DOI 10.37882/2223-2966.2026.01.09

СИСТЕМАТИЗАЦИЯ МЕТОДОВ СЖАТИЯ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ ПО ФАЗЕ ПРИМЕНЕНИЯ И ХАРАКТЕРУ ВОЗДЕЙСТВИЯ

SYSTEMATIZATION OF METHODS FOR COMPRESSING LARGE LANGUAGE MODELS BY APPLICATION PHASE AND IMPACT TYPE

F. Bayazitov

Summary. *Relevance*. Modern large language models (LLMs) demonstrate high quality across a wide range of tasks, but their scale makes deployment challenging under limited computing resources. Existing reviews of LLM compression methods focus primarily on technical taxonomy — pruning, quantization, distillation — and fail to address either the timing of interventions in the model lifecycle or the nature of modifications. This limits the ability to consciously select a compression strategy based on the available modification stages and the required balance between efficiency and flexibility.

Research Hypothesis. A two-dimensional classification by application phase and intervention type allows us to identify a fundamental tradeoff between structural robustness and contextual adaptability, enabling an informed choice of compression method based on the available intervention stage and the nature of the target constraints.

Results. The proposed classification reveals a consistent trend: a shift from static methods in early phases to dynamic methods at the inference stage. The analysis shows that static methods provide predictable resource reduction but require model modification, while dynamic methods preserve the original weights but are context— and hardware-dependent. The most significant gap is the lack of dynamic methods in early stages. The findings form the basis for an informed choice of LLM compression strategy based on available intervention stages and practical constraints.

Keywords: large language models, model compression, dynamic compression, static compression, life cycle.

Баязитов Фанур Анурович
Уфимский государственный нефтяной технический университет
fanur-bayazitov@mail.ru

Аннотация: *Актуальность*. Современные большие языковые модели (LLM) демонстрируют высокое качество в широком спектре задач, однако их масштаб затрудняет развёртывание в условиях ограниченных вычислительных ресурсов. Существующие обзоры методов сжатия LLM фокусируются преимущественно на технической таксономии — прореживание, квантование, дистилляция — и не отражают ни времени вмешательства в жизненный цикл модели, ни природы изменений. Это ограничивает возможность осознанного выбора стратегии сжатия в зависимости от доступных этапов модификации и требуемого баланса между эффективностью и гибкостью.

Гипотеза исследования. Двумерная классификация по фазе применения и типу воздействия позволяет выявить фундаментальный компромисс между структурной устойчивостью и контекстной адаптивностью, что делает возможным обоснованный выбор метода сжатия с учётом доступного этапа вмешательства и природы целевых ограничений.

Результаты. Предложенная классификация выявляет устойчивую тенденцию: смещение от статических методов на ранних фазах к динамическим — на этапе инференса. Анализ показывает, что статические методы обеспечивают предсказуемое сокращение ресурсов, но требуют модификации модели, тогда как динамические сохраняют исходные веса, но зависят от контекста и аппаратуры. Наиболее значимый пробел — отсутствие динамических методов на ранних этапах. Полученные выводы формируют основу для осознанного выбора стратегии сжатия LLM в зависимости от доступных этапов вмешательства и практических ограничений.

Ключевые слова: большие языковые модели, сжатие моделей, динамическое сжатие, статическое сжатие, жизненный цикл модели.

Введение

Рост масштабов больших языковых моделей (Large language models, LLMs) сопровождается увеличением вычислительных, энергетических и инфраструктурных затрат, что ограничивает их развёртывание в ресурсоограниченных средах. В ответ на эту проблему разработано множество методов сжатия, направленных на снижение числа параметров, объёма памяти или вычислительной сложности без существенной потери качества. Существующие обзоры преимущественно классифицируют такие методы по техническому принципу: прореживание, квантование, дистилляция и другие.

В настоящей работе предлагается альтернативная двумерная систематизация методов сжатия LLM, основанная на двух независимых измерениях: фазе применения и характере воздействия. На основе анализа ключевых работ демонстрируется, что такая классификация выявляет устойчивую эволюционную тенденцию: смещение от статически сжатых моделей к адаптивным стратегиям, активным только на этапе инференса. Кро-

ме того, схема позволяет идентифицировать незаполненные ниши, в частности — отсутствие динамических методов на ранних фазах жизненного цикла. Цель статьи предложить аналитический инструмент для их осмысленного сопоставления и выбора в зависимости от практических ограничений.

Предварительные сведения

Под сжатием LLM понимается совокупность методов, направленных на снижение ресурсоёмкости модели при сохранении её функциональных характеристик. Основные целевые метрики включают: объём памяти, задержку инференса, пропускную способность и качество.

Текущие модели базируются на архитектуре трансформера, включающей слои внимания и полносвязные нейронные сети. Сжатие может затрагивать веса, активации или вычислительный граф. Важным различием является характер изменений: статическое сжатие приводит к созданию новой, фиксированной модели, в то время как динамическое сохраняет исходные параметры и модифицирует вычисления лишь в ходе инференса, в зависимости от входных данных. Далее эти понятия лежат в основе предлагаемой классификации.

Двумерная классификация методов сжатия LLM

Существующие обзоры методов сжатия LLM преимущественно опираются на таксономии, основанные на техническом принципе реализации: квантование, прунинг, дистилляция знаний и т.п. [1, 3]. Такой подход, хотя и обеспечивает удобную категоризацию, зачастую размывает принципиальные различия во времени и характере вмешательства в модель, что ограничивает аналитическую ценность сравнения методов.

В настоящей работе предлагается альтернативная двумерная классификационная схема, позволяющая систематизировать методы сжатия по двум независимым, но взаимодополняющим измерениям, по фазе применения и по характеру воздействия.

Фаза применения — временной этап жизненного цикла модели, на котором осуществляется сжатие. Делится на 4 типа: pre-training — сжатие интегрировано непосредственно в процесс масштабного предобучения; post-training — сжатие применяется к полностью обученной модели без дополнительной адаптации; fine-tuning — сжатие совмещено с адаптацией модели к конкретной задаче или домену; инференс — сжатие реализуется динамически в процессе генерации ответа.

Характер воздействия — природа изменения, вносимого в модель. Можно выделить 2 глобальных способа: статическое воздействие — предполагает постоянное и необратимое изменение структуры или параметров модели; динамическое воздействие — означает контекстно-зависимое изменение, активное только в ходе отдельного прохода инференса. В первом случае результат сжатия фиксирован и не зависит от входных данных, во втором структура исходной модели остаётся нетронутой.

Предложенная схема превосходит одномерные таксономии, поскольку одновременно раскрывает эволюцию подходов к сжатию, выявляет пробелы в существующих исследованиях и усиливает аналитическую применимость анализа. В частности, она отражает фундаментальный переход от универсального, статического сжатия модели к контекстно-зависимому сжатию по мере приближения к этапу развёртывания. Та же классификационная структура выявляет недостаточную разработанность динамических методов сжатия на ранних этапах жизненного цикла модели — в частности, в pre-training и post-training, что указывает на перспективность разработки архитектур с встроенной адаптивной разрежённостью. Кроме того, классификация связывает выбор метода сжатия не только с целевыми ограничениями, но и с возможностями вмешательства на конкретном этапе эксплуатации модели.

Наглядное представление предложенной классификации дано на рис. 1.

Диаграмма отображает двумерное пространство, в котором горизонтальная ось соответствует последовательности фаз применения — от pre-training к инференсу, а вертикальное разделение обозначает характер воздействия: статическое и динамическое. Такая визуализация подчёркивает эволюционную траекторию методов сжатия и позволяет наглядно идентифицировать структурные пробелы в существующих исследованиях.

Обзор методов сжатия в рамках предложенной классификации

Сжатие на этапе pre-training

На этапе pre-training сжатие интегрируется непосредственно в процесс масштабного обучения модели. Работа [7] представляет подход, в котором разреженность вводится на ранних стадиях предобучения моделей. Такое вмешательство носит статический характер: итоговая модель фиксирована, её структура не меняется в зависимости от входных данных, а улучшения достигаются за счёт сокращения числа параметров и операций на этапе последующего развёртывания.

Post-training сжатие

Данный тип сжатия применяется к полностью обученной модели без какого-либо дополнительного обу-

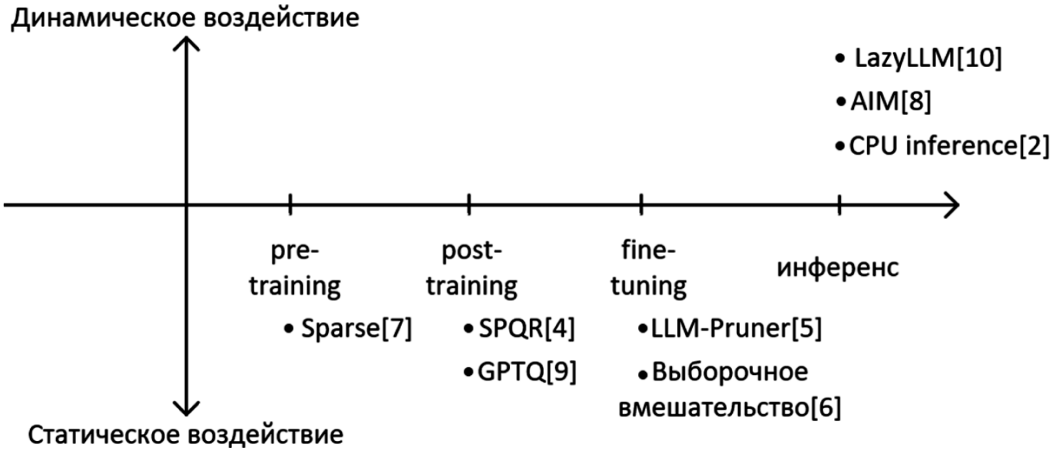

Рис. 1. Классификация методов сжатия LLM

Источник: анализ автора

чения или адаптации. Метод SPQR [4] реализует сжатие весов путём комбинации структурированного разреживания и точного квантования с восстановлением потерь через линейное уравнение. В работе GPTQ [9] реализуют 4-битное квантование с коррекцией ошибок, также вносятся статические изменения в модель. Оба метода позволяют резко сократить объём памяти при минимальной деградации качества, но требуют совместимых инференс-движков.

Сжатие при дообучении

Этап дообучения предоставляет возможность адаптировать сжатие под конкретную задачу или домен. В работе [5] предложен метод LLM-Pruner, который выполняет структурное прореживание на основе градиентной чувствительности модулей модели в ходе fine-tuning. Удаление компонентов происходит итеративно, с сохранением функциональной эквивалентности на уровне представлений, что позволяет минимизировать падение качества. Аналогично, [6] разрабатывают метод дистилляции знаний, в котором процесс обучения меньшей модели модифицируется через выборочное вмешательство, направленное на сохранение ключевых семантических свойств. Оба подхода приводят к созданию новой, компактной, но фиксированной модели, что соответствует статическому типу воздействия.

Сжатие во время инференса

На этапе инференса сжатие реализуется динамически, адаптируясь к каждому конкретному входному запросу без изменения исходных весов модели. В методе [2] предлагается оптимизация инференса LLM на центральных процессорах, в котором комбинируются post-training квантование и runtime-оптимизации, адаптивные к аппаратной архитектуре и структуре входа. В работе [8] вводят метод AIM (Adaptive Inference for Multi-modal LLMs), основанный на слиянии и прореживании токенов в ходе генерации: малоинформативные или избыточные токены объединяются или исключаются на лету, что сокращает вычислительную нагрузку пропорционально сложности запроса. В LazyLLM [10] реализуют динамическое прореживание токенов на основе их семантической важности. Все три подхода являются динамическими: вычислительная экономия достигается за счёт контекстно-зависимых решений, а исходная модель остаётся нетронутой, что обеспечивает полную обратную совместимость и гибкость развёртывания.

Сравнительный анализ и ограничения

Предложенная классификация позволяет не только систематизировать существующие методы, но и выявить устойчивые паттерны, ограничения и незаполненные пространства исследований. В таблице 1 обобщены ключевые характеристики шести рассмотренных подходов по трём измерениям: фаза применения, тип воздействия и основные практические свойства.

При анализе выявлены три ключевых наблюдения. Во-первых, все статические методы необратимы: они порождают новую, фиксированную модель, что исключает возможность использования исходной полноразмерной версии без хранения двух копий. Во-вторых, динамические методы обеспечивают полную обратную совместимость, поскольку не модифицируют веса, но их выгода проявляется только в инференсе и часто зависит от аппаратной платформы или структуры запроса. В-третьих, фаза применения напрямую коррелирует с гибкостью: чем ближе метод к инференсу, тем выше его адаптивность, но ниже предсказуемость выигрыша.

Наиболее значимый пробел, обнаруживаемый предложенной схемой, — отсутствие динамических методов на этапах pre-training и post-training. Все подходы на этих фазах являются статическими. Это указывает на недостаточную проработанность концепции условно-разрежён-

Таблица 1.

Сравнительные характеристики методов сжатия LLM

Метод	Фаза	Тип	Снижение размера	Падение качества	Аппарат. зависим.	Обратная совм.
Sparse [7]	pre-training	статический	высокое	умеренное	низкая	—
SPQR [4]	post-training	статический	очень высокое	минимальное	средняя	—
GPTQ [9]	post-training	статический	очень высокое	минимальное	высокая	—
LLM-Pruner [5]	fine-tuning	статический	среднее	умеренное	низкая	—
Выбор. вмеш. [6]	fine-tuning	статический	среднее	умеренное	низкая	—
CPU Inference [2]	инференс	динамич.	низко-среднее	минимальное	высокая	полная
AIM [8]	инференс	динамич.	перем.	минимальное	средняя	полная
LazyLLM [10]	инференс	динамич	перем.	минимальное	средняя	полная

Источник: анализ автора

ных архитектур, в которых модель могла бы обучаться с «потенциалом» к динамическому сжатию, активируемому в инференсе.

Примечательно, что в современных исследованиях наблюдается концентрация усилий на поздних этапах жизненного цикла модели: методы, основанные на fine-tuning и инференса, представлены значительно шире, чем подходы, ориентированные на pre-training или post-training сжатие. Это отражает общий сдвиг в прикладных исследованиях в сторону адаптации и оптимизации уже существующих фундаментальных моделей, а не их изначального конструирования с учётом компактности.

В совокупности, выявленные направления — концентрация динамических методов на этапе инференса, отсутствие адаптивных стратегий на ранних фазах, а также противоположные свойства статических и динамических подходов — указывают на компромисс, лежащий в основе современных стратегий сжатия LLM: между структурной устойчивостью и контекстной адаптивностью. Данный компромисс проявляется не только в технических характеристиках методов, но и в их позиционировании вдоль жизненного цикла модели. Двумерная классификация по фазе применения и типу воздействия позволяет систематизировать этот компромисс, делая его явным и поддающимся анализу при проектировании или выборе стратегии сжатия.

Заключение

В статье предложена двумерная классификация методов сжатия больших языковых моделей, основанная на фазе применения и характере воздействия. Анализ ключевых работ в рамках этой схемы выявил устойчивую тенденцию — по мере приближения к этапу развёртывания наблюдается смещение от фиксированных, структурно модифицированных моделей к адаптивным, контекстно-зависимым стратегиям, не изменяющим исходные параметры. Этот сдвиг отражает эволюцию приоритетов в области — от универсального уменьшения размера модели к гибкой оптимизации её использования под конкретные условия инференса.

Практическая значимость предложенной систематизации заключается в том, что она позволяет соотнести выбор метода сжатия с двумя факторами: доступным этапом вмешательства в жизненный цикл модели и требуемым балансом между предсказуемостью ресурсных характеристик и адаптивностью к входным данным. Например, при отсутствии возможности дообучения или модификации модели после развёртывания предпочтение следует отдавать динамическим методам, несмотря на их аппаратную зависимость. Напротив, при наличии ресурсов на fine-tuning статические подходы обеспечивают более стабильное сокращение вычислительной нагрузки.

Анализ также обнажил существенный пробел: отсутствие динамических методов на этапах pre-training и post-training. Это указывает на потенциал для разработки архитектур, в которых разреженность или сжимаемость закладываются на ранних стадиях, но реализуются адаптивно в инференсе.

Таким образом, систематизация методов сжатия LLM через время и характер вмешательства позволяет выявить структурные закономерности и нерешённые задачи, что может способствовать более осознанному проектированию эффективных и практичных решений в условиях растущих требований к масштабируемости языковых моделей.

ЛИТЕРАТУРА

1. Xu C., McAuley J. A survey on model compression and acceleration for pretrained language models //Proceedings of the AAAI Conference on Artificial Intelligence. — 2023. — Т. 37. — №. 9. — С. 10566–10575.
2. Shen H., Chang H., Luo Y., [и др.] Efficient llm inference on cpus //Enhancing LLM Performance: Efficacy, Fine-Tuning, and Inference Techniques. — Cham: Springer Nature Switzerland, 2025. — С. 33–46.
3. Chavan A., Magazine R., Kushwaha S., [и др.] Faster and lighter llms: A survey on current challenges and way forward //arXiv preprint arXiv:2402.01799. — 2024.
4. Dettmers T., Svirschevski R., Egiazarian V. [и др.] Spqr: A sparse-quantized representation for near-lossless llm weight compression //arXiv preprint arXiv:2306.03078. — 2023.
5. Ma X., Fang G., Wang X. Llm-pruner: On the structural pruning of large language models //Advances in neural information processing systems. — 2023. — Т. 36. — С. 21702–21720.
6. Татарникова Т.М., Мокрецов Н.С. Метод дистилляции знаний для языковых моделей на основе выборочного вмешательства в обучение // Программные продукты и системы. 2025. №2.
7. Agarwalla A., Gupta A., Marques A. [и др.] Enabling high-sparsity foundational llama models with efficient pretraining and deployment //arXiv preprint arXiv:2405.03594. — 2024.
8. Zhong Y., Liu Z., Li Y. [и др.] Aim: Adaptive inference of multi-modal llms via token merging and pruning //Proceedings of the IEEE/CVF International Conference on Computer Vision. — 2025. — С. 20180–20192.
9. Frantar E., Ashkboos S., Hoefler T. [и др.] GPTQ: Accurate post-training quantization for generative pre-trained transformers //arXiv preprint arXiv:2210.17323. — 2022.
10. Fu Q., Cho M., Merth T. [и др.] Lazyllm: Dynamic token pruning for efficient long context llm inference //arXiv preprint arXiv:2407.14057. — 2024.

© Баязитов Фанур Анурович (fanur-bayazitov@mail.ru)
Журнал «Современная наука: актуальные проблемы теории и практики»