НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
июль–август 2025 Том 25 № 4 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
Jule–August 2025 Vol. 25 No 4 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

doi: 10.17586/2226-1494-2025-25-4-737-743
УДК 004.258

# Оптимизация моделей дистилляции знаний для языковых моделей

**Татьяна Михайловна Татарникова[1]✉, Никита Сергеевич Мокрецов[2]**

[1] Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, 190000, Российская Федерация
[2] Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, 197022, Российская Федерация

[1] Tm-tatarn@yandex.ru✉, https://orcid.org/0000-0002-6419-0072
[2] mokrecovnikita6374@gmail.com, https://orcid.org/0009-0009-1399-8504

**Аннотация**

**Введение.** Обсуждается проблема оптимизации больших нейронных сетей на примере языковых моделей. Размеры больших языковых моделей являются препятствием для их практического применения в условиях ограниченных объемов вычислительных ресурсов и памяти. Одним из развиваемых направлений сжатия моделей больших нейронных сетей служит дистилляция знаний — передача знаний от большой модели учителя к меньшей модели ученика без существенной потери точности результата. Известные в настоящее время методы дистилляции знаний имеют определенные недостатки: неточная передача знаний, долгий процесс обучения, накапливание ошибки в длинных последовательностях. **Метод.** Предлагаются методы, способствующие повышению качества дистилляции знаний применительно к языковым моделям: выборочное вмешательство учителя в процесс обучения ученика и низкоранговая адаптация. Первый подход основан на передаче токенов учителя при обучении ученика на слои нейронной сети, для которых достигается экспоненциально убывающий порог измерений расхождения между распределениями вероятностей учителя и ученика. Второй подход предлагает уменьшение количества параметров в нейронной сети путем замены полносвязных слоев на низкоранговые, что позволяет снизить риск переобучения и ускорить процесс обучения. Показаны ограничения каждого метода при работе с длинными последовательностями. Предложено комбинировать методы для получения усовершенствованной модели классической дистилляции знаний для длинных последовательностей. **Основные результаты.** Применение комбинированного подхода к дистилляции знаний на длинных последовательностях позволило значительно сжать результирующую модель с небольшой потерей качества, а также ощутимо снизить затрачиваемую память GPU и время вывода ответа. **Обсуждение.** Взаимодополняющие подходы к оптимизации процесса передачи знаний и сжатию моделей показали лучшие результаты, чем выборочное вмешательство учителя в процесс обучения ученика и низкоранговая адаптация по отдельности. Таким образом, качество ответов усовершенствованной модели классической дистилляции знаний на длинных последовательностях показало 97 % качества полной донастройки и 98 % качества метода низкоранговой адаптации по показателям ROGUE-L и Perplexity, при учете того, что количество обучаемых параметров снижается на 99 % по сравнению с полной донастройкой и на 49 % в сравнении с низкоранговой адаптацией. Кроме того, использование памяти GPU в сравнении с этими же методами уменьшается на 75 % и 30 % соответственно, а время вывода ответа на 30 %. Предложенная комбинация методов дистилляции знаний может найти применение в задачах с ограниченными вычислительными ресурсами.

**Ключевые слова**

большие языковые модели, длинные последовательности, нейронные сети, дистилляция знаний, модель учителя, модель ученика, выборочное вмешательство в процесс обучения, низкоранговая адаптация

**Ссылка для цитирования:** Татарникова Т.М., Мокрецов Н.С. Оптимизация моделей дистилляции знаний для языковых моделей // Научно-технический вестник информационных технологий, механики и оптики. 2025. Т. 25, № 4. С. 737–743. doi: 10.17586/2226-1494-2025-25-4-737-743

# Optimizing knowledge distillation models for language models

**Tatiana M. Tatarnikova[1]✉, Nikita S. Mokretsov[2]**

[1] Saint Petersburg State University of Aerospace Instrumentation (SUAI), Saint Petersburg, 190000, Russian Federation

[2] Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, 197022, Russian Federation

[1] Tm-tatarn@yandex.ru✉, https://orcid.org/0000-0002-6419-0072

[2] mokrecovnikita6374@gmail.com, https://orcid.org/0009-0009-1399-8504

**Abstract**

The problem of optimizing large neural networks is discussed using the example of language models. The size of large language models is an obstacle to their practical application in conditions of limited amounts of computing resources and memory. One of the areas of compression of large neural network models being developed is knowledge distillation, the transfer of knowledge from a large teacher model to a smaller student model without significant loss of result accuracy. Currently known methods of distilling knowledge have certain disadvantages: inaccurate knowledge transfer, long learning process, accumulation of errors in long sequences. The methods that contribute to improving the quality of knowledge distillation in relation to language models are proposed: selective teacher intervention in the student's learning process and low-level adaptation. The first approach is based on the transfer of teacher tokens when teaching a student to neural network layers, for which an exponentially decreasing threshold of measuring the discrepancy between the probability distributions of the teacher and the student is reached. The second approach suggests reducing the number of parameters in a neural network by replacing fully connected layers with low-rank ones, which reduces the risk of overfitting and speeds up the learning process. The limitations of each method when working with long sequences are shown. It is proposed to combine methods to obtain an improved model of classical distillation of knowledge for long sequences. The use of a combined approach to distilling knowledge on long sequences made it possible to significantly compress the resulting model with a slight loss of quality as well as significantly reduce GPU memory consumption and response output time. Complementary approaches to optimizing the knowledge transfer process and model compression showed better results than selective teacher intervention in the student learning process and low-rank adaptation separately. Thus, the quality of answers of the improved classical knowledge distillation model on long sequences showed 97 % of the quality of full fine-tuning and 98 % of the quality of the low-rank adaptation method in terms of ROGUE-L and Perplexity, given that the number of trainable parameters is reduced by 99 % compared to full fine-tuning and by 49 % compared to low-rank adaptation. In addition, GPU memory usage is reduced by 75 % and 30 %, respectively, and inference time by 30 %. The proposed combination of knowledge distillation methods can find application in problems with limited computational resources.

**Keywords**

large language models, long sequences, neural networks, knowledge distillation, teacher model, student model, selective intervention in the learning process, low-rank adaptation

**For citation:** Tatarnikova T.M., Mokretsov N.S. Optimizing knowledge distillation models for language models. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2025, vol. 25, no. 4, pp. 737–743 (in Russian). doi: 10.17586/2226-1494-2025-25-4-737-743

## Введение

Современные крупномасштабные языковые модели (Large Language Model, LLM) демонстрируют высокие результаты в различных задачах обработки естественного языка [1, 2]. Однако размеры таких моделей являются препятствием для их практического применения в условиях ограниченных объемов вычислительных ресурсов и памяти [3, 4]. По этой причине оптимизация использования ресурсов при работе с большими нейронными моделями представляет собой актуальную задачу.

В настоящее время, в качестве подхода для оптимизации модели нейронной сети, особое внимание уделяется идее дистилляции знаний. Суть данного подхода заключается в переносе знаний из точной, но громоздкой модели нейронной сети (учителя) в более компактную (ученика), с учетом ограничения вычислительных ресурсов или размеров чипа, на которых планируется запускать оптимизированную версию модели.

Для ускорения обучения языковых моделей при реализации дистилляции знаний предложены несколько методов, использующих выходные данные модели ученика (Student-Generated Outputs, SGO) [5]. Анализ источников [6, 7] показывает, что применение методов SGO приводит к уменьшению несоответствия между выводами при обучении и использовании модели, а также повышению производительности. Но такие заключения можно сделать относительно коротких последовательностей языковых моделей, для длинных последовательностей задача остается нерешенной. Этому есть несколько объяснений:

— методы SGO направлены на эффективное обучение модели ученика, поэтому необходимость точного ответа со стороны модели учителя часто игнорируется. Соответственно, ошибки при обучении, вызванные разрывом в размерах между моделями учителя и ученика, накапливаются в процессе обучения модели;

— эффект неточной передачи знаний: метод дистилляции знаний основан на предположении, что учитель предоставляет надежные данные, поэтому некорректное «руководство» со стороны модели учителя перерастает в серьезную проблему, при которой ученик получает высокие штрафы за правильные прогнозы и низкие за неправильные;

— авторегрессионная природа языковых моделей, которая способствует тому, что ошибки ученика могут накапливаться в длинных последовательностях.

Целью работы является исследование подходов к усовершенствованию классической дистилляции знаний для длинных последовательностей: стратегическое вмешательство модели учителя в процесс генерации последовательностей знаний учеником. При этом назначение модели учителя — эффективное обучение ученика и низкоранговая адаптация, назначение процесса генерации — повысить производительность обучения модели. Как ожидается комбинация этих подходов к дистилляции знаний на длинных последовательностях должна показать лучшие результаты.

## Методы

**Описание подхода дистилляции знаний.** Идея дистилляции знаний заключается в передаче знаний от модели учителя к модели ученика без существенной потери точности результата. Обозначим:

— модель учителя:

$$T = p(y|x),$$

где $x$ — входной набор данных для обучения; $y$ — вывод модели учителя;

— модель ученика:

$$S = q(y|x),$$

процесс передачи знаний — дистилляция знаний:

$$£ = (p(y|x), q(y|x)).$$

Классическая дистилляция знаний заключается в том, что на выводе модели сравниваются «логиты» учителя и ученика при одних и тех же входных данных. Логит — логарифмическая функция, используемая для преобразования вероятности в линейный интервал [8]. Эти значения — токены — используются для предсказания класса объекта и обычно преобразуются в вероятности с помощью функции активации Softmax.

Целью обучения модели ученика является минимизация расхождения $D$ между распределениями токенов моделей ученика и учителя, которое может быть оценено метрикой Дженсена–Шеннона (JSD) [9]:

$$JSD(p||q)(y|x) = \sum_{t=1}^{|y|} \sum_{y_t \in V} D(p||q)(y_t|y_{<t}, x),$$

где $p$ и $q$ — вероятностные распределения токенов для моделей учителя и ученика; $y$ — целевая последовательность; $x$ — запрос (входная последовательность); $y < t$ — подпоследовательность токенов до времени $(t - 1)$; $V$ — словарь всех возможных токенов; $D(p||q)$ — мера расхождения между распределениями вероятностей токенов моделей учителя и ученика.

На рисунке приведена упрощенная схема дистилляции знаний.

**Выборочное вмешательство учителя в процесс обучения ученика.** Основная идея данного метода заключается в выборочном переключении между моделями ученика и учителя для генерации следующего токена при обнаружении значительных расхождений между их вероятностными распределениями. Такая стратегия позволяет сбалансировать необходимость обучения ученика на своих данных и необходимость предотвращения накопления ошибок в длинных последовательностях [10].

Для управления этим процессом применяется экспоненциально убывающий порог, который увеличивает участие учителя по мере продвижения последовательности, предотвращая ошибочное руководство в длинных последовательностях

$$\tau_t = \tau_0 e^{-\lambda t},$$

где $\tau_0$ — начальный порог (установлен равным 1); $\lambda$ — скорость убывания, контролирующая, насколько быстро порог уменьшается с течением времени.

Для определения момента вмешательства учителя также используется расхождение JSD, которое в этом случае предоставляет симметричную и ограниченную меру различия между двумя распределениями вероятностей [1]:

$$JSD(p||q) = \frac{1}{2} D_{KL}(p||m) + \frac{1}{2} D_{KL}(q||m),$$

где $m = \frac{1}{2}(p + q)$ — средние распределения; $D_{KL}$ — KL-дивергенция (дивергенция Кульбака–Лейблера — мера расхождения между двумя вероятностными распределениями $p$ и $q$, для которых $D_{KL}(p||q) \neq D_{KL}(q||p)$).

На каждом временном шаге $t$ вычисляется JSD между распределениями ученика и учителя. Если расхождение превышает предопределенный порог $\tau_t$, происходит переключение с модели ученика на модель учителя для генерации следующего токена.

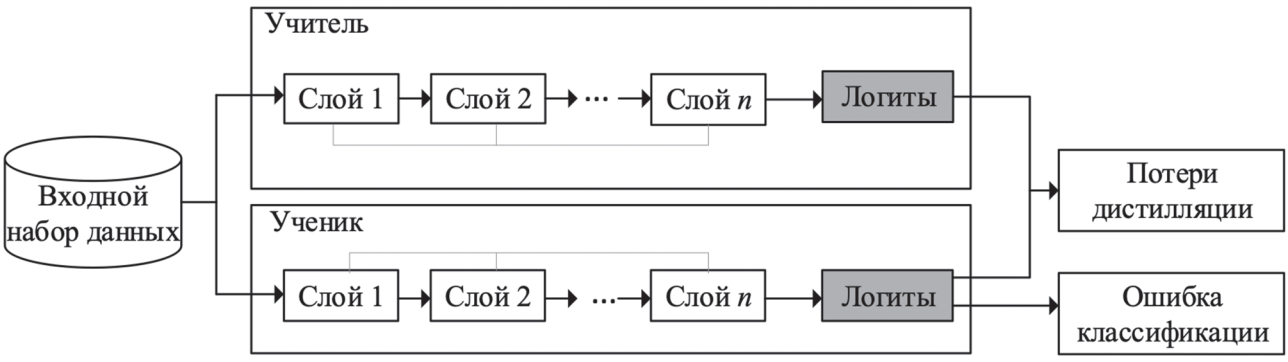


*Рисунок*. Архитектура дистилляции знаний

*Figure*. The architecture of knowledge distillation

## Результаты и обсуждение

Качество метода выборочного вмешательства учителя в процесс обучения ученика (SwitchLLM) оценим по следующим метрикам:

— Recall-Oriented Understudy for Gisting Evaluation for Longest Common Subsequence (ROGUE-L) — качество обобщения текста, используется для оценки задач автоматического создания краткого содержания (заголовка, резюме, аннотации) исходного текста и сгенерированного текста [11]. Метрика измеряет наибольшую последовательность слов, которая встречается как в выходном тексте модели, так и в эталонном тексте, позволяя словам быть в любом порядке. Чем выше значение ROGUE-L, тем лучше совпадение.

$$ROGUE\text{-}L = \frac{\sum_{S} LCS(S, S')}{\sum_{S} \sum_{gram_n \in S} \text{len}(S)},$$

где LCS — длина наибольшей общей подпоследовательности между эталонным текстом $S$ и текстом, который выдала модель $S'$; len($S$) — длина эталонного текста $S$; $gram_n$ — последовательности длиною $n$ символов ($n$-граммы).

— перплексия (Perplexity) — коэффициент неопределенности. Perplexity можно интерпретировать как обратную вероятность $P$ предложения, нормированную на количество слов $N$. В основном используется обратная вероятность: меньшее значение Perplexity указывает на более высокую точность модели.

$$Perplexity = \sqrt[N]{\frac{1}{P(w_1, w_2, \ldots, w_N,)}},$$

где $w_i$ — $i$-ое слово в предложении.

Предложения могут иметь разную длину, поэтому показатель не зависит от их размера.

Эксперимент проведен на бенчмарках Dolly и S-NI — базах эталонных задач для больших языковых моделей. Роль модели учителя выполняла вторая версия модели OpenLLaMA с 7 млрд параметров. Результаты метода SwitchLLM показаны в сравнении с методами SGO — MiniLLM и DistiLLM с таким же количеством параметров, как и для метода SwitchLLM, равным 3 млрд. Результаты методов дистилляции знаний и оценка дисперсии Var представлены в табл. 1.

**Низкоранговая адаптация.** Методы низкоранговой адаптации предназначены для эффективного дообучения больших языковых моделей с минимальными вычислительными затратами. Вместо того чтобы обновлять все параметры модели, как при полном дообучении, низкоранговая адаптация позволяет изменять лишь небольшую часть параметров, сохраняя большинство исходных весов модели фиксированными. Одна из популярных техник низкоранговой адаптации — метод Low-Rank Adaptation (LoRA), который уменьшает количество обучаемых параметров за счет введения небольших матриц разложения ранга, сохраняя при этом производительность, сравнимую с полной донастройкой во многих задачах [12].

Метод LoRA включает следующие этапы.

Этап 1. Выбирается предварительно обученная матрица весов **W0**.

Предварительно обученная матрица весов **W0** имеет размер $d \times k$.

Этап 2. Низкоранговая декомпозиция.

Обновление весов **ΔW** в матрице **W0** представляется в виде произведения двух матриц **B** и **A** низкого ранга размерностями $d \times r$ и $r \times k$.

Ранг $r$ намного меньше рангов $d$ и $k$, что обеспечивает более эффективное в вычислительном отношении представление.

Этап 3. Обучение.

В процессе обучения матрица **W0** остается неизменной — происходит «замораживание» весов.

*Таблица 1*. Оценки ROGUE-L и Perplexity различных методов дистилляции знаний

*Table 1.* ROGUE-L and Perplexity assessments of various knowledge distillation methods

| Метрика | Бенчмарк | Оценки | Метод (число параметров, млрд) | | | |
|---|---|---|---|---|---|---|
| | | | OpenLLaMA2 (7) | MiniLLM (3) | DistiLLM (3) | SwitchLLM (3) |
| ROGUE-L | Dolly | LCS ∈ [1; 10] | 28,80 | 27,40 | 28,30 | 28,60 |
| | | LCS ∈ [11; 20] | 16,60 | 16,50 | 20,20 | 26,50 |
| | | Var | 0,77 | 0,82 | 0,82 | 0,80 |
| | S-NI | LCS ∈ [1; 10] | 34,80 | 35,40 | 35,10 | 36,10 |
| | | LCS ∈ [11; 20] | 29,70 | 28,80 | 21,60 | 31,30 |
| | | Var | 0,79 | 0,84 | 0,85 | 0,83 |
| Perplexity | Dolly | $N = 10$ | 3,68 | 3,99 | 3,77 | 3,68 |
| | | $N = 20$ | 3,70 | 4,00 | 4,00 | 3,70 |
| | | Var | 0,11 | 0,41 | 0,37 | 0,32 |
| | S-NI | $N = 10$ | 3,69 | 3,99 | 3,78 | 3,70 |
| | | $N = 20$ | 3,70 | 4,10 | 4,10 | 3,90 |
| | | Var | 0,12 | 0,41 | 0,37 | 0,33 |

**A** и **B** являются обучаемыми параметрами — в них вносятся корректировки.

Этап 4. Умножение и сложение.

Предварительно обученная матрица весов **W0** и матрица обновлений **ΔW**, которая является продуктом **B** и **A**, умножаются на один и тот же вход **x**. Результаты этих умножений складываются

$$\mathbf{h} = \mathbf{W0x} + \mathbf{\Delta Wx} = \mathbf{W0x} + \mathbf{BAx},$$

где **h** — результат после применения обновлений к входу **x**.

Как видно, метод LoRA позволяет более эффективно обновлять матрицу больших весов, представляя обновления с использованием разложения низкого ранга, что сказывается на эффективности вычислений и использовании памяти.

Предлагаемое решение — комбинация выборочного вмешательства учителя в процесс обучения ученика и низкоранговой адаптации. Процесс начинается с выбора и донастройки модели учителя для поставленной задачи, где функция потерь выглядит следующим образом:

$$£^{P}_{task} = \frac{1}{|D_{task}|} \sum_{(x_i y_i \in D_{task}} £_{CE}(P(x_i) y_i),$$

где $£_{CE}$ — функция потерь перекрестной энтропии (Cross-Entropy loss), которая измеряет расхождение между предсказанными вероятностями $P(x_i)$ и метками $y_i$; $x_i$ — входные данные; $y_i$ — метки для входных данных; $D_{task}$ — обучающий набор данных.

Затем инициализируется более компактная модель ученика из того же семейства моделей, в которую интегрируются модули LoRA. Эти модули внедряются в слои **A** и **B** модели, где матрицы весов для модели ученика $q$ и значений весов v декомпозируются на предобученные веса и низкоранговые матрицы:

$$W_q = W_q^{base} + A_q B_q,\ W_v = W_v^{base} + A_v B_v,$$

где $W_q^{base}$, $W_v^{base}$ — предобученные веса модели ученика; $A_q$, $B_q$, $A_v$, $B_v$ — низкоранговые матрицы, которые являются обучаемыми параметрами.

Функция потерь для модели ученика комбинирует две компоненты: задачу минимизации ошибки на целевой задаче и задачу минимизации расхождения между выходами учителя и ученика. В результате общая функция потерь имеет вид:

$$£^{s}_{total} = \alpha £^{P}_{task} + (1 - \alpha) £_{KD}(z^s, z^t),$$

где α — весовой коэффициент; $£^{P}_{task}$ — функция потерь на целевой задаче, аналогичная модели ученика; $£_{KD}$ — функция потерь для дистилляции знаний.

Для подбора весового коэффициента a использовалась библиотека Optuna, суть которой заключается в обучении вариаций моделей с различными весовыми коэффициентами и получением наилучшего из них.

Функция потерь для общего метода дистилляции знаний представляет собой расхождение между выходами учителя и ученика и измеряется через KL-дивергенцию:

$$£_{KD}(z^s, z^t) = D_{KL}(p^s || p^t),$$

где $z^s$, $z^t$ — значения слоя Softmax моделей ученика и учителя; $p^s$, $p^t$ — распределение вероятностей, предсказанное моделями ученика и учителя; $D_{KL}$ — KL-дивергенция.

На этапе дистилляции знаний модель ученик обучается под руководством модели учителя, при этом обновляются только низкоранговые матрицы **A** и **B** метода LoRA. Функция потерь комбинирует задачу минимизации расхождения между выходами учителя и ученика (измеряемого через KL-дивергенцию) и задачу минимизации ошибки на целевой задаче. Параметр α управляет балансом между этими двумя компонентами.

Для оценки SwitchLLM-LoRA проводился следующий эксперимент:

— в качестве моделей учителей выбраны популярные модели BERT, RoBERTa и DeBERTaV3 [13, 14];
— моделями учениками выбраны компактные модели, которые принадлежат к тому же семейству, что и их более крупные модели учителя, в частности, для BERT-base, DeBERTa-v3-base и RoBERTa-base соответственно;

*Таблица 2*. Сравнение методов FFT, LoRA и SwitchLLM-LoRA
*Table 2.* Comparison of FFT, LoRA, and SwitchLLM-LoRA methods

| Модель учителя | Модель ученика | Метод | Числопараметров, млн | Память GPU, МБ | Время вывода, с |
|---|---|---|---|---|---|
| BERT-base | DistilBERT-base | FFT | 110 | 1322,0 | 6,10 |
| | | LoRA | 2,9 | 463,5 | 6,22 |
| | | SwitchLLM-LoRA | **1,2** | **296,8** | **5,36** |
| RoBERTa-base | DistilRoBERTa-base | FFT | 125 | 1515,9 | 7,21 |
| | | LoRA | 2,9 | 531,9 | 7,19 |
| | | SwitchLLM-LoRA | **1,2** | **358,3** | **4,44** |
| DeBERTa-v3-base | DeBERTa-v3-small | FFT | 183 | 2234,5 | 14,37 |
| | | LoRA | 2,9 | 763,4 | 15,62 |
| | | SwitchLLM-LoRA | **1,5** | **590,3** | **10,38** |

*Примечание*. Жирным шрифтом выделены результаты предложенного метода.

— сравнительных анализ проведен между тремя методами донастройки этих моделей: полная донастройка (full fine-tune, FFT), LoRA и SwitchLLM-LoRA;
— оценки качества моделей проведена на бенчмарке GLUE, который представляет собой набор из 9 задач по обработке естественного языка, предназначенных для оценки эффективности моделей в широком спектре задач, связанных с пониманием языка. Эти задачи включают в себя логику, анализ тональности, сходство текстов и многое другое. Бенчмарк GLUE является своего рода стандартом для сравнения способностей моделей понимать и обрабатывать текст. Результаты эксперимента приведены в табл. 2.

## Заключение

Показано, что выборочное вмешательство учителя в процесс обучения ученика превосходит прочие современные методы дистилляции знаний, использующих выходные данные модели ученика по метрике ROGUE-L, что подчеркивается важность вмешательства учителя при обучении ученика, вместо того чтобы полагаться только на выходные данные, сгенерированные учеником. Это указывает на то, что данный метод эффективно сокращает разрыв между моделями ученика и учителя, особенно когда количество параметров модели ученика значительно ниже, чем у модели учителя.

Подходы к усовершенствованию классической дистилляции знаний, рассмотренные в работе, демонстрируют потенциал в решении проблемы эффективного использования ресурсов при работе с большими языковыми моделями, но при этом имеют определенные ограничения. Метод выборочного вмешательство учителя в процесс обучения ученика способствует улучшению качества генерации последовательностей, метод низкоранговой адаптации способствует оптимизации использования параметров и снижении требований к памяти.

Поскольку рассмотренные методы усовершенствования классической дистилляции знаний предлагают различные, но взаимодополняющие подходы к оптимизации процесса передачи знаний и сжатия моделей, то их комбинация показала лучшие результаты: показатели ответов модели SwitchLLM-LoRA достигают 97 % качества FFT и 98 % качества LoRA по показателям ROGUE-L и Perplexity, при учете того, что количество обучаемых параметров снижается на 99 % по сравнению с FFT и на 49 % в сравнении с LoRA. Кроме того, использование памяти GPU уменьшается на 75 % и 30 % по сравнению с FFT и LoRA соответственно, а время вывода ответа на 30 %.

## Литература


1. Дудихин В.В., Кондрашов П.Е. Методология использования больших языковых моделей для решения задач государственного и муниципального управления по интеллектуальному реферированию и автоматическому формированию текстового контента // Государственное управление. Электронный вестник. 2024. № 105. С. 169–179. https://doi.org/10.55959/MSU2070-1381-105-2024-169-179
2. Кузнецов А.В. Цифровая история и искусственный интеллект: перспективы и риски применения больших языковых моделей // Новые информационные технологии в образовании и науке. 2022. № 5. С. 53–57. https://doi.org/10.17853/2587-6910-2022-05-53-57
3. Мокрецов Н.С., Татарникова Т.М. Алгоритм оптимизации моделей нейронных сетей для обработки текста на естественном языке // Прикладной искусственный интеллект: перспективы и риски: Сборник докладов Международной научной конференции. 2024. С. 280–282.
4. Houlsby N., Giurgiu A., Jastrzebski S., Morrone B., Laroussilhe Q., Gesmundo A., Attariyan M., Gelly S. Parameter-efficient transfer learning for NLP // Proc. of the 36th International Conference on Machine Learning. 2019. V. 97. P. 2790–2799.
5. Liao B., Meng Y., Monz C. Parameter-efficient fine-tuning without introducing new latency // Proc. of the 61st Annual Meeting of the Association for Computational Linguistics. 2023. V. 1. P. 4242–4260. https://doi.org/10.18653/v1/2023.acl-long.233
6. Lv K., Yang Y., Liu T., Guo Q., Qiu X. Full parameter fine-tuning for large language models with limited resources // Proc. of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024. V. 1. P. 8187–8198. https://doi.org/10.18653/v1/2024.acl-long.445
7. Khurana A., Subramonyam H., Chilana P.K. Why and when LLM-based assistants can go wrong: investigating the effectiveness of prompt-based interactions for software help-seeking // Proc. of the 29th International Conference on Intelligent User Interfaces. 2024. P. 288–303. https://doi.org/10.1145/3640543.3645200
8. Мокрецов Н.С., Татарникова Т.М. Оптимизация процесса обучения при ограниченном объеме вычислительных ресурсов // Международная конференция по мягким вычислениям и измерениям. 2024. Т. 1. С. 205–208.


## References


1. Dudikhin V.V., Kondrashov P.E. Methodology of using large language models to solve tasks of State and municipal government for intelligent abstracting and automatic generation of text content. *E-journal Public Administration*, 2024, no. 105, pp. 169–179. (in Russian). https://doi.org/10.55959/MSU2070-1381-105-2024-169-179
2. Kuznetsov A.V. Digital history and artificial intelligence: perspectives and risks of pretrained language models. *New Information Technologies in Education and Science*, 2022, no. 5, pp. 53–57. (in Russian). https://doi.org/10.17853/2587-6910-2022-05-53-57
3. Mokretsov N.S., Tatarnikova T.M. Algorithm for optimizing neural network models for natural language text processing. *Proc. of the Applied Artificial Intelligence: Prospects and Risks*, 2024. pp. 280–282. (in Russian)
4. Houlsby N., Giurgiu A., Jastrzebski S., Morrone B., Laroussilhe Q., Gesmundo A., Attariyan M., Gelly S. Parameter-efficient transfer learning for NLP. *Proc. of the 36th International Conference on Machine Learning*, 2019. vol. 97, pp. 2790–2799.
5. Liao B., Meng Y., Monz C. Parameter-efficient fine-tuning without introducing new latency. *Proc. of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023, vol. 1, pp. 4242–4260. https://doi.org/10.18653/v1/2023.acl-long.233
6. Lv K., Yang Y., Liu T., Guo Q., Qiu X. Full parameter fine-tuning for large language models with limited resources. *Proc. of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024, vol. 1, pp. 8187–8198. https://doi.org/10.18653/v1/2024.acl-long.445
7. Khurana A., Subramonyam H., Chilana P.K. Why and when LLM-based assistants can go wrong: investigating the effectiveness of prompt-based interactions for software help-seeking. *Proc. of the 29th International Conference on Intelligent User Interfaces*, 2024, pp. 288–303. https://doi.org/10.1145/3640543.3645200
8. Mokretsov N.S., Tatarnikova T.M. Optimizing the learning process with limited computational resources. *Proc. of the International Conference on Soft Computing and Measurement*, 2024, vol. 1, pp. 205–208. (in Russian)
9. Ouyang L., Wu J., Jiang X., Almeida D., Wainwright C., Mishkin P., Zhang C., Agarwal S., Slama K., Ray A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 2022, vol. 35, pp. 27730–27744.

9. Ouyang L., Wu J., Jiang X., Almeida D., Wainwright C., Mishkin P., Zhang C., Agarwal S., Slama K., Ray A., et al. Training language models to follow instructions with human feedback // Advances in Neural Information Processing Systems. 2022. V. 35. P. 27730–27744.
10. Borgeaud S., Mensch A., Hoffmann J., Cai T., Rutherford E., Millican K., et al. Improving language models by retrieving from trillions of tokens // Proc. of the 39th International Conference on Machine Learning. 2022. P. 2206–2240.
11. Белякова А.Ю., Беляков Ю.Д. Обзор задачи автоматической суммаризации текста // Инженерный вестник Дона. 2020. № 10 (70). С. 142–159.
12. Швыров В.В., Капустин Д.А., Кущенко А.В., Сентяй Р.Н. Дообучение больших языковых моделей с использованием техники LoRA для решения задач статического анализа программного кода // Вестник Луганского государственного университета имени Владимира Даля. 2023. № 12 (78). С. 210–215.
13. Liu Z., Lin W., Shi Y., Zhao J. A robustly optimized BERT pre-training approach with post-training // Lecture Notes in Computer Science. 2021. V. 12869. P. 471–484. https://doi.org/10.1007/978-3-030-84186-7_31
14. Jiao X., Yin Y., Shang L., Jiang X., Chen X., Li L., Wang F., Liu Q. TinyBERT: distilling BERT for natural language understanding // Findings of the Association for Computational Linguistics: EMNLP. 2020. P. 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

10. Borgeaud S., Mensch A., Hoffmann J., Cai T., Rutherford E., Millican K., et al. Improving language models by retrieving from trillions of tokens. *Proc. of the 39th International Conference on Machine Learning*, 2022, pp. 2206–2240.
11. Belyakova A.Y., Belyakov Y.D. Overview of text summarization methods. *Ingineering Journal of Don*, 2020, no. 10 (70), pp. 142–159 (in Russian)
12. Shvyrov V.V., Kapustin D.A., Kushchenko A.V., Sentyay R.N. Large language models fine-tuning with the LoRA technique to solve problems of static analysis of program code. *Vestnik of the Lugansk Vladimir Dahl National University*, 2023, no. 12 (78), pp. 210–215. (in Russian)
13. Liu Z., Lin W., Shi Y., Zhao J. A robustly optimized BERT pre-training approach with post-training. *Lecture Notes in Computer Science*, 2021, vol. 12869, pp. 471–484. https://doi.org/10.1007/978-3-030-84186-7_31
14. Jiao X., Yin Y., Shang L., Jiang X., Chen X., Li L., Wang F., Liu Q. TinyBERT: distilling BERT for natural language understanding. *Findings of the Association for Computational Linguistics: EMNLP*, 2020, pp. 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

## Авторы

**Татарникова Татьяна Михайловна** — доктор технических наук, профессор, директор института информационных технологий и программирования, Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, 190000, Российская Федерация, SC 36715607400, https://orcid.org/0000-0002-6419-0072, Tm-tatarn@yandex.ru

**Мокрецов Никита Сергеевич** — аспирант, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург, 197022, Российская Федерация, SC 57821230900, https://orcid.org/0009-0009-1399-8504, mokrecovnikita6374@gmail.com

## Authors

**Tatiana M. Tatarnikova** — D.Sc., Professor, Director of the Institute of Information Technologies and Programming, Saint Petersburg State University of Aerospace Instrumentation (SUAI), Saint Petersburg, 190000, Russian Federation, SC 36715607400, https://orcid.org/0000-0002-6419-0072, Tm-tatarn@yandex.ru

**Nikita S. Mokretsov** — PhD Student, Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, 197022, Russian Federation, SC 57821230900, https://orcid.org/0009-0009-1399-8504, mokrecovnikita6374@gmail.com

*Статья поступила в редакцию 25.03.2025*
*Одобрена после рецензирования 20.06.2025*
*Принята к печати 23.07.2025*

*Received 25.03.2025*
*Approved after reviewing 20.06.2025*
*Accepted 23.07.2025*