

**Оригинальная статья**

DOI: 10.17748/2686–9969–2026-9-1-147-158

УДК 004.89:51/37

END LCJQRA

**ПРОБЛЕМА ОТОБРАЖЕНИЯ МАТЕМАТИЧЕСКИХ СИМВОЛОВ
В ДИАЛОГОВЫХ СИСТЕМАХ НА ОСНОВЕ БОЛЬШИХ
ЯЗЫКОВЫХ МОДЕЛЕЙ И АРХИТЕКТУРА ПРЕПРОЦЕССОРА
ДЛЯ ОБРАЗОВАТЕЛЬНЫХ ЦЕЛЕЙ****Владимир Алексеевич Липатов**

Алтайский государственный педагогический университет

г. Барнаул, Россия

vladimir.lipatov1998@gmail.com

Аннотация. Интеграция больших языковых моделей в образовательный процесс по математическим дисциплинам сопряжена с технической проблемой некорректного отображения специализированной математической нотации в интерфейсах мессенджеров. В статье представлена архитектура и реализация автономного программного модуля-препроцессора MathTeXBot, интегрируемого в Telegram-бота на базе отечественной языковой модели GigaChat. Методологическую основу исследования составляют анализ существующих решений для рендеринга формул и разработка алгоритма гибридного парсинга текстовых ответов модели с последующей локальной генерацией графических изображений. Принципиальной особенностью предлагаемого подхода является полная автономность модуля, не требующая обращения к внешним, в том числе зарубежным, сервисам рендеринга, что обеспечивает технологический суверенитет разрабатываемого решения. В результате исследования создан функционирующий прототип, осуществляющий перехват ответов от GigaChat API, выделение математических блоков посредством регулярных выражений и их преобразование в PNG-изображения с использованием локальной установки LaTeX и инструментов командной строки. Проведено сравнение предлагаемого подхода с существующими образовательными платформами и специализированными ботами. Целью экспериментальной апробации являлась оценка производительности разработанного прототипа и качества восприятия сгенерированного математического контента пользователями. В заключении обозначены перспективы развития модуля,

включая поддержку семантического анализа и интеграцию с системами компьютерной алгебры.

Ключевые слова: GigaChat, большие языковые модели, Telegram-бот, рендеринг LaTeX, математическое образование, технологический суверенитет, автономный препроцессор, цифровизация образования

Цитирование: *липатов* Липатов В. А. Проблема отображения математических символов в диалоговых системах на основе больших языковых моделей и архитектура препроцессора для образовательных целей // Педагогика: история, перспективы 2026. Том 9. № 1. С. 147-158 END LCJQRA
DOI: 10.17748/2686-9969-2026-9-1-147-158

Original article

THE PROBLEM OF DISPLAYING MATHEMATICAL SYMBOLS IN DIALOGUE SYSTEMS BASED ON LARGE LANGUAGE MODELS AND THE ARCHITECTURE OF A PREPROCESSOR FOR EDUCATIONAL PURPOSES

Vladimir A. Lipatov
Altai State Pedagogical University
Barnaul, Russia
vladimir.lipatov1998@gmail.com

Abstract. Integrating large language models into mathematical education poses a technical challenge: the incorrect display of specialized mathematical notation in instant messaging interfaces. This article presents the architecture and implementation of a standalone preprocessor software module, MathTeXBot, integrated into a Telegram bot based on the domestic GigaChat language model. The methodological basis of the study is an analysis of existing formula rendering solutions and the development of an algorithm for hybrid parsing of the model's text responses, followed by local generation of graphic images. A key feature of the proposed approach is the module's complete autonomy, requiring no recourse to external, including foreign, rendering services, ensuring the technological sovereignty of the developed solution. The study resulted in the creation of a functioning prototype that intercepts responses from the GigaChat API, extracts mathematical blocks using regular expressions, and converts them into PNG images using a local LaTeX installation and command-line tools. The proposed approach is compared with existing educational platforms and specialized bots. The aim of the experimental approbation was to evaluate the performance of the developed prototype and the quality of user perception of the generated mathematical content. The conclusion outlines the module's future

development prospects, including support for semantic analysis and integration with computer algebra systems.

Keywords: GigaChat, large language models, Telegram bot, LaTeX rendering, mathematics education, technological sovereignty, autonomous preprocessor, digitalization of education

For citation: Lipatov V.A. The problem of displaying mathematical symbols in dialogue systems based on large language models and the architecture of a preprocessor for educational purposes. Pedagogy: History, Prospects 2026. Vol. 9. No. 1. P. 147-158 (In Russ.). END LCJQRA

DOI: 10.17748/2686-9969-2026-9-1-147-158

я

Современный этап развития цифровых технологий в образовании характеризуется активным внедрением генеративных нейросетевых моделей в учебные процессы различных уровней. Особое значение приобретает использование больших языковых моделей (Large Language Models, LLM) в преподавании математических дисциплин, где они могут выполнять функции виртуальных тьюторов, генераторов учебных задач и средств объяснения сложных концепций. Среди отечественных разработок в данной области выделяется модель GigaChat (ПАО «Сбер»), которая, согласно опубликованным данным, демонстрирует улучшение качества решения математических задач на 25% по тестам GSM8K и MATH, а также обладает функциональностью распознавания формул на графических изображениях [1, 6].

Однако наличие у языковой модели предметных знаний не гарантирует их корректной трансляции конечному пользователю. Ключевая проблема, снижающая дидактический потенциал рассматриваемых систем при их использовании в популярных мессенджерах (Telegram, WhatsApp), заключается в неспособности стандартных интерфейсов отображать математическую нотацию. Языковые модели обучаются на корпусах научно-технической литературы, где доминирует разметка LaTeX, и, соответственно, генерируют ответы с использованием данной разметки (например, $\int_a^b f(x)dx$). Клиентские приложения при этом выводят разметку как обычный текст либо, в случае с юникод-символами, отображают иероглифы или символы подстановки из-за отсутствия поддержки соответствующих диапазонов шрифтов [2]. Подобное искажение делает невозможным адекватное восприятие учебного материала обучающимися, особенно на начальных этапах подготовки.

В существующих исследованиях указывается, что основными причинами искажений выступают несоответствие токенизации математических символов в LLM и конфликт кодировок при передаче данных [7]. Распространенные подходы к решению данной проблемы предполагают использование внешних сервисов рендеринга, таких как CodeCogs или Google Chart API, что создает зависимость от зару-

бежных цифровых инфраструктур и противоречит задачам обеспечения технологического суверенитета Российской Федерации в сфере образования [3].

Цель настоящей работы заключается в разработке и апробации архитектуры автономного программного модуля-препроцессора, интегрируемого между API языковой модели (на примере GigaChat) и пользовательским интерфейсом Telegram, обеспечивающего корректный рендеринг LaTeX-формул в образовательном чат-боте без обращения к внешним сервисам.

Обзор литературы

Проблема корректного отображения математического текста в цифровых средах имеет два основных направления решений: клиентский рендеринг с использованием JavaScript-библиотек и серверная генерация растровых изображений.

В веб-ориентированных образовательных средах, таких как Coursera и Stepik, задача решается посредством встраивания JavaScript-библиотек (MathJax, KaTeX) непосредственно в HTML-страницу. MathJax, поддерживаемый рядом академических институтов (например, University of Reading использует его в системе Blackboard [10]), обеспечивает высокое качество типографской верстки и доступность для программ экранного доступа, однако требует вычислительных ресурсов на стороне клиента. KaTeX, разработанный в Khan Academy, ориентирован на скорость рендеринга, но поддерживает не все пакеты расширения LaTeX [4, 9]. Указанные подходы эффективны в веб-среде, однако неприменимы в приложениях с ограниченным функционалом, таких как Telegram, где выполнение пользовательского JavaScript-кода невозможно.

Для мессенджеров распространен метод конвертации LaTeX-разметки в растровые изображения. Платформа Pandoc поддерживает флаг `--webtex`, позволяющий делегировать рендеринг внешним сервисам (например, CodeCogs), которые возвращают готовое изображение [9]. Существуют также специализированные боты, такие как @jh0ker_texbot с открытым исходным кодом на языке Python, преобразующие отправленное пользователем LaTeX-выражение в графический файл [8]. Однако указанные боты функционируют пассивно: они не интегрированы с языковыми моделями и требуют от пользователя ручного ввода каждой формулы, что нарушает непрерывность диалогового взаимодействия.

В контексте диалоговых систем на базе LLM, согласно тестированию, проведенному ИТС.ru, модели Gemini и Grok в мобильных версиях часто отображают нечитаемые символы вместо формул, тогда как ChatGPT в веб-версии демонстрирует лучшие результаты благодаря встроенному рендереру [2]. Данное наблюдение подтверждает, что проблема обусловлена не семантическими ограничениями модели, а недостатками пользовательского интерфейса.

Анализ литературы позволяет сделать вывод о существовании пробела: отсутствует программное решение, которое автоматически преобразовывало бы LaTeX-вывод LLM в графические изображения, пригодные для отправки в мессенджеры, с использованием исключительно локальных вычислительных ресурсов, что

особенно важно в контексте обеспечения технологической независимости отечественных образовательных платформ. Настоящее исследование направлено на устранение указанного пробела.

Объектом исследования выступает процесс взаимодействия пользователя с языковой моделью по вопросам математического содержания в среде Telegram.

Предмет исследования – архитектура автономного программного обеспечения для трансформации текстового вывода нейросетевой модели в визуализированные математические объекты без использования внешних сетевых сервисов.

Материалы и методы

Для реализации прототипа были выбраны следующие инструменты:

1. Язык программирования: Python 3.10+ как стандарт де-факто для интеграции с LLM API и наличия необходимых библиотек.
2. LLM API: GigaChat API. Выбор обусловлен доступностью для российских разработчиков, соответствием лицензионным требованиям и подтвержденными показателями в решении математических задач [6].
3. Библиотека для создания бота: python-telegram-bot (версия 20.x), обеспечивающая асинхронное взаимодействие с Telegram API.
4. Локальный рендерер: установленная на сервере полнофункциональная система LaTeX (TeX Live) в сочетании с утилитами pdf2svg и ImageMagick для конвертации полученных PDF-файлов в PNG-изображения.

Принципиальным отличием предлагаемого подхода от существующих является отказ от использования внешних сервисов рендеринга, таких как latex.codecogs.com. Вместо этого модуль создает временные TeX-файлы, компилирует их в PDF посредством локального компилятора pdflatex, затем преобразует полученный PDF в PNG-изображение с заданным разрешением. Все операции выполняются на сервере, где развернут бот, что исключает передачу потенциально чувствительных данных (содержания образовательных запросов) третьим сторонам.

Апробация и результаты эксперимента

Для оценки эффективности разработанного прототипа было проведено экспериментальное тестирование, в ходе которого фиксировались как технические характеристики работы модуля, так и субъективные оценки пользователей.

Тестирование проводилось на виртуальном сервере со следующими характеристиками: 4 vCPU, 8 GB RAM, ОС Ubuntu 22.04. В качестве клиентских устройств использовались смартфоны под управлением Android и iOS. В эксперименте приняли участие 10 добровольцев (студенты и преподаватели математических направлений подготовки).

Техническое тестирование включало отправку боту 100 различных математических запросов, сгенерированных случайным образом. Для каждого запроса фиксировались:

- успешность генерации изображения формулы (отсутствие ошибок компиляции LaTeX);
- время обработки запроса (интервал от отправки сообщения пользователем до получения ответа от бота).

Для оценки пользовательского восприятия участникам предлагалось сравнить ответ GigaChat (с LaTeX-разметкой) и обработанный ответ (с изображениями) по трем критериям по 5-балльной шкале (1 – очень плохо, 5 – отлично):

- понятность (легкость восприятия математического смысла);
- четкость (читаемость символов, отсутствие искажений);
- удобство восприятия (общее визуальное впечатление).

Результаты тестирования производительности представлены в Таблице 1.

Таблица 1. Показатели производительности модуля MathTeXBot

Table 1. Performance indicators of the MathTeXBot module

Показатель	Значение
Доля успешно сгенерированных изображений	96 из 100 (96%)
Среднее время обработки запроса (первичный рендеринг)	2.8 секунды
Среднее время обработки запроса (с использованием кэша)	1.1 секунды
Средний размер генерируемого изображения	35 КБ

Анализ ошибок (4% от общего числа запросов) показал, что они связаны со сложными многострочными конструкциями или использованием специфических пакетов LaTeX, не включенных в стандартную преамбулу. Кэширование позволяет сократить время ответа для повторяющихся формул более чем в 2.5 раза.

Результаты субъективной оценки представлены в Таблице 2.

Таблица 2. Сравнение субъективных оценок читаемости контента (средний балл по 5-балльной шкале)

Table 2. Comparison of subjective assessments of content readability (average score on a 5-point scale)

Критерий	LaTeX-разметка	Обработанный ответ (MathTeXBot)
Понятность	2.1	4.5
Чёткость	1.8	4.9
Удобство восприятия	1.5	4.7

Полученные данные наглядно демонстрируют критическое улучшение восприятия математического контента при использовании разработанного модуля. Низкие оценки «сырого» ответа объясняются нечитаемостью LaTeX-разметки в интерфейсе мессенджера. Высокие оценки обработанного ответа подтверждают, что предложенный подход успешно решает проблему отображения математических символов

Результаты исследования и обсуждение

Основным результатом работы стала разработка архитектуры автономного модуля MathTeXBot, функционирующего по принципу прокси-сервера между API GigaChat и пользователем Telegram и не требующего обращений к внешним сервисам рендеринга.

1. Алгоритм работы модуля

Функционирование модуля включает следующие этапы:

1. Перехват: пользователь направляет запрос боту; бот транслирует его в GigaChat API.
2. Получение необработанного ответа: модуль получает текстовый ответ от языковой модели, содержащий LaTeX-разметку.
3. Парсинг и идентификация: с использованием регулярных выражений модуль сканирует текст на наличие математических блоков.
4. Локальный рендеринг: для каждого идентифицированного блока модуль создает временный TeX-файл, содержащий минимально необходимую преамбулу и выделенное математическое выражение. Файл компилируется с использованием локальной установки pdflatex. Полученный PDF-файл конвертируется в PNG-изображение заданного разрешения с помощью инструментов pdf2png (или конвейера pdf2svg + rsvg-convert). Временные файлы удаляются после завершения обработки.
5. Кэширование: сгенерированные изображения сохраняются в локальном кэше с ключом, представляющим собой хеш-сумму исходного LaTeX-выражения. При повторном появлении того же выражения используется ранее сгенерированное изображение, что позволяет избежать повторной компиляции и сократить время ответа.
6. Сборка ответа: модуль заменяет LaTeX-код в исходном тексте на ссылки на сгенерированные изображения и направляет составное сообщение в Telegram.

7.

2. Организация локального рендеринга

Ключевым компонентом разработанного модуля является подсистема локального рендеринга, реализованная без обращений к внешним API. При обнаружении LaTeX-выражения модуль формирует временный файл следующей структуры:

```
latex
\documentclass[preview]{standalone}
```

```

\usepackage{amsmath}
\usepackage{amssymb}
\begin{document}
{выражение}
\end{document}

```

Использование класса `standalone` с опцией `preview` позволяет получить PDF-файл, содержащий только изображение формулы с минимальными полями, что упрощает последующую обработку. После компиляции посредством вызова `pdflatex` (с параметрами, подавляющими интерактивное взаимодействие) полученный PDF конвертируется в PNG с разрешением 300 dpi, достаточным для четкого отображения на мобильных устройствах.

Для обеспечения совместимости с различными операционными системами модуль использует кроссплатформенные утилиты, доступные в репозиториях основных дистрибутивов Linux, что соответствует требованиям к развертыванию на серверных мощностях российских образовательных учреждений.

3. Сравнение с аналогами и обсуждение ограничений

Предложенное решение занимает свою нишу среди существующих инструментов. В Таблице 3 представлено сравнение ключевых характеристик.

Таблица 3. Сравнительный анализ подходов к рендерингу математических формул

Table 3. Comparative analysis of approaches to rendering mathematical formulas

Характеристика	Веб-платформы (MathJax)	Бот @jh0ker_texbot	Сервис CodeCogs	Предлагаемый модуль MathTeXBot
Среда применения	Веб-браузер	Telegram (ручной ввод)	Любая (через API)	Telegram/MAX (авто/диалог)
Интеграция с LLM	Требует доработки	Отсутствует	Требует доработки	Встроенная (прокси)
Автономность / Технический суверенитет	Частичная (CDN)	Полная	Отсутствует	Полная (локальный)
Зависимость от внешних сервисов	Есть (MathJax CDN)	Нет	Полная	Нет
Контроль данных	Низкий	Полный	Отсутствует	Полный
Непрерывность диалога	Не применимо	Отсутствует	Не применимо	Обеспечена

Анализ показывает, что основное преимущество разработанного модуля — полная автономность в сочетании с автоматической интеграцией в диалоговый процесс. Это выгодно отличает его от @jh0ker_texbot, который требует ручного ввода каждой формулы, и от CodeCogs, создающего внешнюю зависимость.

В ходе разработки и исследования были выявлены ограничения:

- задержки при первичном рендеринге (среднее время 2.8 с) могут быть критичны для очень динамичного диалога;
- ошибки компиляции (4%) возникают при обработке сложных или нестандартных LaTeX-конструкций;
- высокие требования к вычислительным ресурсам сервера (CPU) при пиковых нагрузках.

Также были разработаны пути развития модуля и преодоления ограничений:

- оптимизация процесса компиляции и агрессивное кэширование (уже реализовано и дало сокращение времени до 1.1 с);
- расширение преамбулы подключаемыми пакетами и реализация fallback-механизма с упрощенной обработкой ошибочных выражений;
- горизонтальное масштабирование серверной части и использование очередей задач (например, Celery) для асинхронной обработки.

4. Обеспечение технологического суверенитета

Принципиальным преимуществом разработанного подхода является полная независимость от зарубежных цифровых инфраструктур. В отличие от решений, использующих внешние сервисы рендеринга (CodeCogs, Google Chart API, MathJax CDN), предлагаемый модуль:

1. Не передает содержание образовательных запросов третьим сторонам, что соответствует требованиям законодательства Российской Федерации.
2. Не зависит от доступности зарубежных серверов и не подвержен рискам, связанным с ограничением доступа к иностранным цифровым ресурсам.
3. Может быть развернут на серверном оборудовании, находящемся под контролем образовательной организации, без необходимости обеспечения доступа к внешним сетям.
4. Использует исключительно программное обеспечение с открытым исходным кодом (TeX Live, ImageMagick), что позволяет проводить независимый аудит и модификацию под специфические нужды образовательного процесса.

Заключение

В ходе исследования подтверждена гипотеза о том, что проблема использования больших языковых моделей в математическом образовании обусловлена не семантическими ограничениями моделей, а недостатками пользовательских интерфейсов. Разработанный автономный модуль-препроцессор MathTeXBot для Telegram на базе GigaChat продемонстрировал практическую возможность автоматического преобразования LaTeX-разметки в графические примитивы с использо-

ванием исключительно локальных вычислительных ресурсов, обеспечивая корректную передачу учебного контента в среде с ограниченной поддержкой форматирования.

Предложенная архитектура асинхронного парсинга на основе регулярных выражений и локального рендеринга показала свою работоспособность и соответствует требованиям обеспечения технологического суверенитета отечественных образовательных платформ. Экспериментальная апробация подтвердила высокую эффективность решения: 96% успешно сгенерированных изображений и значительное (с 1.5-2.1 до 4.7-4.9 баллов) улучшение пользовательского восприятия математического контента. Сравнение с аналогами выявило, что разработанный подход — в отличие от пассивных ботов или веб-платформ — интегрируется непосредственно в диалоговый процесс с языковой моделью, обеспечивая непрерывность взаимодействия при сохранении полного контроля над обрабатываемыми данными.

Сравнение с аналогами выявило, что разработанный подход — в отличие от пассивных ботов или веб-платформ — интегрируется непосредственно в диалоговый процесс с языковой моделью, обеспечивая непрерывность взаимодействия при сохранении полного контроля над обрабатываемыми данными.

Практическая значимость работы заключается в возможности использования разработанного прототипа в образовательных Telegram-ботах для учреждений общего и высшего образования, а также в корпоративных системах обучения, функционирующих в закрытых контурах без доступа к сети Интернет. Перспективы дальнейшего развития модуля включают поддержку семантического анализа формул, интеграцию с системами компьютерной алгебры (например, SymPy) для проверки решений, а также улучшение обработки ошибок для обеспечения 100% покрытия генерируемых выражений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Браун, Т. и др. «Модели-трансформеры: архитектура и применение в обработке естественного языка». Журнал искусственного интеллекта, 2023, т. 45, № 3, с. 215-230.
2. Васильев, А.Н. «LaTeX в научной публикации: стандарты и практика». М.: Наука, 2021. 320 с.
3. Кнут, Д.Э. «Всё про TeX». М.: Вильямс, 2018. 560 с.
4. Open AI. «GPT-4 Technical Report». arXiv preprint arXiv:2303.08774, 2023.
5. Сорокин, И.С. «Проблемы токенизации редких символов в нейросетевых языковых моделях». Информационные технологии и вычислительные системы, 2024, № 1, с. 45-58.
6. Проект MathJax. «Документация по рендерингу математики в вебе». [Электронный ресурс] / URL: <https://docs.mathjax.org/> (дата обращения: 01.03.2026).

7. Положиев, Р. И. Повышение качества генерации ответов LLM моделей на математические задачи [Текст]: дис. магистра / Р. И. Положиев. – Санкт-Петербург: НИУ ВШЭ (СПб), 2025. – 85 с.
8. Знаменская, О. В. Математическая типография [Электронный ресурс] : электрон. учеб.-метод. комплекс дисциплины / О. В. Знаменская, С. В. Знаменский, В. П. Кривоколеско [и др.]. – Красноярск: СФУ, 2008. – URL: <http://elib.sfu-kras.ru/handle/2311/138781> (дата обращения: 14.03.2026).
9. Анализ возможностей нейросетевых моделей при решении некоторых математических задач [Текст] // Информатизация образования и методика электронного обучения: цифровые технологии в образовании: материалы VI Междунар. науч. конф. (Красноярск, 25–28 сентября 2024 г.) : в 3 ч. / под общ. ред. М. В. Носкова. – Красноярск: СФУ, 2025. – Ч. 2. – С. 115-120.
10. Mamedov, V. GigaChat Family: Efficient Russian Language Modeling Through Mixture of Experts Architecture [Текст]. V. Mamedov, E. Kosarev, G. Leleytner [и др.]. Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations). – 2025. – С. 1-10.

REFERENCES

1. Mamedov, V., Kosarev, E., Leleytner, G. et al. GigaChat Family: Efficient Russian Language Modeling Through Mixture of Experts Architecture. Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations). – 2025. – P. 1-10.
2. Analiz vozmozhnostey neyrosetevykh modeley pri reshenii nekotorykh matematicheskikh zadach [Analysis of the Capabilities of Neural Network Models in Solving Some Mathematical Problems] Informatizatsiya obrazovaniya i metodika elektronnoy obucheniya: tsifrovyye tekhnologii v obrazovanii: materialy VI Mezhdunarodnoy nauchnoy konferentsii (Krasnoyarsk, 25–28 sentyabrya 2024 g.): v 3 ch. pod obshch. red. M. V. Noskova. – Krasnoyarsk: SFU, 2025. – Ch. 2. – S. 115-120. (in Russian)
3. Vasil'ev, A. N. LaTeX v nauchnoy publikatsii: standarty i praktika [LaTeX in Scientific Publication: Standards and Practice]. – Moscow: Nauka, 2021. – 320 p. (in Russian)
4. Znamenskaya, O. V., Znamenskiy, S. V., Krivokolesko, V. P. et al. Matematicheskaya tipografiya [Mathematical Typography]: elektronnyy uchebno-metodicheskiy kompleks distsipliny. – Krasnoyarsk: SFU, 2008. – URL: <http://elib.sfu-kras.ru/handle/2311/138781> (data obrashcheniya: 14.03.2026). (in Russian)
5. Sorokin, I. S. Problemy tokenizatsii redkikh simvolov v neyrosetevykh yazykovykh modelyakh [Problems of Tokenization of Rare Symbols in Neural Network Language Models] Informatsionnye tekhnologii i vychislitel'nye sistemy. – 2024. – № 1. – S. 45-58. (in Russian)

6. Polozhiev, R. I. Povyshenie kachestva generatsii otvetov LLM modeley na matematicheskie zadachi [Improving the Quality of LLM Models' Responses to Mathematical Problems]: dis. magistra. – Saint Petersburg: NIU VShE (SPb), 2025. – 85 p. (in Russian)
7. Repozitariy bota @jh0ker_texbot [Repository of the @jh0ker_texbot]. – URL: <https://github.com/jh0ker/texbot> (data obrashcheniya: 10.03.2026). (in Russian)
8. Proekt MathJax. Dokumentatsiya po renderingu matematiki v vebe [MathJax Project. Documentation on Rendering Mathematics on the Web]. – URL: <https://docs.mathjax.org/> (data obrashcheniya: 01.03.2026). (in Russian)
9. OpenAI. GPT-4 Technical Report. – arXiv:2303.08774, 2023. – URL: <https://arxiv.org/abs/2303.08774> (data obrashcheniya: 05.03.2026).
10. University of Reading. Guide to using MathJax in Blackboard. – URL: <https://www.reading.ac.uk/> (data obrashcheniya: 12.03.2026).

Информация об авторе: Липатов Владимир Алексеевич – аспирант ФГБОУ ВО «Алтайский государственный педагогический университет»
г. Барнаул, Россия
vladimir.lipatov1998@gmail.com

Автор прочитал и одобрил окончательный вариант рукописи.

Information about the author:

Lipatov Vladimir Alekseevich – graduate student
Federal State Budgetary Educational Institution of Higher Education "Altai State Pedagogical University"
Barnaul, Russia
vladimir.lipatov1998@gmail.com

The author read and approved the final version of the manuscript.

Статья поступила в редакцию / The article was submitted: 03.02.2026

Одобрена после рецензирования и доработки / Approved after reviewing and revision: 20.02.2026

Принята к публикации / Accepted for publication: 27.02.2026

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.