преодоления эффекта чрезмерного обдумывания используется различные модификации процесса дообучения (обучение на данных с разной длиной, обучение с подкреплением с наградой за длину [62] или адаптивному переключению в режим без рассуждений [63]) и использования (назначение бюджета токенов на основе оценки сложности задачи и принудительное завершение процесса рассуждения [61]).

Сам процесс рассуждений, который генерируется БРМ – не обязательно коррелирует с ответом модели, что демонстрирует недостаточность мониторинга цепочки рассуждений для ее контроля и анализа [64].

Дообучение БЯМ с использованием обучения с подкреплением (RL) и проверяемыми вознаграждениями (RLVR) улучшает способность моделей выбирать эффективные пути рассуждения уже присутствующие в базовых моделях, а не дает новую способность к рассуждению [65]. Проведенное сравнение метрики pass@k (измеряет вероятность того, что хотя бы один из k ответов будет правильным) базовой модели (Qwen2.5 (7B/14B/32B), LLaMA-3.1-8B для математических задач и DeepSeek-R1-DistillQwen-14B для генерации кода) и их дообученных версий (GRPO) показало, что хотя при низких значениях выборки (pass@1) модели обученные RLVR превосходят свои базовые версии, но все меняется по мере увеличения числа выборок ответа. Базовые модели не только догоняют, но и превосходят модели, дообученные при помощи RLVR на контрольных тестах на математическое рассуждение (MATH500, AIME24, Minerva, Olympiad) и генерацию кода (LiveCodeBench, HumanEval+) [65].

Отмечается, что «момент озарения» или «Aha Moment» (проявился при обучении с подкреплением модели DeepSeek-R1-Zero [33]), на самом деле проявляется у базовой модели, что позволяет сделать заключение о том, что данные языковые артефакты отражают элементы обучающей выборки, а не проявление новых когнитивных процессов и способностей благодаря RL [66].

Сравнение контролируемого дообучения (SFT) и обучение с подкреплением показывает, что по сравнению с RL, модели дообученные рассуждению на математических задачах при помощи SFT не обобщают полученную способность на задачи из других областей [34]. Причем метод дообучения является более важным фактором, так как модели дообученные при помощи RL демонстрируют более высокий индекс переносимости, а дообучение моделей с использованием SFT может приводить к переобучению на математический домен, что приводит к снижению производительности в других областях. При этом, для понимания возможных причин различий в обобщающей способности было проведено сравнения главных компонент (англ. Principal Component Analysis, PCA), полученных из весовых значений, извлекаемых по слоям моделей. Сравнение величин сдвига (евклидово расстояние) PCA показало, что SFT сильнее приводит к изменению весов модели (что может приводить к ухудшению производительности), тогда как использование RL нарушает латентную структуру базовой модели в минимальной степени [34].

Исходя из успешного прохождение контрольных тестов, может сложиться впечатление, что БЯМ способны «понимать» концепции, которые скрываются за вопросами, но на самом деле – истинного осмысления у моделей не возникает [67]. Аналогичное заклю-

чение можно дать и рассуждающим моделям. Используя контрольные тесты, отличные от стандартных, было показано, что БРМ (OpenAI o1/o3, DeepSeek-R1, Claude 3.7 Sonnet Thinking, Gemini Thinking) менее эффективны (по сравнению со стандартными БЯМ) в решении простых задач, превосходят их на задачах средней сложности, но не справляются на сложных задачах (демонстрируя, при этом, снижение количества генерируемых токенов рассуждений) [68]. При этом, даже предоставление явных пошаговых алгоритмов решения используемых для проверки головоломок (Ханойская башня, переправа через реку и т.п.) не улучшает производительность моделей.

Несмотря на критику использованной методологии исследования, которая указывает на корреляцию обвала с достижением максимальной длины вывода, использования неразрешимых проблем (тест «переправа через реку» с $N \geq 6$, при вместимости лодки равной 3) и используемой методологии оценки правильности ответа [69], сама суть выводов подтверждается другим исследованием, которое так же демонстрирует «утомляемость» БРМ при решении сложных задач [70]. Проведенный анализ возникавших при решении задач затруднений, выявил: накопление ошибок с увеличением шагов рассуждений (пропуск информации, фактические ошибки), трудности с длинным контекстом (сложность выделения релевантных данных приводит к пропуску важной информации), статистические сокращения (в некоторых случаях модель может упрощать или делать догадки), недостаточное отслеживание состояния при выполнении многошаговых рассуждений (проявляется, например, в падении производительности при подсчете нескольких слов), плохая обобщающая способность вне распределения обучающих данных (OOD), ошибки копирования и проблемы с токенизацией (проявляется в задачах подсчета числа символов) [70]. Полученные результаты указывают, что БРМ в большей степени имитируют обучающие данные, а не выполняют «истинные рассуждения».

Таким образом, несмотря на успехи БРМ на стандартных тестах, их способность к рассуждению серьезно ограничена ростом сложности задач и глубины вывода. При этом, длинные цепочки рассуждений приводят к проблеме «чрезмерного обдумывания», лишним итерациям и некорректным выводам.

Заключение. В настоящее время большие языковые модели находятся на этапе интенсивного развития. Постоянно появляются новые модели, области и способы их использования. Однако базовая архитектура и процедура обучения БЯМ сводится к статистическому моделированию языка, что не наделяет БЯМ способностью рассуждения. При этом, именно способность к рассуждению является одним из важнейших аспектов систем ИИ.

В данной статье проведен обзор современных методов рассуждений, применяемых в больших языковых моделях в 2025 году, с акцентом на улучшение способности БЯМ к рассуждению и построению смысловых цепочек. Рассмотренные подходы, такие как построение цепочек мыслей, вычисления во время тестирования и интеграция внешних инструментов, позволяют моделям более эффективно справляться с многоступенчатыми задачами.

Для дальнейшего повышения интеллектуальных возможностей БЯМ требуется развитие данных подходов, включая внедрение модульной архитектуры, дообучение на синтетических данных, обучение с подкреплением и развитие многоагентского подхода. Данные направления являются актуальными для дальнейших исследований и представляют собой перспективные направления как для теоретического, так и для практического улучшения больших языковых моделей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Vaswani A.* Attention is all you need // Advances in Neural Information Processing Systems. – 2017.
2. *Mirzadeh I., Alizadeh K., Shahrokhi H., Tuzel O., Bengio S., Farajtabar M.* Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models // arXiv preprint arXiv:2410.05229. – 2024.
3. *Minaee S., Mikolov T., Nikzad N., Chenaghlu M., Socher R., Amatriain X., Gao J.* Large language models: A survey // arXiv preprint arXiv:2402.06196. – 2024.

4. *Berglund L., Tong M., Kaufmann M., Balesni M., Stickland A. C., Korbak T., Evans O.* The reversal curse: Llms trained on" a is b" fail to learn" b is a" // arXiv preprint arXiv:2309.12288. – 2023.
5. *Wu Z., Qiu L., Ross A., Akyürek E., Chen B., Wang B., Kim N., Andreas J., Kim Y.* Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks // arXiv preprint arXiv:2307.02477. – 2023.
6. *Wang L., Ma C., Feng X., Zhang Z., Yang H., Zhang J., Chen Z., Tang J., Chen X., Lin Y., others.* A survey on large language model based autonomous agents // Frontiers of Computer Science. – 2024. – Vol. 18, No. 6. – P. 186345.
7. Arkoudas K. GPT-4 can't reason // arXiv preprint arXiv:2308.03762. – 2023.
8. *Chang Y., Wang X., Wang J., Wu Y., Yang L., Zhu K., Chen H., Yi X., Wang C., Wang Y., Ye W., Zhang Y., Chang Y., Yu P. S., Yang Q., Xie X.* A Survey on Evaluation of Large Language Models // ACM Trans. Intell. Syst. Technol. – 2024. – Vol. 15, No. 3. – DOI: 10.1145/3641289.
9. *Mahowald K., Ivanova A.A., Blank I. A., Kanwisher N., Tenenbaum J.B., Fedorenko E.* Dissociating language and thought in large language models // Trends in Cognitive Sciences. – 2024.
10. *Plaat A., Wong A., Verberne S., Broekens J., Stein N. van, Back T.* Reasoning with large language models, a survey // arXiv preprint arXiv:2407.11511. – 2024.
11. *Penedo G., Kydlíček H., Lozhkov A., Mitchell M., Raffel C., Von Werra L., Wolf T., others*. The fineweb datasets: Decanting the web for the finest text data at scale // arXiv preprint arXiv:2406.17557. – 2024.
12. *Shao Z., Wang P., Zhu Q., Xu R., Song J., Bi X., Zhang H., Zhang M., Li Y., Wu Y., others*. Deepseekmath: Pushing the limits of mathematical reasoning in open language models // arXiv preprint arXiv:2402.03300. – 2024.
13. *Aryabumi V., Su Y., Ma R., Morisot A., Zhang I., Locatelli A., Fadaee M., Üstün A., Hooker S.* To Code, or Not To Code? Exploring Impact of Code in Pre-training // arXiv preprint arXiv:2408.10914. – 2024.
14. *Morishita T., Morio G., Yamaguchi A., Sogawa Y.* Enhancing Reasoning Capabilities of LLMs via Principled Synthetic Logic Corpus // arXiv preprint arXiv:2411.12498. – 2024.
15. *Abdin M., Aneja J., Behl H., Bubeck S., Eldan R., Gunasekar S., Harrison M., Hewett R. J., Javaheripi M., Kauffmann P., others.* Phi-4 Technical Report // arXiv preprint arXiv:2412.08905. – 2024.
16. *Chen A., Li A., Gong B., Jiang B., Fei B., Yang B., Shan B., Yu C., Wang C., Zhu C., others*. MiniMax-M1: Scaling Test-Time Compute Efficiently with Lightning Attention // arXiv preprint arXiv:2506.13585. – 2025.
17. *Veličković P., Perivolaropoulos C., Barbero F., Pascanu R.* Softmax is not enough (for sharp out-of-distribution) // arXiv preprint arXiv:2410.01104. – 2024.
18. *Zhang Y., Backurs A., Bubeck S., Eldan R., Gunasekar S., Wagner T.* Unveiling transformers with lego: a synthetic reasoning task // arXiv preprint arXiv:2206.04301. – 2022.
19. *Li C., Liu J., Chen Y., Jia Y., Li Z.* KunlunBaize: LLM with Multi-Scale Convolution and Multi-Token Prediction Under TransformerX Framework // arXiv preprint arXiv:2503.04784. – 2025.
20. *Hao S., Sukhbaatar S., Su D., Li X., Hu Z., Weston J., Tian Y.* Training large language models to reason in a continuous latent space // arXiv preprint arXiv:2412.06769. – 2024.
21. *Зайцев Д.В.* Почему большие языковые модели не (всегда) рассуждают как люди? // Вестник Московского университета. Серия 7. Философия. – 2024. – № 1. – С. 76-93.
22. *Pagnoni A., Pasunuru R., Rodriguez P., Nguyen J., Muller B., Li M., Zhou C., Yu L., Weston J., Zettlemoyer L., others*. Byte Latent Transformer: Patches Scale Better Than Tokens // arXiv preprint arXiv:2412.09871. – 2024.
23. *Xia B., Shen B., Zhu D., Zhang D., Wang G., Zhang H., Liu H., Xiao J., Dong J., Zhao L., others*. MiMo: Unlocking the Reasoning Potential of Language Model–From Pretraining to Posttraining // arXiv preprint arXiv:2505.07608. – 2025.
24. *Zelikman E., Wu Y., Mu J., Goodman N.* Star: Bootstrapping reasoning with reasoning // Advances in Neural Information Processing Systems. – 2022. – Vol. 35. – P. 15476-15488.
25. *Wu T., Lan J., Yuan W., Jiao J., Weston J., Sukhbaatar S.* Thinking LLMs: General Instruction Following with Thought Generation // arXiv preprint arXiv:2410.10630. – 2024.
26. *Li W.-D., Hu K., Larsen C., Wu Y., Alford S., Woo C., Dunn S. M., Tang H., Naim M., Nguyen D., others.* Combining induction and transduction for abstract reasoning // arXiv preprint arXiv:2411.02272. – 2024.
27. *Kumar A., Zhuang V., Agarwal R., Su Y., Co-Reyes J. D., Singh A., Baumli K., Iqbal S., Bishop C., Roelofs R., others.* Training language models to self-correct via reinforcement learning // arXiv preprint arXiv:2409.12917. – 2024.
28. *Chen J. C.-Y., Wang Z., Palangi H., Han R., Ebrahimi S., Le L., Perot V., Mishra S., Bansal M., Lee C.-Y., others.* Reverse Thinking Makes LLMs Stronger Reasoners // arXiv preprint arXiv:2411.19865. – 2024.

29. *Muennighoff N., Yang Z., Shi W., Li X. L., Fei-Fei L., Hajishirzi H., Zettlemoyer L., Liang P., Candès E., Hashimoto T.* s1: Simple test-time scaling // arXiv preprint arXiv:2501.19393. – 2025.
30. *Ye Y., Huang Z., Xiao Y., Chern E., Xia S., Liu P.* LIMO: Less is More for Reasoning. – 2025.
31. Learning to Reason with LLMs. – URL: https://openai.com/index/learning-to-reason-with-llms/ (дата обращения: 22.11.2024).
32. *Zeng Z., Cheng Q., Yin Z., Wang B., Li S., Zhou Y., Guo Q., Huang X., Qiu X.* Scaling of Search and Learning: A Roadmap to Reproduce o1 from Reinforcement Learning Perspective. – 2024.
33. *Guo D., Yang D., Zhang H., Song J., Zhang R., Xu R., Zhu Q., Ma S., Wang P., Bi X., others*. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning // arXiv preprint arXiv:2501.12948. – 2025.
34. *Huan M., Li Y., Zheng T., Xu X., Kim S., Du M., Poovendran R., Neubig G., Yue X.* Does Math Reasoning Improve General LLM Capabilities? Understanding Transferability of LLM Reasoning. – 2025.
35. *Wei J., Wang X., Schuurmans D., Bosma M., Xia F., Chi E., Le Q. V., Zhou D., others*. Chain-of-thought prompting elicits reasoning in large language models // Advances in neural information processing systems. – 2022. – Vol. 35. – P. 24824-24837.
36. *Kojima T., Gu S.S., Reid M., Matsuo Y., Iwasawa Y.* Large language models are zero-shot reasoners // Advances in neural information processing systems. – 2022. – Vol. 35. – P. 22199-22213.
37. *Long J.* Large language model guided tree-of-thought // arXiv preprint arXiv:2305.08291. – 2023.
38. *Besta M., Blach N., Kubicek A., Gerstenberger R., Podstawski M., Gianinazzi L., Gajda J., Lehmann T., Niewiadomski H., Nyczyk P., others*. Graph of thoughts: Solving elaborate problems with large language models. – 2024. – P. 17682-17690.
39. *Wang X., Wei J., Schuurmans D., Le Q., Chi E., Narang S., Chowdhery A., Zhou D.* Self-consistency improves chain of thought reasoning in language models // arXiv preprint arXiv:2203.11171. – 2022.
40. *Madaan A., Tandon N., Gupta P., Hallinan S., Gao L., Wiegreffe S., Alon U., Dziri N., Prabhumoye S., Yang Y., others*. Self-refine: Iterative refinement with self-feedback // Advances in Neural Information Processing Systems. – 2024. – Vol. 36.
41. *Miao N., Teh Y.W., Rainforth T.* Selfcheck: Using llms to zero-shot check their own step-by-step reasoning // arXiv preprint arXiv:2308.00436. – 2023.
42. *Shinn N., Cassano F., Gopinath A., Narasimhan K., Yao S.* Reflexion: Language agents with verbal reinforcement learning // Advances in Neural Information Processing Systems. – 2024. – Vol. 36.
43. *Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W., Rocktäschel T., others*. Retrieval-augmented generation for knowledge-intensive nlp tasks // Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – P. 9459-9474.
44. *Jiang J., Chen J., Li J., Ren R., Wang S., Zhao W. X., Song Y., Zhang T.* RAG-Star: Enhancing Deliberative Reasoning with Retrieval Augmented Verification and Refinement // arXiv preprint arXiv:2412.12881. – 2024.
45. *Zhou P., Pujara J., Ren X., Chen X., Cheng H.-T., Le Q. V., Chi E. H., Zhou D., Mishra S., Zheng H. S.* Self-discover: Large language models self-compose reasoning structures // arXiv preprint arXiv:2402.03620. – 2024.
46. *Akyürek E., Damani M., Qiu L., Guo H., Kim Y., Andreas J.* The Surprising Effectiveness of Test-Time Training for Abstract Reasoning // arXiv preprint arXiv:2411.07279. – 2024.
47. *Snell C., Lee J., Xu K., Kumar A.* Scaling llm test-time compute optimally can be more effective than scaling model parameters // arXiv preprint arXiv:2408.03314. – 2024.
48. *Zhong T., Liu Z., Pan Y., Zhang Y., Zhou Y., Liang S., Wu Z., Lyu Y., Shu P., Yu X., others*. Evaluation of openai o1: Opportunities and challenges of agi // arXiv preprint arXiv:2409.18486. – 2024.
49. *Zhao Y., Yin H., Zeng B., Wang H., Shi T., Lyu C., Wang L., Luo W., Zhang K.* Marco-o1: Towards Open Reasoning Models for Open-Ended Solutions // arXiv preprint arXiv:2411.14405. – 2024.
50. Scaling test-time compute - a Hugging Face Space by HuggingFaceH4. – URL: https://huggingface.co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute (дата обращения: 17.12.2024).
51. *Zheng C., Zhang Z., Zhang B., Lin R., Lu K., Yu B., Liu D., Zhou J., Lin J.* ProcessBench: Identifying Process Errors in Mathematical Reasoning // arXiv preprint arXiv:2412.06559. – 2024.
52. *Schick T., Dwivedi-Yu J., Dessì R., Raileanu R., Lomeli M., Hambro E., Zettlemoyer L., Cancedda N., Scialom T.* Toolformer: Language models can teach themselves to use tools // Advances in Neural Information Processing Systems. – 2023. – Vol. 36. – P. 68539-68551.
53. *Li C., Liang J., Zeng A., Chen X., Hausman K., Sadigh D., Levine S., Fei-Fei L., Xia F., Ichter B.* Chain of code: Reasoning with a language model-augmented code emulator // arXiv preprint arXiv:2312.04474. – 2023.
54. *Chen W., Ma X., Wang X., Cohen W.W.* Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks // arXiv preprint arXiv:2211.12588. – 2022.

55. *Gao L., Madaan A., Zhou S., Alon U., Liu P., Yang Y., Callan J., Neubig G.* Pal: Program-aided language models. – 2023. – P. 10764-10799.
56. *Yao S., Zhao J., Yu D., Du N., Shafran I., Narasimhan K., Cao Y.* React: Synergizing reasoning and acting in language models // arXiv preprint arXiv:2210.03629. – 2022.
57. *Gou Z., Shao Z., Gong Y., Shen Y., Yang Y., Huang M., Duan N., Chen W.* Tora: A tool-integrated reasoning agent for mathematical problem solving // arXiv preprint arXiv:2309.17452. – 2023.
58. *Motwani S. R., Smith C., Das R. J., Rybchuk M., Torr P.H., Laptev I., Pizzati F., Clark R., Witt C.S.* de MALT: Improving Reasoning with Multi-Agent LLM Training // arXiv preprint arXiv:2412.01928. – 2024.
59. *Wang Q., Wang Z., Su Y., Tong H., Song Y.* Rethinking the Bounds of LLM Reasoning: Are Multi-Agent Discussions the Key? // arXiv preprint arXiv:2402.18272. – 2024.
60. *Chen X., Xu J., Liang T., He Z., Pang J., Yu D., Song L., Liu Q., Zhou M., Zhang Z., others.* Do not think that much for 2+ 3=? on the overthinking of o1-like llms // arXiv preprint arXiv:2412.21187. – 2024.
61. *Pu X., Saxon M., Hua W., Wang W.Y.* THOUGHTTERMINATOR: Benchmarking, Calibrating, and Mitigating Overthinking in Reasoning Models // arXiv preprint arXiv:2504.13367. – 2025.
62. *Sui Y., Chuang Y.-N., Wang G., Zhang J., Zhang T., Yuan J., Liu H., Wen A., Chen H., Hu X., others*. Stop Overthinking: A Survey on Efficient Reasoning for Large Language Models // arXiv preprint arXiv:2503.16419. – 2025.
63. *Fang G., Ma X., Wang X.* Thinkless: LLM Learns When to Think // arXiv preprint arXiv:2505.13379. – 2025.
64. *Chen Y., Benton J., Radhakrishnan A., Uesato J., Denison C., Schulman J., Somani A., Hase P., Wagner M., Roger F., others*. Reasoning Models Don't Always Say What They Think // arXiv preprint arXiv:2505.05410. – 2025.
65. *Yue Y., Chen Z., Lu R., Zhao A., Wang Z., Song S., Huang G.* Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? // arXiv preprint arXiv:2504.13837. – 2025.
66. *Liu Z., Chen C., Li W., Pang T., Du C., Lin M.* There May Not be Aha Moment in R1-Zero-like Training — A Pilot Study. – 2025.
67. *Mancoridis M., Weeks B., Vafa K., Mullainathan S.* Potemkin Understanding in Large Language Models // arXiv preprint arXiv:2506.21521. – 2025.
68. *Shojaee P., Mirzadeh I., Alizadeh K., Horton M., Bengio S., Farajtabar M.* The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity // arXiv preprint arXiv:2506.06941. – 2025.
69. *Opus C., Lawsen A.* The Illusion of the Illusion of Thinking // arXiv preprint ArXiv:2506.09250. – 2025.
70. *Malek A., Ge J., Jin C., György A., Szepesvári C.* Frontier LLMs Still Struggle with Simple Reasoning Tasks // arXiv preprint arXiv:2507.07313. – 2025.

REFERENCES

1. *Vaswani A.* Attention is all you need, *Advances in Neural Information Processing Systems*, 2017.
2. *Mirzadeh I., Alizadeh K., Shahrokhi H., Tuzel O., Bengio S., Farajtabar M.* Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models, *arXiv preprint arXiv:2410.05229*, 2024.
3. *Minaee S., Mikolov T., Nikzad N., Chenaghlu M., Socher R., Amatriain X., Gao J.* Large language models: A survey, *arXiv preprint arXiv:2402.06196*, 2024.
4. *Berglund L., Tong M., Kaufmann M., Balesni M., Stickland A. C., Korbak T., Evans O.* The reversal curse: Llms trained on" a is b" fail to learn" b is a", *arXiv preprint arXiv:2309.12288*, 2023.
5. *Wu Z., Qiu L., Ross A., Akyürek E., Chen B., Wang B., Kim N., Andreas J., Kim Y.* Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks, *arXiv preprint arXiv:2307.02477*, 2023.
6. *Wang L., Ma C., Feng X., Zhang Z., Yang H., Zhang J., Chen Z., Tang J., Chen X., Lin Y., others.* A survey on large language model based autonomous agents, *Frontiers of Computer Science*, 2024, Vol. 18, No. 6, pp. 186345.
7. Arkoudas K. GPT-4 can't reason, *arXiv preprint arXiv:2308.03762*, 2023.
8. *Chang Y., Wang X., Wang J., Wu Y., Yang L., Zhu K., Chen H., Yi X., Wang C., Wang Y., Ye W., Zhang Y., Chang Y., Yu P. S., Yang Q., Xie X.* A Survey on Evaluation of Large Language Models, *ACM Trans. Intell. Syst. Technol*, 2024, Vol. 15, No. 3. DOI: 10.1145/3641289.
9. *Mahowald K., Ivanova A.A., Blank I.A., Kanwisher N., Tenenbaum J.B., Fedorenko E.* Dissociating language and thought in large language models, *Trends in Cognitive Sciences*, 2024.

10. *Plaat A., Wong A., Verberne S., Broekens J., Stein N. van, Back T.* Reasoning with large language models, a survey, *arXiv preprint arXiv:2407.11511*, 2024.
11. *Penedo G., Kydlíček H., Lozhkov A., Mitchell M., Raffel C., Von Werra L., Wolf T., others*. The fineweb datasets: Decanting the web for the finest text data at scale, *arXiv preprint arXiv:2406.17557*, 2024.
12. *Shao Z., Wang P., Zhu Q., Xu R., Song J., Bi X., Zhang H., Zhang M., Li Y., Wu Y., others.* Deepseekmath: Pushing the limits of mathematical reasoning in open language models, *arXiv preprint arXiv:2402.03300*, 2024.
13. *Aryabumi V., Su Y., Ma R., Morisot A., Zhang I., Locatelli A., Fadaee M., Üstün A., Hooker S.* To Code, or Not To Code? Exploring Impact of Code in Pre-training, *arXiv preprint arXiv:2408.10914*, 2024.
14. *Morishita T., Morio G., Yamaguchi A., Sogawa Y.* Enhancing Reasoning Capabilities of LLMs via Principled Synthetic Logic Corpus, *arXiv preprint arXiv:2411.12498*, 2024.
15. *Abdin M., Aneja J., Behl H., Bubeck S., Eldan R., Gunasekar S., Harrison M., Hewett R. J., Javaheripi M., Kauffmann P., others*. Phi-4 Technical Report, *arXiv preprint arXiv:2412.08905*, 2024.
16. *Chen A., Li A., Gong B., Jiang B., Fei B., Yang B., Shan B., Yu C., Wang C., Zhu C., others*. MiniMax-M1: Scaling Test-Time Compute Efficiently with Lightning Attention, *arXiv preprint arXiv:2506.13585*, 2025.
17. *Veličković P., Perivolaropoulos C., Barbero F., Pascanu R.* Softmax is not enough (for sharp out-of-distribution), *arXiv preprint arXiv:2410.01104*, 2024.
18. *Zhang Y., Backurs A., Bubeck S., Eldan R., Gunasekar S., Wagner T.* Unveiling transformers with lego: a synthetic reasoning task, *arXiv preprint arXiv:2206.04301*, 2022.
19. *Li C., Liu J., Chen Y., Jia Y., Li Z.* KunlunBaize: LLM with Multi-Scale Convolution and Multi-Token Prediction Under TransformerX Framework, *arXiv preprint arXiv:2503.04784*, 2025.
20. *Hao S., Sukhbaatar S., Su D., Li X., Hu Z., Weston J., Tian Y.* Training large language models to reason in a continuous latent space, *arXiv preprint arXiv:2412.06769*, 2024.
21. *Zaytsev D.V.* Pochemu bol'shie yazykovye modeli ne (vsegda) rassuzhdayut kak lyudi? [Why don't large language models (always) reason like humans?], *Vestnik Moskovskogo universiteta. Seriya 7. Filosofiya* [Moscow University Bulletin. Series 7. Philosophy], 2024, No. 1, pp. 76-93.
22. *Pagnoni A., Pasunuru R., Rodriguez P., Nguyen J., Muller B., Li M., Zhou C., Yu L., Weston J., Zettlemoyer L., others*. Byte Latent Transformer: Patches Scale Better Than Tokens, *arXiv preprint arXiv:2412.09871*, 2024.
23. *Xia B., Shen B., Zhu D., Zhang D., Wang G., Zhang H., Liu H., Xiao J., Dong J., Zhao L., others*. MiMo: Unlocking the Reasoning Potential of Language Model–From Pretraining to Posttraining, *arXiv preprint arXiv:2505.07608*, 2025.
24. *Zelikman E., Wu Y., Mu J., Goodman N.* Star: Bootstrapping reasoning with reasoning, *Advances in Neural Information Processing Systems*, 2022, Vol. 35, pp. 15476-15488.
25. *Wu T., Lan J., Yuan W., Jiao J., Weston J., Sukhbaatar S.* Thinking LLMs: General Instruction Following with Thought Generation // arXiv preprint arXiv:2410.10630. – 2024.
26. *Li W.-D., Hu K., Larsen C., Wu Y., Alford S., Woo C., Dunn S. M., Tang H., Naim M., Nguyen D., others*. Combining induction and transduction for abstract reasoning, *arXiv preprint arXiv:2411.02272*, 2024.
27. *Kumar A., Zhuang V., Agarwal R., Su Y., Co-Reyes J. D., Singh A., Baumli K., Iqbal S., Bishop C., Roelofs R., others.* Training language models to self-correct via reinforcement learning, *arXiv preprint arXiv:2409.12917*, 2024.
28. *Chen J. C.-Y., Wang Z., Palangi H., Han R., Ebrahimi S., Le L., Perot V., Mishra S., Bansal M., Lee C.-Y., others.* Reverse Thinking Makes LLMs Stronger Reasoners, *arXiv preprint arXiv:2411.19865*, 2024.
29. *Muennighoff N., Yang Z., Shi W., Li X. L., Fei-Fei L., Hajishirzi H., Zettlemoyer L., Liang P., Candès E., Hashimoto T.* s1: Simple test-time scaling, *arXiv preprint arXiv:2501.19393*, 2025.
30. *Ye Y., Huang Z., Xiao Y., Chern E., Xia S., Liu P.* LIMO: Less is More for Reasoning, 2025.
31. Learning to Reason with LLMs. Available at: https://openai.com/index/learning-to-reason-with-llms/ (accessed 22 November 2024).
32. *Zeng Z., Cheng Q., Yin Z., Wang B., Li S., Zhou Y., Guo Q., Huang X., Qiu X.* Scaling of Search and Learning: A Roadmap to Reproduce o1 from Reinforcement Learning Perspective, 2024.
33. *Guo D., Yang D., Zhang H., Song J., Zhang R., Xu R., Zhu Q., Ma S., Wang P., Bi X., others*. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, *arXiv preprint arXiv:2501.12948*, 2025.
34. *Huan M., Li Y., Zheng T., Xu X., Kim S., Du M., Poovendran R., Neubig G., Yue X.* Does Math Reasoning Improve General LLM Capabilities? Understanding Transferability of LLM Reasoning, 2025.

35. *Wei J., Wang X., Schuurmans D., Bosma M., Xia F., Chi E., Le Q. V., Zhou D., others*. Chain-of-thought prompting elicits reasoning in large language models, *Advances in neural information processing systems*, 2022, Vol. 35, pp. 24824-24837.
36. *Kojima T., Gu S.S., Reid M., Matsuo Y., Iwasawa Y.* Large language models are zero-shot reasoners, *Advances in neural information processing systems*, 2022, Vol. 35, pp. 22199-22213.
37. *Long J.* Large language model guided tree-of-thought, *arXiv preprint arXiv:2305.08291*, 2023.
38. *Besta M., Blach N., Kubicek A., Gerstenberger R., Podstawski M., Gianinazzi L., Gajda J., Lehmann T., Niewiadomski H., Nyczyk P., others*. Graph of thoughts: Solving elaborate problems with large language models, 2024, pp. 17682-17690.
39. *Wang X., Wei J., Schuurmans D., Le Q., Chi E., Narang S., Chowdhery A., Zhou D.* Self-consistency improves chain of thought reasoning in language models, *arXiv preprint arXiv:2203.11171*, 2022.
40. *Madaan A., Tandon N., Gupta P., Hallinan S., Gao L., Wiegreffe S., Alon U., Dziri N., Prabhumoye S., Yang Y., others*. Self-refine: Iterative refinement with self-feedback, *Advances in Neural Information Processing Systems*, 2024, Vol. 36.
41. *Miao N., Teh Y.W., Rainforth T.* Selfcheck: Using llms to zero-shot check their own step-by-step reasoning, *arXiv preprint arXiv:2308.00436*, 2023.
42. *Shinn N., Cassano F., Gopinath A., Narasimhan K., Yao S.* Reflexion: Language agents with verbal reinforcement learning, *Advances in Neural Information Processing Systems*, 2024, Vol. 36.
43. *Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W., Rocktäschel T., others.* Retrieval-augmented generation for knowledge-intensive nlp tasks, *Advances in Neural Information Processing Systems*, 2020, Vol. 33, pp. 9459-9474.
44. *Jiang J., Chen J., Li J., Ren R., Wang S., Zhao W. X., Song Y., Zhang T.* RAG-Star: Enhancing Deliberative Reasoning with Retrieval Augmented Verification and Refinement, *arXiv preprint arXiv:2412.12881*, 2024.
45. *Zhou P., Pujara J., Ren X., Chen X., Cheng H.-T., Le Q.V., Chi E.H., Zhou D., Mishra S., Zheng H.S.* Self-discover: Large language models self-compose reasoning structures, *arXiv preprint arXiv:2402.03620*, 2024.
46. *Akyürek E., Damani M., Qiu L., Guo H., Kim Y., Andreas J.* The Surprising Effectiveness of Test-Time Training for Abstract Reasoning, *arXiv preprint arXiv:2411.07279*, 2024.
47. *Snell C., Lee J., Xu K., Kumar A.* Scaling llm test-time compute optimally can be more effective than scaling model parameters, *arXiv preprint arXiv:2408.03314*, 2024.
48. *Zhong T., Liu Z., Pan Y., Zhang Y., Zhou Y., Liang S., Wu Z., Lyu Y., Shu P., Yu X., others*. Evaluation of openai o1: Opportunities and challenges of agi, *arXiv preprint arXiv:2409.18486*, 2024.
49. *Zhao Y., Yin H., Zeng B., Wang H., Shi T., Lyu C., Wang L., Luo W., Zhang K.* Marco-o1: Towards Open Reasoning Models for Open-Ended Solutions, *arXiv preprint arXiv:2411.14405*, 2024.
50. Scaling test-time compute - a Hugging Face Space by HuggingFaceH4. Available at: https://huggingface.co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute (accessed 17 Desenber 2024).
51. *Zheng C., Zhang Z., Zhang B., Lin R., Lu K., Yu B., Liu D., Zhou J., Lin J.* ProcessBench: Identifying Process Errors in Mathematical Reasoning, *arXiv preprint arXiv:2412.06559*, 2024.
52. *Schick T., Dwivedi-Yu J., Dessì R., Raileanu R., Lomeli M., Hambro E., Zettlemoyer L., Cancedda N., Scialom T.* Toolformer: Language models can teach themselves to use tools, *Advances in Neural Information Processing Systems*, 2023, Vol. 36, pp. 68539-68551.
53. *Li C., Liang J., Zeng A., Chen X., Hausman K., Sadigh D., Levine S., Fei-Fei L., Xia F., Ichter B.* Chain of code: Reasoning with a language model-augmented code emulator, *arXiv preprint arXiv:2312.04474*, 2023.
54. *Chen W., Ma X., Wang X., Cohen W.W.* Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks, *arXiv preprint arXiv:2211.12588*, 2022.
55. *Gao L., Madaan A., Zhou S., Alon U., Liu P., Yang Y., Callan J., Neubig G.* Pal: Program-aided language models, 2023, pp. 10764-10799.
56. *Yao S., Zhao J., Yu D., Du N., Shafran I., Narasimhan K., Cao Y.* React: Synergizing reasoning and acting in language models, *arXiv preprint arXiv:2210.03629*, 2022.
57. *Gou Z., Shao Z., Gong Y., Shen Y., Yang Y., Huang M., Duan N., Chen W.* Tora: A tool-integrated reasoning agent for mathematical problem solving, *arXiv preprint arXiv:2309.17452*, 2023.
58. *Motwani S. R., Smith C., Das R. J., Rybchuk M., Torr P.H., Laptev I., Pizzati F., Clark R., Witt C.S.* de MALT: Improving Reasoning with Multi-Agent LLM Training, *arXiv preprint arXiv:2412.01928*, 2024.
59. *Wang Q., Wang Z., Su Y., Tong H., Song Y.* Rethinking the Bounds of LLM Reasoning: Are Multi-Agent Discussions the Key?, *arXiv preprint arXiv:2402.18272*, 2024.

60. *Chen X., Xu J., Liang T., He Z., Pang J., Yu D., Song L., Liu Q., Zhou M., Zhang Z., others*. Do not think that much for 2+ 3=? on the overthinking of o1-like llms, *arXiv preprint arXiv:2412.21187*, 2024.
61. *Pu X., Saxon M., Hua W., Wang W.Y.* THOUGHTTERMINATOR: Benchmarking, Calibrating, and Mitigating Overthinking in Reasoning Models, *arXiv preprint arXiv:2504.13367*, 2025.
62. *Sui Y., Chuang Y.-N., Wang G., Zhang J., Zhang T., Yuan J., Liu H., Wen A., Chen H., Hu X., others*. Stop Overthinking: A Survey on Efficient Reasoning for Large Language Models, *arXiv preprint arXiv:2503.16419*, 2025.
63. *Fang G., Ma X., Wang X.* Thinkless: LLM Learns When to Think, *arXiv preprint arXiv:2505.13379*, 2025.
64. *Chen Y., Benton J., Radhakrishnan A., Uesato J., Denison C., Schulman J., Somani A., Hase P., Wagner M., Roger F., others*. Reasoning Models Don't Always Say What They Think, *arXiv preprint arXiv:2505.05410*, 2025.
65. *Yue Y., Chen Z., Lu R., Zhao A., Wang Z., Song S., Huang G.* Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?, *arXiv preprint arXiv:2504.13837*, 2025.
66. *Liu Z., Chen C., Li W., Pang T., Du C., Lin M.* There May Not be Aha Moment in R1-Zero-like Training — A Pilot Study, 2025.
67. *Mancoridis M., Weeks B., Vafa K., Mullainathan S.* Potemkin Understanding in Large Language Models, *arXiv preprint arXiv:2506.21521*, 2025.
68. *Shojaee P., Mirzadeh I., Alizadeh K., Horton M., Bengio S., Farajtabar M.* The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity, *arXiv preprint arXiv:2506.06941*, 2025.
69. *Opus C., Lawsen A.* The Illusion of the Illusion of Thinking, *arXiv preprint ArXiv:2506.09250*, 2025.
70. *Malek A., Ge J., Jin C., György A., Szepesvári C.* Frontier LLMs Still Struggle with Simple Reasoning Tasks, *arXiv preprint arXiv:2507.07313*, 2025.

Савинов Владимир Борисович – Балтийский федеральный университет имени Иммануила Канта; e-mail: vsavinov@kantiana.ru; г. Калининград, Россия; Балтийский центр нейротехнологий и искусственного интеллекта; аспирант.

Шушарина Наталья Николаевна – Балтийский федеральный университет имени Иммануила Канта; e-mail: nshusharina@kantiana.ru; г. Калининград, Россия; Балтийский центр нейротехнологий и искусственного интеллекта; к. пед. н.; начальник управления развития и инновационной деятельности; старший научный сотрудник.

Savinov Vladimir Borisovich – Immanuel Kant Baltic Federal University; e-mail: vsavinov@kantiana.ru; Kaliningrad, Russia; Center for Neurotechnologies and Artificial Intelligence; postgraduate student.

Shusharina Natalia Nikolaevna – Immanuel Kant Baltic Federal University; e-mail: nshusharina@kantiana.ru; Kaliningrad, Russia; Baltic Center for Neurotechnologies and Artificial Intelligence; cand. of ped. sc.; head of the Department of Development and Innovation Activity; senior researcher.

УДК 004.722 DOI 10.18522/2311-3103-2026-1-270-284

А.В. Бобряков, С.А. Прокопенко

РАННЕЕ ВЫЯВЛЕНИЕ ПРОИЗВОДСТВЕННОГО БРАКА НА МЕЛКОСЕРИЙНЫХ ПРОИЗВОДСТВАХ С ИСПОЛЬЗОВАНИЕМ НЕЙРО-НЕЧЕТКИХ СИСТЕМ

***Постановка задачи**: Повышение требований к качеству продукции на мелкосерийных производствах и сложность раннего выявления производственного брака актуализируют необходимость разработки инновационных подходов к прогнозированию и контролю дефектов на ранних этапах производства сложных технических объектов. Известные методы, применяемые в массовом производстве, не подходят для мелкосерийных производств вследствие высокой изменчивости технологических процессов и недостаточности данных для традиционного статистического анализа. **Целью работы** является снижение уровня производственного брака путем раннего выявления отклонений на подготовительных этапах производства. Предлагается использование ней-*