УДК 004.048

# ПРИМЕНЕНИЕ КВАНТОВАННЫХ АЛГОРИТМОВ ДЛЯ АДАПТАЦИИ ЯЗЫКОВЫХ МОДЕЛЕЙ В ЗАДАЧЕ ВЕРИФИКАЦИИ ХОДА РЕШЕНИЯ КВАДРАТНЫХ УРАВНЕНИЙ

**А. Н. Хайбуллин**[1] [0009-0001-7321-956X], **Д. Н. Тумаков**[2] [0000-0003-0564-8335]

[1, 2]*Казанский (Приволжский) федеральный университет, г. Казань, Россия*

[1]almaz.khaybullin@mail.ru, [2]dtumakov@kpfu.ru

***Аннотация***

Работа посвящена исследованию квантованных подходов к адаптации языковых моделей для задачи автоматической пошаговой проверки корректности хода решения квадратных уравнений. Рассмотрена результативность подходов параметрически эффективного дообучения (PEFT) при адаптации языковых моделей DeepSeek-R1-Distill-Qwen-1.5B и InternLM2-Math-Plus-1.8B для создания математического верификатора (Process-supervised Reward Models, PRM). Эксперименты проведены на синтетическом наборе данных квадратных уравнений, дополненном негативным сэмплированием для имитации ошибок обучающихся. Выполнено сравнительное тестирование стандартных (LoRA, DoRA, rsLoRA) и квантованных (QLoRA, QDoRA, LoftQ) алгоритмов тонкой настройки. Дополнительно изучена обобщающая способность нейросетей (Zero-shot Transfer) на структурно отличающемся наборе линейных уравнений. Результаты показали, что квантование решает проблемы численной стабильности вычислений для нестандартных архитектур (InternLM2), обеспечивая при этом качество, сопоставимое со стандартными методами. Для модели DeepSeek-R1 метод QLoRA достиг точности (Accuracy) 97.77%, а методы QDoRA и LoftQ – по 98%, что лишь незначительно уступает классическому алгоритму LoRA (98.67%). Аналогично для нестандартной архитектуры InternLM2 применение QLoRA позволило достичь точности 92.67% (против 93% у базового LoRA). Однако алгоритмы без понижения разрядности весов (LoRA) склонны сохранять более богатое представление выученных паттернов, обеспечивая хорошую способность к переносу

знаний для моделей класса Reasoning (Accuracy DeepSeek-R1 66.8% против 61.4% у QLoRA на новых данных).

***Ключевые слова**: языковые модели, параметрически эффективное дообучение, квантованные методы обучения, математическое рассуждение, автоматизированная проверка решений, модели вознаграждения с контролем за процессом.*

**ВВЕДЕНИЕ**

Развитие больших языковых моделей (Large Language Model, LLM) открыло новые перспективы в области математики [1], в частности, для создания интеллектуальных систем пошаговой проверки решений алгебраических задач. Однако современные архитектуры часто подвержены логическим ошибкам и «галлюцинациям» [2, 3] при выполнении точных математических вычислений. Эта проблема существенно усугубляется, когда перед системой ставится задача не просто сгенерировать конечный ответ, а выступить в роли верификатора – оценить корректность каждого промежуточного алгебраического преобразования, выполненного обучающимся (подход Process-supervised Reward Models, PRM) [4].

Адаптация базовых моделей нейронных сетей под узкоспециализированные задачи с использованием классического полного дообучения (Supervised Fine-Tuning, SFT) требует значительных вычислительных мощностей, что делает процесс труднореализуемым на потребительском оборудовании. В связи с этим актуальность приобретает исследование методов параметрически эффективного дообучения (Parameter-Efficient Fine-Tuning, PEFT). Современные алгоритмы, такие как LoRA, DoRA, а также их модификации, использующие квантование (QLoRA, QDoRA) и усовершенствованные механизмы инициализации и стабилизации ранга (LoftQ, rsLoRA) [5-8], позволяют снизить потребление видеопамяти. При этом научный интерес представляет изучение влияния квантованного шума на обобщающую способность моделей в задачах проверки решений математических задач.

Целью настоящей работы был сравнительный анализ эффективности адаптации языковых моделей, таких как DeepSeek-R1-Distill-Qwen-1.5B и InternLM2-Math-Plus-1.8B, для задачи проверки корректности промежуточных шагов при

решении квадратных уравнений. Для достижения поставленной цели в рамках исследования выполнены оценки:

- эффективности различных методов PEFT (LoRA, QLoRA, DoRA, QDoRA, LoftQ, rsLoRA) при дообучении моделей на синтетическом датасете квадратных уравнений с использованием негативного сэмплирования (включения примеров с намеренно сгенерированными алгоритмическими ошибками для обучения верификатора);
- обобщающей способности дообученных моделей путем тестирования их навыков на структурно отличной задаче – проверке алгоритмических шагов при решении уравнений меньшего порядка (линейных уравнений).

## 1. МЕТОДОЛОГИЯ ИССЛЕДОВАНИЯ И ДАННЫЕ

На сегодняшний день необходимо использование алгоритмически сгенерированных данных, что продиктовано современным кризисом обучающих выборок. Распространение языковых моделей приводит к оттоку пользователей с образовательных и Q&A-платформ [9, 10], а вынужденное рекурсивное обучение на неконтролируемом ИИ-контенте вызывает деградацию способностей нейросетей [11].

Для решения рассмотренной проблемы написана программа на языке Python, генерирующая синтетический датасет, который использован при обучении и валидации для задачи верификации квадратных уравнений.

Объем созданной обучающей выборки составил 24000 примеров, а тестовой – 3000 примеров. Генерация исходных заданий произведена по нескольким структурным шаблонам для обеспечения высокой вариативности данных и предотвращения переобучения моделей на одном синтаксическом формате [12]. В датасете представлены следующие типы квадратных уравнений:

- стандартная форма (например, $2x^2 – 2x – 60 = 0$);
- смещенная форма с переменными в разных частях уравнения (например, $–7x^2 = –20x$);
- уравнения с многочленами на обеих сторонах (например, $x^2 + 4x + 7 = –2x + 2$);
- уравнения с дробными выражениями (например, $(6x^2 + 17x + 7) / 4 = 0$);

- конструкции с полным квадратом и разностью квадратов (например, $(x + 3)^2 = 16$ или $4x^2 – 25 = 0$).

Датасет адаптирован для обучения моделей-верификаторов (Process-supervised Reward Model). Для каждого сгенерированного уравнения выстроена цепочка решения: приведение к стандартному виду, вычисление дискриминанта, извлечение корня из дискриминанта и нахождение финальных корней уравнения. Для обучения модели, отличающей математически корректные действия от ошибочных, применен метод негативного сэмплирования (Negative Sampling). Баланс классов выдержан в строгой пропорции: 50% верных шагов (correct) и 50% ошибочных (incorrect). Ошибки сгенерированы в соответствии с типичными неточностями в решениях задач обучающимися:

- алгебраические ошибки: искажение знаков в формуле дискриминанта (например, использование $D = b^2 + 4ac$ вместо $D = b^2 – 4ac$);
- синтаксические ошибки переноса: потеря знака минус перед коэффициентом $b$ в итоговой формуле корней или приравнивание числителя дроби к единице вместо нуля при избавлении от знаменателя.

Формат данных представлен в виде структуры JSONL, где каждый пример выражен изолированным срезом решения, содержащим исходное условие задачи, массив математически верных предыдущих шагов, текущий проверяемый шаг с потенциально внедренной ошибкой и целевую метку класса.

Для проверки способности моделей к переносу знаний использован дополнительный тестовый набор данных, состоящий из 5000 линейных уравнений. Эти уравнения не использованы при обучении. Целью такого эксперимента была проверка способности моделей, обученных верифицировать квадратные уравнения, применять выученные логические паттерны проверки (арифметическая корректность, правила переноса слагаемых) к более простым, но структурно отличным задачам, которые отсутствовали в процессе Fine-Tuning.

## 2. ПАРАМЕТРЫ И МЕТОДЫ ОБУЧЕНИЯ

Были использованы две модели со схожим количеством параметров (~1.6–1.8 млрд), но с различным предварительным обучением:

1) DeepSeek-R1-Distill-Qwen-1.5B – модель на базе архитектуры Qwen-2.5, прошедшая дистилляцию знаний (перенос знаний с помощью генерации размеченных данных от более крупной модели-учителя) от модели DeepSeek, обладающей расширенными способностями к логическому выводу (Reasoning) и генерации цепочек рассуждений (Chain-of-Thought), что, согласно современным исследованиям [13], является важным фактором для успешного решения математических задач;

2) InternLM2-Math-Plus-1.8B – специализированная математическая модель, обученная на корпусе научных (математических) текстов.

**Метод LoRA**

В качестве базового подхода к адаптации моделей рассмотрен метод LoRA (Low-Rank Adaptation). Его принцип основан на фиксации (заморозке) весов предварительно обученной сети и интеграции в архитектуру Transformer дополнительных обучаемых матриц низкого ранга. Вместо обновления полной матрицы весов *W* при дообучении оптимизируется ее приращение $\Delta W$, которое может быть представлено в виде произведения двух матриц *B* и *A* меньшей размерности.

Обновленные веса $W'$ в LoRA имеет следующий вид:

$$W' = W_0 + \Delta W = W_0 + B \cdot A,$$

где $W_0$ – замороженные веса, $B$ и $A$ – обучаемые матрицы ранга $r \ll d$ (ранга исходной матрицы $W_0$).

**Метод QLoRA**

Метод QLoRA (Quantized Low-Rank Adaptation) позволяет заморозить основные веса модели, квантованные до 4 бит, и обучать лишь дополнительные матрицы низкого ранга (адаптеры).

Параметры обучения были унифицированы для обеспечения корректности сравнения. Базовые веса исходной модели квантованы в формат 4-bit NF4. Для оптимизации процесса градиентного спуска использован алгоритм paged_adamw_8bit. Скорость обучения выбрана равной $2 \cdot 10^{-4}$.

Использование 4-битного квантования (NF4) позволило сократить потребление видеопамяти для весов модели почти в 4 раза по сравнению с 16-битным

представлением (FP16). При этом точность модели остается практически неизменной по сравнению с полным дообучением [14]. Однако при дальнейшем уменьшении разрядности весов (до 3 бит и менее) происходит значительное снижение точности при обучении модели [15].

Гиперпараметры метода LoRA заданы следующим образом: параметр масштабирования ($\alpha$) зафиксирован на значении 16 для обеих моделей, а ранг матриц ($r$) варьируется от 8 до 16. Дообучение применено к целевым линейным модулям: для DeepSeek это – слои $q_{\text{proj}}$, $k_{\text{proj}}$, $v_{\text{proj}}$, $o_{\text{proj}}$, а также слои FFN ($gate_{\text{proj}}$, $up_{\text{proj}}$, $down_{\text{proj}}$); для InternLM – аналогичные слои $w_{\text{qkv}}$, $w_0$, $w_1$, $w_2$, $w_3$.

**Метод DoRA**

Метод DoRA (Weight-Decomposed Low-Rank Adaptation) – это развитие классического подхода LoRA, направленное на устранение его фундаментальных ограничений. При использовании стандартного метода LoRA изменения амплитуды и направления весов сильно коррелируют, что отличает динамику обновления весов от полного дообучения (Full Fine-Tuning). Для решения этой проблемы метод DoRA разделяет исходную матрицу весов $W$ предварительно обученной модели на два независимых компонента:

- амплитуду $m$ – обучаемый вектор, отвечающий за масштаб (силу) активации признаков;
- направление $V$ – матрицу пространственной ориентации весов (инициализируется базовыми весами $W_0$), которая в процессе дообучения обновляется за счет низкорангового приращения $B \cdot A$.

Математически исходная матрица весов представлена как произведение амплитуды на нормализованную матрицу направления. При дообучении базовые веса $W_0$ заморожены, а обновление направления обусловлено прибавлением обучаемых матриц $B$ и $A$ низкого ранга, аналогично классическому методу LoRA. Итоговое обновление весов в методе DoRA имеет следующий вид:

$$W' = m\frac{(W_0 + B \cdot A)}{\|W_0 + B \cdot A\|_c},$$

где $m$ – обучаемый вектор амплитуды, $W_0$ – замороженная базовая матрица весов, $B$ и $A$ – обучаемые матрицы низкого ранга, а $\|\cdot\|_c$ – операция векторной нормализации по столбцам. Такая декомпозиция позволяет модели независимо

корректировать силу активации логических паттернов (через параметр $m$) и их семантическое направление (через матрицы $B$ и $A$). Благодаря этому паттерн обновления весов в DoRA практически идентичен полному дообучению, обеспечивая более высокую точность на сложных аналитических задачах, но требуя при этом существенно меньше обучаемых параметров.

Как показано в [16], классический LoRA склонен изменять амплитуду и направление весов пропорционально, что ограничивает его способность к тонкой настройке. Декомпозиция в DoRA позволяет модели изолированно и более интенсивно корректировать направление логических связей (матрицы $B$ и $A$), не разрушая базовые знания модели о математическом синтаксисе.

**Метод QDoRA**

Для эффективной адаптации базовых языковых моделей к задаче пошаговой математической верификации в условиях ограниченных вычислительных ресурсов также был применен гибридный метод QDoRA. В таком подходе объединены преимущества 4-битного квантования из метода QLoRA [6] для снижения потребления видеопамяти и описанные выше архитектурные улучшения метода DoRA, направленные на декомпозицию весов.

Базовая матрица весов $W_0$ в данном методе остается замороженной и квантованной в 4-битный формат, а обучению подлежат только вектор амплитуды $m$ и матрицы низкого ранга $B$ и $A$.

**Метод LoftQ**

В качестве третьего метода параметрически эффективного дообучения был рассмотрен алгоритм LoftQ (LoRA-Fine-Tuning-Aware Quantization). Этот метод направлен на устранение ключевого недостатка стандартного подхода QLoRA: при квантовании весов базовой модели неизбежно возникновение ошибки представления (quantization discrepancy) – различия между исходными высокоточными весами и их квантованным аналогом.

В стандартном QLoRA матрица $A$ инициализируется случайными значениями, а матрица $B$ – нулями. Такой подход изначально игнорирует ошибку квантования, что может замедлить сходимость на задачах, требующих точного логического вывода.

LoftQ решает указанную проблему путем совместной оптимизации процессов квантования и инициализации с использованием сингулярного разложения (Singular Value Decomposition, SVD). Метод находит такие начальные значения матриц $B$ и $A$, при которых сумма квантованных весов и их произведения аппроксимируют исходную матрицу на этапе инициализации:

$$W \approx W_q + B \cdot A.$$

Математически задача инициализации LoftQ может быть представлена в виде

$$\min_{W_q,B,A} \left\| W - (W_q + B \cdot A) \right\|_F ,$$

где $\|\cdot\|_F$ – норма Фробениуса, $W_q$ – квантованные веса в 4-битном формате, $B \in R^{d\times r}$ и $A \in R^{r\times k}$ – обучаемые матрицы ранга $r$. При таком подходе компенсация ошибки квантования предусмотрена уже в точке инициализации.

Применение в ходе экспериментов метода LoftQ к модели DeepSeek-R1-Distill (архитектура Qwen) обеспечило стабильную сходимость функции потерь (0.84 на первом шаге против 0.99 при стандартном QLoRA) и подтвердило эффективность SVD-инициализации.

Попытка применить LoftQ к модели InternLM2-Math-Plus выявила ограничение метода. Несмотря на то что SVD-разложение весов выполнялось успешно, при обратном распространении ошибки (backward pass) значение функции потерь становилось равным NaN. Программная отладка подтвердила, что прямой проход (forward pass) выполнялся корректно, однако градиенты не вычислялись для нестандартных слоев внимания InternLM2.

Таким образом, в отличие от метода QLoRA, который квантует веса напрямую через стандартный механизм bitsandbytes и совместим с широким спектром архитектур, метод LoftQ предъявил повышенные требования к стандартизации архитектуры целевой модели. В связи с этим в рамках настоящей работы метод LoftQ был применен исключительно к модели DeepSeek-R1-Distill.

**Метод rsLoRA**

Для преодоления ограничений классических алгоритмов дообучения при использовании высоких значений $r$ ранга матриц дополнительно был исследован метод rsLoRA (Rank-Stabilized Low-Rank Adaptation). Известно, что в стандартном методе LoRA приращение матриц масштабируемо на коэффициент $\alpha/r$. Увеличение ранга для представления более сложных логических паттернов приводило к непропорциональному уменьшению величины приращения весов, что требовало перенастройки гиперпараметров (в частности скорости обучения) и создавало риск расходимости алгоритма.

Ключевым отличием метода rsLoRA является замена линейного масштабирования на деление на корень из ранга:

$$\Delta W = \frac{\alpha}{\sqrt{r}} B \cdot A.$$

Такая модификация стабилизирует дисперсию активаций при прямом проходе и нормализует градиенты при обратном распространении ошибки. Теоретически это позволяет безопасно увеличивать ранг обучаемых матриц без риска коллапса обучения. Однако это часто ведет к нестабильности, для устранения которой и был применен метод rsLoRA.

## 3. ОЦЕНКА ТОЧНОСТИ КВАНТОВАННЫХ МОДЕЛЕЙ

Задача автоматической проверки решений сформулирована как бинарная классификация. Положительный класс назначается в случае, если проверяемый математический шаг являлся алгебраически верным. Отрицательный класс присваивается, если в текущем шаге вычислений допущена ошибка. Вычисление и интерпретация метрик произведены относительно положительного класса:

- Precision (точность) – доля действительно верных шагов среди всех элементов, классифицированных как корректные. Высокий показатель точности означает низкий процент ложноположительных срабатываний, т. е. ситуаций, когда система ошибочно принимает неверный шаг в решении за правильный;
- Recall (полнота) – доля корректно идентифицированных верных шагов от их общего фактического числа в тестовой выборке. Эта метрика показывает спо-

собность модели не упускать правильные решения, минимизируя ложноотрицательные срабатывания (ситуации, когда верный шаг классифицируется как неверный);

- Accuracy (общая точность) – доля правильных ответов модели по обоим классам, демонстрируя способность системы к оценке как верных, так и ошибочных действий.

### 3.1. Результаты верификации хода решений квадратных уравнений

Процесс оценки пользовательского решения проведен итеративно для каждого отдельного шага, а не для всей последовательности вычислений целиком. Различное количество шагов в решении (например, подробное решение за семь или краткое решение за пять итераций) не влияло на процесс проверки, поскольку система анализировала логическую и алгебраическую связности каждого шага с предыдущими. Метрики для всех рассмотренных моделей приведены в табл. 1. В ней символ «–» означает, что данный метод дообучения не совместим с моделью InternLM2-Math-Plus-1.8B в силу специфики ее архитектуры.

Дообучение моделей проведено на одной видеокарте NVIDIA T4 (16 GB VRAM) в течение трех эпох. В качестве функции потерь использована перекрестная энтропия (Cross-Entropy) с маскированием (Masked Loss), где градиенты рассчитывались исключительно на токенах ответа (correct/incorrect) [17].

Перед дообучением (в режиме Zero-shot) обе модели продемонстрировали неудовлетворительные результаты (менее 70%). Это объясняется отсутствием у базовых моделей понимания специфического формата проверки решения математических задач [18]. После дообучения результаты существенно улучшились (около 98%).

Табл. 1. Сравнительные результаты моделей для различных методов дообучения.

| Модель (метод) | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| DeepSeek-R1-Distill-Qwen-1.5B (Zero-shot) | 51.4% | 0.490 | 0.570 | 0.510 |
| InternLM2-Math-Plus-1.8B (Zero-shot) | 64% | 0.588 | 0.833 | 0.649 |

| | | | | |
|---|---|---|---|---|
| DeepSeek-R1-Distill-Qwen-1.5B (QLoRA) | 97.77% | 0.980 | 0.979 | 0.976 |
| DeepSeek-R1-Distill-Qwen-1.5B (LoRA) | 98.67% | 0.974 | 0.997 | 0.987 |
| DeepSeek-R1-Distill-Qwen-1.5B (LoftQ) | 98% | 0.969 | 0.994 | 0.981 |
| DeepSeek-R1-Distill-Qwen-1.5B (DoRA) | 97% | 0.963 | 0.981 | 0.972 |
| DeepSeek-R1-Distill-Qwen-1.5B (QDoRA) | 98% | 0.981 | 0.989 | 0.990 |
| InternLM2-Math-Plus-1.8B (QDoRA) | – | – | – | – |
| InternLM2-Math-Plus-1.8B (LoRA) | 93% | 0.936 | 0.921 | 0.924 |
| InternLM2-Math-Plus-1.8B (LoftQ) | – | – | – | – |
| InternLM2-Math-Plus-1.8B (rsLoRA) | 91.7% | 0.906 | 0.938 | 0.923 |
| InternLM2-Math-Plus-1.8B (QLoRA) | 92.67% | 0.923 | 0.917 | 0.921 |

Среди всех примененных методов дообучения для модели DeepSeek-R1-Distill-Qwen-1.5B наиболее высокие результаты показали классический LoRA и гибридный метод QDoRA. Так, LoRA продемонстрировал максимальную точность (98.67%) и наивысший Recall (0.997), что подтвердило эффективность этого метода в минимизации ложноотрицательных срабатываний. Метод QLoRA, имеющий 4-битное квантование базовой модели с последующей адаптацией, показал незначительно более низкую точность (97.77%) по сравнению с полновесной LoRA, что стало оправданным компромиссом за экономию видеопамяти. Наибольшее значение F1-Score (0.99) достигнуто при использовании метода QDoRA. Данный подход позволил достичь хорошего баланса между полнотой и точностью.

Для специализированной архитектуры модели InternLM2-Math-Plus-1.8B наилучшие значения метрик также продемонстрировал метод LoRA (Accuracy = 93%, F1-Score = 0.924), в то время как квантованный алгоритм QLoRA показал сопоставимые, немного худшие результаты с точностью, равной 92.67%. Метод rsLoRA несколько уступил базовым подходам по общей точности (91.7%), однако обеспечил максимальную для данной модели полноту, равную 0.938.

Динамика функций ошибки для различных методов при дообучении модели DeepSeek-R1-Distill-Qwen-1.5B представлена на рис. 1. Все методы демонстрируют близкую динамику сходимости: резкое снижение функции потерь в первые 300–500 шагов с последующим плавным убыванием до стабилизации значений на уровне 0.060–0.067. Здесь шагом считается один проход модели при обучении на одном батче. Весь датасет разбит на 1500 батчей, размером 16 примеров. Таким образом, все обучение проходит за три эпохи (4500 шагов).

Можно заметить, что методы LoRA и QLoRA сходятся медленнее, но в конце обучения их функции ошибки принимают немного худшие значения, чем для остальных методов. Однако, исходя из результатов табл. 1, видим, что метод LoRA дает самую хорошую точность.

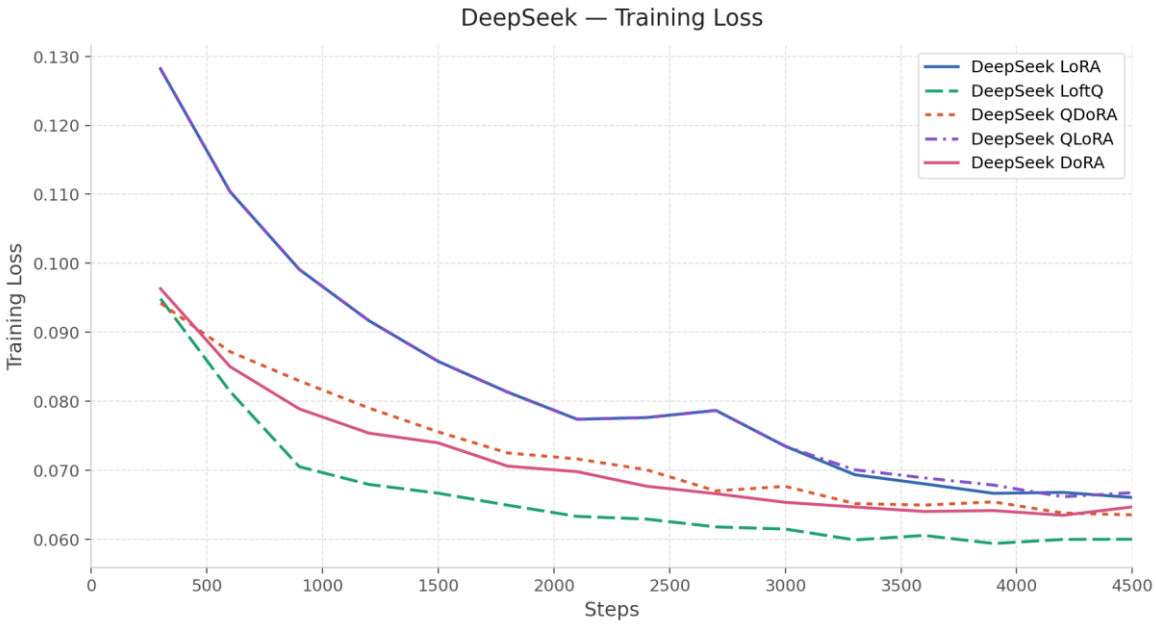


Рис. 1. Динамика функции потерь для модели DeepSeek-R1-Distill-Qwen-1.5B на каждом шаге обучения.

На рис. 2 представлены зависимости функции потерь при дообучении модели InternLM2-Math-Plus-1.8B. Отметим, что для этой модели методы DoRA, QDoRA и LoftQ выдают ошибки при обучении.

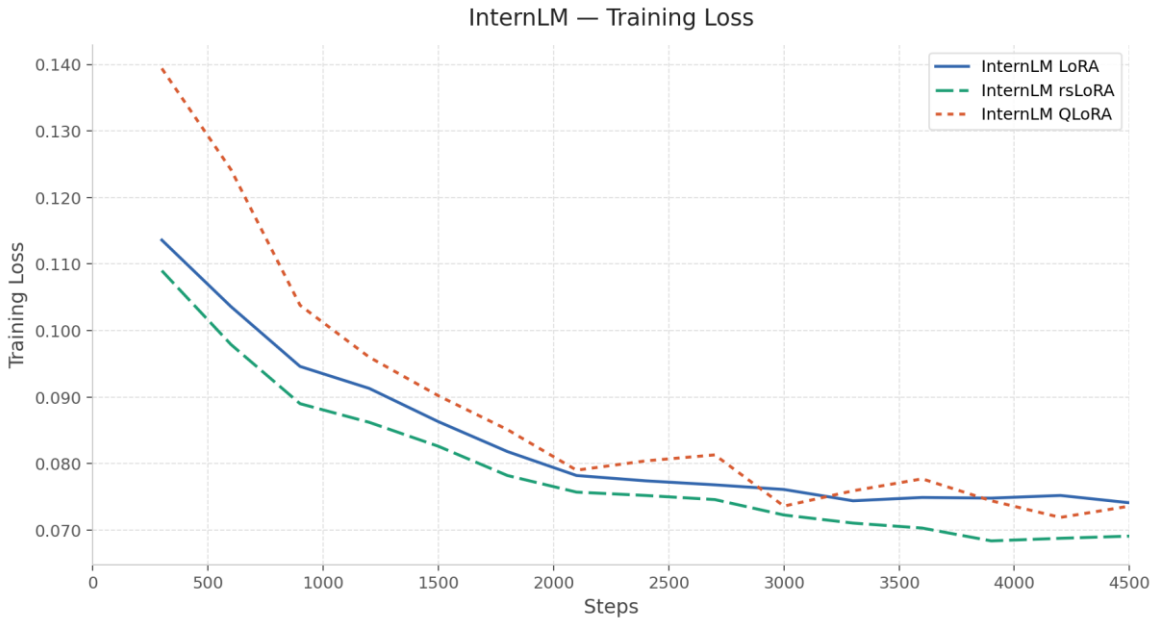


Рис. 2. Динамика функции потерь для модели InternLM2-Math-Plus-1.8B на каждом шаге обучения.

Как и для модели от DeepSeek, в этом случае все методы обучения достигают к концу третьей эпохи примерно одинаковые значения для функции ошибки в интервале от 0.70 до 0.75.

В ходе экспериментов метод rsLoRA продемонстрировал лучшую динамику функции потерь, частично компенсируя склонность архитектуры InternLM2 к численным нестабильностям, описанным ранее. Тем не менее по итоговым метрикам на задаче верификации квадратных уравнений метод rsLoRA показал точность 91.7%, незначительно уступив базовому QLoRA (92.67%) и стандартному LoRA (93%).

Важным моментом обучения также является и его время. В табл. 2 приведено время дообучения для каждой модели.

Табл. 2. Время дообучения моделей.

| Модель | Метод | Время |
|---|---|---|
| DeepSeek-R1-Distill-Qwen-1.5B | LoftQ | 4 ч 15 мин |
| | LoRA | 15 ч 44 мин |
| | QDoRA | 9 ч 51 мин |
| | QLoRA | 4 ч 28 мин |
| | DoRA | 50 ч 21 мин |
| InternLM2-Math-Plus-1.8B | LoRA | 4 ч 13 мин |
| | RsLoRA | 4 ч 21 мин |
| | QLoRA | 6 ч 26 мин |

Результаты, представленные в табл. 2, показывают существенные различия во времени дообучения в зависимости как от выбранного метода, так и от архитектуры модели.

Выявлена заметная аномалия во времени дообучения методом DoRA для модели DeepSeek-R1-Distill-Qwen-1.5B – 50 ч 21 мин, что в 3.2 раза превышает время стандартного LoRA (15 ч 44 мин) и более чем в 11 раз – время квантованного QLoRA (4 ч 28 мин). Это объясняется тем, что DoRA декомпозирует матрицы весов на компоненты величины и направления, что существенно увеличивает вычислительную нагрузку для данной архитектуры. Применение квантования в QDoRA позволило сократить время до 9 ч 51 мин, т. е. почти в 5 раз по сравнению с базовым DoRA.

Квантованные методы (QLoRA, LoftQ) для DeepSeek-R1 также значительно выиграли по скорости дообучения относительно стандартного LoRA: время сократилось примерно в 3.5 раза – до 4–4.5 ч. Это демонстрирует практическое преимущество квантования с точки зрения объема видеопамяти и временных затрат на обучение.

Для модели InternLM2-Math-Plus-1.8B все протестированные методы (LoRA, RsLoRA, QLoRA) показали сопоставимое время дообучения в диапазоне 4–6.5 ч, что свидетельствует о меньшей чувствительности этой архитектуры к выбору метода PEFT с точки зрения вычислительных затрат. Стандартный LoRA для

InternLM2 (4 ч 13 мин) оказался почти в 4 раза быстрее, чем LoRA для DeepSeek-R1 (15 ч 44 мин).

### 3.2. Результаты обобщающей способности моделей: верификация хода решения линейных уравнений

Второй этап тестирования был проведен на линейных уравнениях. Важно отметить, что модели не были обучены на этом типе задач. Результаты представлены в табл. 3. Здесь «–» означает что данный метод дообучения не совместим с моделью InternLM2-Math-Plus-1.8B в силу специфики ее архитектуры.

По результатам эксперимента специализированная математическая модель InternLM2 показала значительное снижение качества проверки решения математических задач при смене типа уравнений. В то же время модель DeepSeek-R1 продемонстрировала более высокую стабильность результатов.

При тестировании обобщающей способности на линейных уравнениях модель InternLM2, дообученная с помощью rsLoRA, показала результат 57.8%, что сопоставимо с базовым LoRA (58%). Характерной особенностью InternLM2 во всех методах дообучения стал высокий Recall (0.87–0.93) при относительно низком Precision (0.57–0.61), что свидетельствует о систематическом смещении модели в сторону одобрения шагов решения – она предпочитает классифицировать шаги как корректные, что приводит к большому числу ложноположительных срабатываний.

Табл. 3. Обобщающая способность моделей.

| Модель (Метод) | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| DeepSeek-R1-Distill-Qwen-1.5B (zero-shot) | 52% | 0.49 | 0.57 | 0.51 |
| DeepSeek-R1-Distill-Qwen-1.5B (QLoRA) | 61.4% | 0.8 | 0.43 | 0.624 |
| DeepSeek-R1-Distill-Qwen-1.5B (QDoRA) | 65% | 0.8 | 0.52 | 0.64 |
| DeepSeek-R1-Distill-Qwen-1.5B (LoftQ) | 61.6% | 0.76 | 0.62 | 0.62 |

| | | | | |
|---|---|---|---|---|
| DeepSeek-R1-Distill-Qwen-1.5B (LoRA) | 66.8% | 0.76 | 0.61 | 0.67 |
| DeepSeek-R1-Distill-Qwen-1.5B (DoRA) | 64.2% | 0.72 | 0.61 | 0.66 |
| InternLM2-Math-Plus-1.8B (zero-shot) | 53.2% | 0.55 | 0.79 | 0.53 |
| InternLM2-Math-Plus-1.8B (QLoRA) | 59.7% | 0.57 | 0.93 | 0.61 |
| InternLM2-Math-Plus-1.8B (QDoRA) | – | – | – | – |
| InternLM2-Math-Plus-1.8B (LoftQ) | – | – | – | – |
| InternLM2-Math-Plus-1.8B (LoRA) | 58% | 0.61 | 0.87 | 0.57 |
| InternLM2-Math-Plus-1.8B (rsLoRA) | 57.8% | 0.58 | 0.92 | 0.57 |

Из таблицы 3 видно, что базовая модель DeepSeek-R1-Distill-Qwen-1.5B не способна выполнять задачу (Accuracy = 52%), в то время как дообученная QDoRA модель продемонстрировала значительный прирост (Accuracy = 65.0%). Модель успешно перенесла выученные паттерны арифметической верификации на новый тип уравнений. Снижение метрики Recall вызвано тем, что структура линейных уравнений отличается от обучающей выборки, что приводило к «гиперкоррекции» [19] – модель излишне строго оценивала незнакомые ей правильные синтаксические конструкции.

Архитектура DeepSeek-R1-Distill-Qwen-1.5B продемонстрировала более высокую устойчивость к смене типа задач. Наилучшую способность к переносу знаний на линейные уравнения показал метод LoRA (Accuracy 66.8%). Гибридные методы с квантованием также обеспечили значительный отрыв от базовой неадаптированной модели (52%): конфигурация QDoRA достигла точности 65.0%, а метод DoRA – 64.2%. Это свидетельствует о том, что архитектуры класса Reasoning, оптимизированные под длинные цепочки рассуждений, способны

обобщать правила верификации на новые типы задач значительно эффективнее, чем узкоспециализированные математические модели.

Для понимания механизмов обобщающей способности был проведен анализ предсказаний базовой и дообученной (LoRA) моделей (DeepSeek-R1-Distill-Qwen-1.5B) на наборе линейных уравнений. Были сделаны следующие выводы.

1. Анализ результатов показал, что благодаря параметрически эффективной адаптации дообученная модель успешно освоила базовые правила алгебраических преобразований:

- устранение ложноотрицательных срабатываний. Базовая модель систематически браковала корректные переносы слагаемых со сменой знака (например, шаг $-9x = 29 - (-14)$). Дообученная нейросеть научилась безошибочно верифицировать такие действия;
- выявление скрытых вычислительных ошибок. Дообучение повысило чувствительность модели к базовой арифметике. Так, при ошибочном вычислении шага $-15x = 26$ (вместо 25) в уравнении $-19 - 15x = 6$ базовая модель пропускала ошибку, тогда как дообученная архитектура ее обнаружила.

2. Несмотря на общий рост точности до 66.8%, дообученная модель продемонстрировала два характерных типа ошибок на заданиях, которых не было в обучающей выборке квадратных уравнений:

- пропуск сложных арифметических ошибок (False Positives). Модель находила структурные ошибки, но все еще пропускала неточности при сложении отрицательных чисел (например, одобряла неверный шаг: $-43 - 10 = -50$). Это объясняется тем, что негативное сэмплирование при обучении фокусировалось на формулах дискриминанта, из-за чего нейросеть не приобрела абсолютной чувствительности к арифметике;
- отторжение незнакомых форматов (False Negatives). Алгоритм решения линейного уравнения сводился к изоляции переменной в виде простой дроби (например, $x = 5/4$ или $x = -29/10$). В обучающей выборке подобные финальные шаги отсутствовали. При обработке правильного, но структурно незнакомого ответа линейного уравнения модель определяла его как ошибочный.

Этот эффект «гиперкоррекции» показал, что нейросеть успешно усвоила синтаксис квадратных уравнений, но сохранила необходимость в дополнительной настройке для работы с универсальными правилами алгебры.

## 4. РАЗЛИЧИЯ В АРХИТЕКТУРАХ РАССМОТРЕННЫХ МОДЕЛЕЙ

Сравнительный анализ архитектурных особенностей моделей DeepSeek-R1-Distill-Qwen-1.5B и InternLM2-Math-Plus-1.8B продемонстрировал различия в их программной реализации и совместимости с современными методами оптимизации.

Модель DeepSeek-R1-Distill-Qwen основана на стандартной архитектуре Qwen2.5, полностью интегрированной в экосистему библиотеки transformers с классическими слоями типа nn.Linear. Такая структура нативно совместима с широким спектром инструментов дообучения, поскольку вычислительные графы для стандартных линейных проекций $Q$, $K$ и $V$ предопределены и стабильны.

В ходе экспериментов по дообучению модели InternLM2-Math-Plus-1.8B методами DoRA, QDoRA и LoftQ были зафиксированы программные сбои: значение функции потерь оставалось равным NaN с первых шагов обучения. Анализ показал, что такое поведение вызвано программным конфликтом между реализацией перечисленных методов в библиотеке “peft” и архитектурными особенностями InternLM2.

Общая причина проблемы состояла в том, что в InternLM2 кастомная (пользовательская) реализация слоев через “trust_remote_code = True”, т. е. архитектура загружена как внешний Python-код, а не из стандартных реализаций transformers.

LoRA и QLoRA сработали потому, что они делали простую операцию: добавляли две низкоранговые матрицы $A$ и $B$ к замороженным весам. Это не требовало никакого знания о внутренней структуре слоев.

## ЗАКЛЮЧЕНИЕ

Проведен сравнительный анализ методов PEFT для задачи проверки хода решения квадратного уравнения. Применение метода дообучения LoRA позволило эффективно адаптировать большие языковые модели на потребительском

оборудовании (NVIDIA T4), существенно снижая требования к вычислительным ресурсам. Наилучший результат показала модель DeepSeek-R1-Distill-Qwen-1.5B, достигшая точности 98.67% на методе дообучения LoRA. Эта модель с меньшим числом параметров превзошла в точности более крупную модель InternLM2 примерно на 5%.

Полученные результаты подтвердили выводы современных исследований дистилляции знаний, которые утверждают, что языковые модели способны превосходить более крупные архитектуры, если в процесс их предобучения включена дистилляция пошаговых логических рассуждений [20].

Эксперимент с проверкой переноса знаний на линейные уравнения выявил ограничения языковых моделей DeepSeek-R1-Distill и InternLM2. Обе модели продемонстрировали падение точности, что свидетельствует о явлении переобучения на целевой домен (domain overfitting), характерном для методов PEFT при смене контекста задачи [21]. Однако, модель DeepSeek-R1 превзошла InternLM2 и показала значительно лучшую обобщающую способность (успешно применила навыки верификации к незнакомому типу задач). Модель DeepSeek-R1 способна хорошо осуществлять перенос знаний: будучи дообученной исключительно на квадратных уравнениях, она успешно повысила точность верификации незнакомых ей линейных уравнений с 52% (в базовом состоянии) до 66.8%. Это доказало, что в процессе тонкой настройки модель не просто запоминает шаблоны квадратных уравнений, а также усваивает фундаментальные алгебраические паттерны проверки логики вычислений.

С точки зрения вычислительной эффективности квантованные методы продемонстрировали наилучшее соотношение точности и временных затрат. Это согласуется с результатами работ [22, 23], где показана практическая эффективность квантованных моделей с 4- и 8-битной разрядностями соответственно. В настоящей работе QLoRA и LoftQ с 4-битными весами сократили время дообучения DeepSeek-R1 в 3.5 раза относительно стандартного LoRA, оставаясь при этом лишь незначительно менее точными (97.77–98.00% против 98.67%).

Отметим также что полученные результаты подтвердили перспективность использования «рассуждающих» моделей (Reasoning Models) для построения интеллектуальных обучающих систем [24].

**СПИСОК ЛИТЕРАТУРЫ**

1. *Romera-Paredes B. et al.* Mathematical discoveries from program search with large language models // Nature. 2024. Vol. 625, No. 7995. P. 468–475. https://doi.org/10.1038/s41586-023-06924-6

2. *Anh-Hoang D., Tran V., Nguyen L.-M.* Survey and analysis of hallucinations in large language models: attribution to prompting strategies or model behavior // Frontiers in Artificial Intelligence. 2025. Vol. 8. Art. 1622292. https://doi.org/10.3389/frai.2025.1622292

3. *Ji Z., Lee N., Frieske R. et al.* Survey of hallucination in natural language generation // ACM Computing Surveys. 2023. Vol. 55, No. 12. Art. 248. P. 1–38. https://doi.org/10.1145/3571730

4. *Zhang Zh., Zheng Ch., Wu Y. et al.* The Lessons of Developing Process Reward Models in Mathematical Reasoning // Findings of the Association for Computational Linguistics: ACL 2025. Vienna, Austria, 2025. P. 10495–10516. https://doi.org/10.18653/v1/2025.findings-acl.547

5. *Hu E.J. et al.* LoRA: Low-Rank Adaptation of Large Language Models // International Conference on Learning Representations (ICLR). 2022.

6. *Dettmers T., Pagnoni A., Holtzman A., Zettlemoyer L.* QLoRA: Efficient Finetuning of Quantized LLMs // Advances in Neural Information Processing Systems. 2023. Vol. 36. P. 10088–10115. https://doi.org/10.52202/075280-0441

7. *Li Y., Yu Y., Liang C. et al.* LoftQ: LoRA-Fine-Tuning-Aware Quantization for Large Language Models // International Conference on Learning Representations (ICLR). 2024.

8. *Hayou S., Ghosh N., Yu B.* LoRA+: Efficient Low Rank Adaptation of Large Models // Proceedings of the 41st International Conference on Machine Learning (ICML). 2024. P. 17921–17943.

9. *Kabir S., Udo-Imeh D.N., Kou B., Zhang T.* Is Stack Overflow Obsolete? An Empirical Study of the Characteristics of ChatGPT Answers to Stack Overflow Questions // Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI ’24). Honolulu, HI, USA, 2024. Art. 935. P. 1–17. https://doi.org/10.1145/3613904.3642596

10. *Del Rio-Chanona R.M., Laurentsyeva N., Wachs J.* Large language models reduce public knowledge sharing on online Q&A platforms // PNAS Nexus. 2024. Vol.3, No. 9. Art. pgae400. https://doi.org/10.1093/pnasnexus/pgae400

11. *Shumailov I., Shumaylov Z., Zhao Y. et al.* AI models collapse when trained on recursively generated data // Nature. 2024. Vol. 631. P. 755–759. https://doi.org/10.1038/s41586-024-07566-y

12. *Yu L. et al.* MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models // International Conference on Learning Representations (ICLR). 2024.

13. *Sprague Z. et al.* To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning // Proceedings of the 13th International Conference on Learning Representations (ICLR). 2025.

14. *Jin R., Du J., Huang W. et al.* A Comprehensive Evaluation of Quantization Strategies for Large Language Models // Findings of the Association for Computational Linguistics: ACL 2024. Bangkok, Thailand, 2024. P. 12186–12215. https://aclanthology.org/2024.findings-acl.726/

15. *Пойманов Д.Р., Шутов М.С.* Исследование квантования больших языковых моделей: оценка эффективности с акцентом на русскоязычные задачи // Электронные библиотеки. 2025. Т. 28. № 5. С. 1138–1164.

16. *Liu S., Wang C., Yin H. et al.* DoRA: Weight-Decomposed Low-Rank Adaptation // Proceedings of the 41st International Conference on Machine Learning (ICML). 2024. P. 32100–32121.

17. *Weng Y. et al.* Large Language Models are Better Reasoners with Self-Verification // Findings of the Association for Computational Linguistics: EMNLP 2023. Singapore, 2023. P. 2550–2575. https://doi.org/10.18653/v1/2023.findings-emnlp.167

18. *Huang J., Chen X., Mishra S. et al.* Large Language Models Cannot Self-Correct Reasoning Yet // Proceedings of the 12th International Conference on Learning Representations (ICLR 2024). 2024.

19. *Park C., Yang Y., Lee C., Lim H.* Comparison of the Evaluation Metrics for Neural Grammatical Error Correction with Overcorrection // IEEE Access. 2020. Vol. 8. P. 106264–106272.

20. *Hsieh C.Y., Li J., Yeh C.K. et al.* Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes // Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023). 2023. P. 8003–8017.

21. *Treviso M., Lee J.-U., Ji T. et al.* Efficient Methods for Natural Language Processing: A Survey // Transactions of the Association for Computational Linguistics. 2023. Vol. 11. P. 826–860.

22. *Вершинин В.К., Ходненко И.В., Иванов С.В.* Нормализация текста, распознанного при помощи технологии оптического распознавания символов, с использованием легковесных LLM // Электронные библиотеки. 2025. Т. 28. № 5. С. 1036–1056.

23. *Нигматуллин А.Р., Лукманов Р.А., Таха А.* Квантование Vision Transformer: CPU-центричный анализ компромисса между размером модели и скоростью инференса // Электронные библиотеки. 2026. Т. 29. № 1. С. 262–286.

24. *Gupta A., Reddig J., Calò T., Weitekamp D., MacLellan C.J.* Beyond Final Answers: Evaluating Large Language Models for Math Tutoring // Artificial Intelligence in Education: 26th International Conference, AIED 2025. Palermo, Italy, July 22–26, 2025. Lecture Notes in Computer Science. Cham: Springer, 2025. Vol. 15877. P. 323–337. https://doi.org/10.1007/978-3-031-98414-3_23

# APPLICATION OF QUANTIZED ALGORITHMS FOR LANGUAGE MODEL ADAPTATION IN THE TASK OF VERIFYING THE SOLUTION PROCESS OF QUADRATIC EQUATIONS

**A. N. Khaybullin[1] [0009-0001-7321-956X], D. N. Tumakov[2] [0000-0003-0564-8335]**

[1, 2]*Kazan (Volga Region) Federal University, Kazan, Russia*

[1] almaz.khaybullin@mail.ru, [2] dmitri.tumakov@kpfu.ru

***Abstract***

This paper investigates quantized approaches to language model adaptation for the task of automatic step-by-step verification of the correctness of quadratic equation solutions. The study examines the effectiveness of Parameter-Efficient Fine-Tuning (PEFT) methods when adapting the DeepSeek-R1-Distill-Qwen-1.5B and InternLM2-Math-Plus-1.8B language models to build a mathematical verifier (Process-supervised Reward Model, PRM). Experiments were conducted on a synthetic dataset of quadratic equations augmented with negative sampling to simulate learner errors. A comparative evaluation of standard (LoRA, DoRA, rsLoRA) and quantized (QLoRA, QDoRA, LoftQ) fine-tuning algorithms was performed. Zero-shot transfer generalization was additionally assessed on a structurally distinct dataset of linear equations. Results show that quantization resolves numerical stability issues for non-standard architectures (InternLM2) while achieving quality comparable to standard methods. For the DeepSeek-R1 model, QLoRA, LoftQ, and QDoRA reached accuracies of 97.77%, 98%, and 98% respectively, only marginally below the standard LoRA (98.67%). Similarly, for the non-standard InternLM2 architecture, QLoRA achieved 92.67% accuracy (vs. 93% for baseline LoRA). However, full-precision algorithms (LoRA) tend to preserve richer representations of learned patterns, providing better knowledge transfer for Reasoning-class models (DeepSeek-R1 accuracy 66.8% vs. 61.4% for QLoRA on unseen data).

***Keywords:*** *language models, parameter-efficient fine-tuning (PEFT), quantized training methods, mathematical reasoning, automated solution verification, process-supervised reward models (PRM).*

**REFERENCES**

1. *Romera-Paredes B. et al.* Mathematical discoveries from program search with large language models // Nature. 2024. Vol. 625, No. 7995. P. 468–475. https://doi.org/10.1038/s41586-023-06924-6

2. *Anh-Hoang D., Tran V., Nguyen L.-M.* Survey and analysis of hallucinations in large language models: attribution to prompting strategies or model behavior // Frontiers in Artificial Intelligence. 2025. Vol. 8. Art. 1622292. https://doi.org/10.3389/frai.2025.1622292

3. *Ji Z., Lee N., Frieske R. et al.* Survey of hallucination in natural language generation // ACM Computing Surveys. 2023. Vol. 55, No. 12. Art. 248. P. 1–38. https://doi.org/10.1145/3571730

4. *Zhang Zh., Zheng Ch., Wu Y. et al.* The Lessons of Developing Process Reward Models in Mathematical Reasoning // Findings of the Association for Computational Linguistics: ACL 2025. Vienna, Austria, 2025. P. 10495–10516. https://doi.org/10.18653/v1/2025.findings-acl.547

5. *Hu E.J. et al.* LoRA: Low-Rank Adaptation of Large Language Models // International Conference on Learning Representations (ICLR). 2022.

6. *Dettmers T., Pagnoni A., Holtzman A., Zettlemoyer L.* QLoRA: Efficient Finetuning of Quantized LLMs // Advances in Neural Information Processing Systems. 2023. Vol. 36. P. 10088–10115. https://doi.org/10.52202/075280-0441

7. *Li Y., Yu Y., Liang C. et al.* LoftQ: LoRA-Fine-Tuning-Aware Quantization for Large Language Models // International Conference on Learning Representations (ICLR). 2024.

8. *Hayou S., Ghosh N., Yu B.* LoRA+: Efficient Low Rank Adaptation of Large Models // Proceedings of the 41st International Conference on Machine Learning (ICML). 2024. P. 17921–17943.

9. *Kabir S., Udo-Imeh D.N., Kou B., Zhang T.* Is Stack Overflow Obsolete? An Empirical Study of the Characteristics of ChatGPT Answers to Stack Overflow Questions // Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI ’24). Honolulu, HI, USA, 2024. Art. 935. P. 1–17. https://doi.org/10.1145/3613904.3642596

10. *Del Rio-Chanona R.M., Laurentsyeva N., Wachs J.* Large language models reduce public knowledge sharing on online Q&A platforms // PNAS Nexus. 2024. Vol.3, No. 9. Art. pgae400. https://doi.org/10.1093/pnasnexus/pgae400

11. *Shumailov I., Shumaylov Z., Zhao Y. et al.* AI models collapse when trained on recursively generated data // Nature. 2024. Vol. 631. P. 755–759. https://doi.org/10.1038/s41586-024-07566-y

12. *Yu L. et al.* MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models // International Conference on Learning Representations (ICLR). 2024.

13. *Sprague Z. et al.* To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning // Proceedings of the 13th International Conference on Learning Representations (ICLR). 2025.

14. *Jin R., Du J., Huang W. et al.* A Comprehensive Evaluation of Quantization Strategies for Large Language Models // Findings of the Association for Computational Linguistics: ACL 2024. Bangkok, Thailand, 2024. P. 12186–12215. https://aclanthology.org/2024.findings-acl.726/

15. *Poymanov D.R., Shutov M.S.* Issledovaniye kvantovaniya bolshikh yazykovykh modeley: otsenka effektivnosti s aktsentom na russkoyazychnyye zadachi [Study of Quantization of Large Language Models: Efficiency Evaluation with an Emphasis on Russian-Language Tasks] // Elektronnyye biblioteki [Russian Digital Libraries Journal]. 2025. Vol. 28, No. 5. P. 1138–1164.

16. *Liu S., Wang C., Yin H. et al.* DoRA: Weight-Decomposed Low-Rank Adaptation // Proceedings of the 41st International Conference on Machine Learning (ICML). 2024. P. 32100–32121.

17. *Weng Y. et al.* Large Language Models are Better Reasoners with Self-Verification // Findings of the Association for Computational Linguistics: EMNLP 2023. Singapore, 2023. P. 2550–2575. https://doi.org/10.18653/v1/2023.findings-emnlp.167

18. *Huang J., Chen X., Mishra S. et al.* Large Language Models Cannot Self-Correct Reasoning Yet // Proceedings of the 12th International Conference on Learning Representations (ICLR 2024). 2024.

19. *Park C., Yang Y., Lee C., Lim H.* Comparison of the Evaluation Metrics for Neural Grammatical Error Correction with Overcorrection // IEEE Access. 2020. Vol. 8. P. 106264–106272.

20. *Hsieh C.Y., Li J., Yeh C.K. et al.* Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes // Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023). 2023. P. 8003–8017.

21. *Treviso M., Lee J.-U., Ji T. et al.* Efficient Methods for Natural Language Processing: A Survey // Transactions of the Association for Computational Linguistics. 2023. Vol. 11. P. 826–860.

22. *Vershinin V.K., Khodnenko I.V., Ivanov S.V.* Normalizatsiya teksta, raspoznannogo pri pomoshchi tekhnologii opticheskogo raspoznavaniya simvolov, s ispolzovaniyem legkovesnykh LLM [Normalization of Text Recognized by Optical Character Recognition Using Lightweight LLMs] // Elektronnyye biblioteki [Russian Digital Libraries Journal]. 2025. Vol. 28, No. 5. P. 1036–1056.

23. *Nigmatullin A.R., Lukmanov R.A., Takha A.* Kvantovaniye Vision Transformer: CPU-tsentrichnyy analiz kompromissa mezhdu razmerom modeli i skorostyu inferensa [Quantization of Vision Transformer: A CPU-Centric Analysis of the Trade-Off Between Model Size and Inference Speed] // Elektronnyye biblioteki [Russian Digital Libraries Journal]. 2026. Vol. 29, No. 1. P. 262–286.

24. *Gupta A., Reddig J., Calò T., Weitekamp D., MacLellan C.J.* Beyond Final Answers: Evaluating Large Language Models for Math Tutoring // Artificial Intelligence in Education: 26th International Conference, AIED 2025. Palermo, Italy, July 22–26, 2025. Lecture Notes in Computer Science. Cham: Springer, 2025. Vol. 15877. P. 323–337. https://doi.org/10.1007/978-3-031-98414-3_23

## СВЕДЕНИЯ ОБ АВТОРАХ

***ХАЙБУЛЛИН Алмаз Наилевич*** – магистрант Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

***Almaz Nailevich KHAYBULLIN*** – master's student at the Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University.

email: almaz.khaybullin@mail.ru

ORCID: 0009-0001-7321-956X

***ТУМАКОВ Дмитрий Николаевич*** – заместитель директора по научной деятельности Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета, кандидат физико-математических наук.

***Dmitrii Nikolaevich TUMAKOV*** – deputy director for research at the Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University; Candidate of Physical and Mathematical Sciences, PhD (Cand. Sci. Phys.–Math.).

email: dtumakov@kpfu.ru

ORCID: 0000-0003-0564-8335

*Материал поступил в редакцию 3 мая 2026 года*