Аудируемая оценка многоролевых LLM-алгоритмов на реальных программных задачах

Халилов Джабраиль Эльнурович
Национальный исследовательский университет «Высшая школа экономики»
Исследовательский проект Xynapse / BVC
github.com/jabrailkhalil/xynapse

2026

Аннотация

Полезный LLM-алгоритм должен быть не только запущен, но и проверяем: каждое утверждение о качестве и стоимости должно восстанавливаться до конкретной задачи, prompt, patch и результата тестов. Такая оценка требует контролировать выбор задач, кодовый контекст, версию модели, полноту генерации, исполняемое тестирование и статистический анализ парных исходов. В статье описан аудируемый контур проверки Bounded Verification Council (BVC), созданный как экспериментальное продолжение дипломного проекта Xynapse. Основная матрица содержит 240 запусков: четыре метода на одних и тех же 60 задачах SWE-bench Verified. Дополнительно выполнены 90 строк аудита повторяемости и 45 запусков с разговорной формулировкой; общий объём официальной оценки равен 375 строкам. Все ожидаемые финальные результаты присутствуют, дубликаты отсутствуют, несовпадения hash prompt и content равны нулю. Для сравнения качества использованы парная разность долей, точный критерий Мак-Немара и стратифицированный paired bootstrap на 10 000 выборок. BVC решил 5/60 задач, один предварительный план — 7/60, direct — 4/60, полный совет — 6/60. Основное сравнение BVC с одним планом дало $-3{,}3$ процентного пункта, 95% интервал $[-10{,}0; +3{,}3]$, $p = 0{,}625$. Следовательно, гипотеза преимущества качества не подтверждена; одновременно нижняя граница интервала лежит ниже заранее заданного margin -5 п. п., поэтому неухудшаемость также не установлена. Практическая ценность контура состоит в том, что он превращает BVC из демонстрации идеи в измеримый инженерный объект и проверяемо разделяет статистический вывод, описательный сигнал и ресурсную эффективность.

Ключевые слова: аудируемость, воспроизводимость, LLM-агенты, программная инженерия, SWE-bench Verified, парный эксперимент, bootstrap, критерий Мак-Немара, Xynapse, BVC.

1. Введение

Бинарный ответ «patch прошёл тесты» кажется простой метрикой, однако путь к достоверному сравнению методов состоит из многих зависимых решений. Если методы видят разные файлы, задача перестаёт быть парной. Если задачи выбраны после просмотра outcome, оценка становится оптимистичной. Если провайдерский timeout удаляется как «неудачный запуск», нарушается полнота. Если методы сравниваются только по агрегированным процентам, игнорируется тот факт, что они решали одни и те же instances.

В выпускной квалификационной работе автора описана интеллектуальная среда Xynapse с интегрированным AI-ассистентом и многоролевым механизмом планирования [13]. Для переноса проектной идеи в научное утверждение потребовался отдельный контур, в котором алгоритм, данные и правила вывода фиксируются до наблюдения результата. Настоящая статья описывает именно этот контур. Программный проект Xynapse и его актуальная реализация доступны в открытом репозитории [12]; диплом цитируется как источник архитектуры, а репозиторий — как источник кода.

В отличие от демонстрации отдельных удачных patch, исследование использует реальные задачи SWE-bench Verified [3], одинаковый исходный контекст, четыре заранее определённых метода и официальный исполняемый evaluator. Результат каждого этапа сохраняется как машиночитаемый артефакт с SHA-256. Такая структура позволяет заново построить сводные таблицы и графики из исходных JSONL, не доверяя вручную переписанным значениям.

Основные принципы контура

- outcome-blind выборка и замороженный протокол;
- одинаковая задача, кодовый контекст и финальная patch-инструкция;
- task-level pairing между методами;
- сохранение невалидных и неуспешных результатов без post-hoc удаления;
- официальный запуск тестов, а не экспертная оценка правдоподобия;
- отдельные серии для повторяемости и разговорной формулировки;
- хэшируемая цепочка от матрицы генерации до итогового анализа.

Что этот контур даёт инженерной команде

- любой итоговый процент можно проследить до конкретной задачи, ответа, patch и официального теста;
- неуспехи и инфраструктурные случаи остаются видимыми, поэтому удачные примеры нельзя незаметно отбирать post hoc;
- качество и расход ресурсов анализируются раздельно, что позволяет честно использовать доказанную экономию без завышения claims;
- модель, выборку или тариф можно заменить, сохранив дизайн сравнения и повторно рассчитав агрегаты из машиночитаемых артефактов.

Это делает экспериментальный контур самостоятельным результатом: он пригоден не для одной презентации, а для последовательного развития BVC.

2. Исследовательские вопросы

Контур отвечает на четыре разных вопроса.

1. **Качество.** Решает ли BVC больше задач, чем один предварительный план? Это первичный contrast.
2. **Дополнительные ориентиры.** Как BVC соотносится с direct и полным советом?
3. **Эффективность.** Какой фактический расход вызовов, токенов и расчётной стоимости несёт каждый метод?
4. **Устойчивость.** Насколько бинарный outcome повторяется при temperature $= 0$ и сохраняется при переходе от формальной к разговорной постановке?

Разделение важно: положительная ресурсная метрика не превращается автоматически в утверждение о качестве, а единичное превышение процента успехов не объявляется доказанным преимуществом.

3. Данные и outcome-blind выборка

3.1. Источник задач

SWE-bench Verified представляет программную задачу как реальное issue и состояние репозитория, к которому модель должна построить применимый patch [3]. Версия Verified была сформирована как 500 примеров после независимой проверки каждой задачи тремя разработчиками и отсечения серьёзно недоопределённых условий или некорректных тестов [5]. Из зафиксированной ревизии набора `91aa3ed51b709be6457e12d00300a6a596d4c6a3` выбрано 60 instances. Выборка сформирована до генерации итоговых patch.

Таблица 1: Стратификация 60 задач по оценке трудоёмкости.

Страта	Число задач	Доля
Medium: 15 минут–1 час	30	50,0%
Hard: 1–4 часа	27	45,0%
Very hard: более 4 часов	3	5,0%

Задачи происходят из 11 Python-репозиториев. Наличие нескольких проектов уменьшает зависимость результата от одной кодовой базы, хотя выборка не предназначена для точной оценки каждого репозитория отдельно.

Отбор был полностью алгоритмическим. Взяты все 3 задачи > 4 часов, 27 из 42 задач длительностью 1–4 часа и 30 из 261 задачи длительностью 15 минут–1 час. Для двух неполных страт репозиторные квоты определялись весом $\sqrt{N_{\text{available}}}$, затем методом наибольших остатков; внутри каждой квоты задачи ранжировались по

$$\text{SHA256}(\texttt{bvc-main60-selective-v1} \| \text{NUL} \| \texttt{instance_id}).$$

В ranking не входили текст issue, patch, тесты, оценка модели либо ручное предпочтение. Метка сложности использовалась только при отборе и bootstrap, но не показывалась модели.

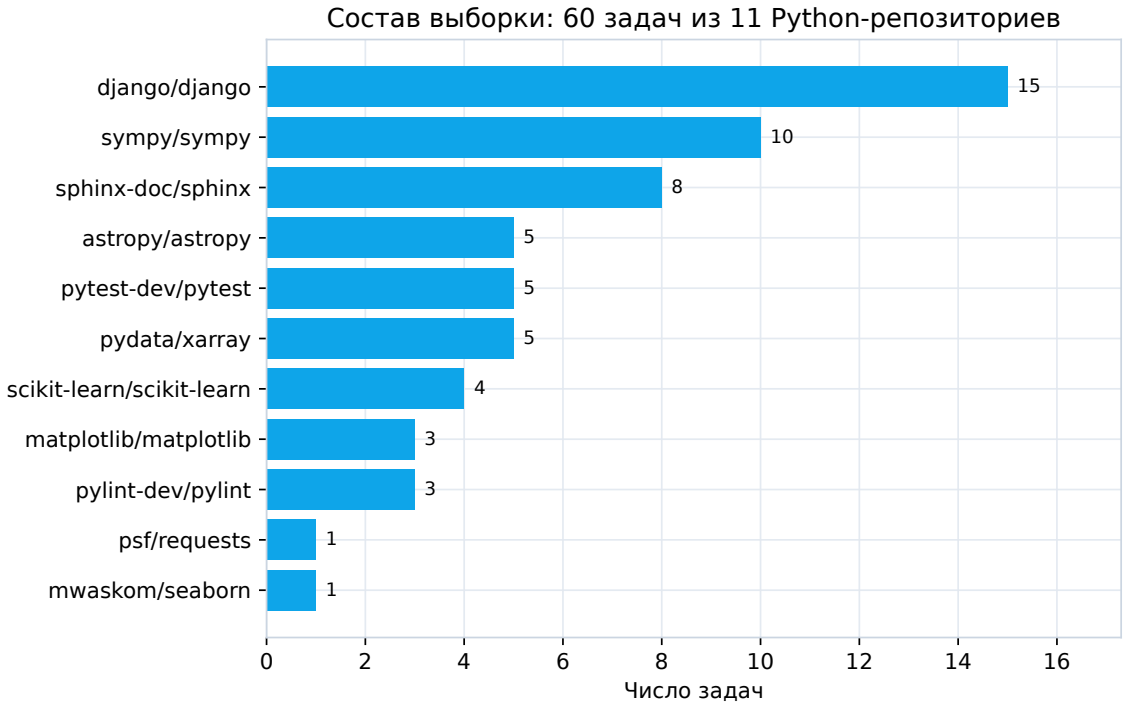


Рис. 1: Число выбранных задач по репозиториям.

3.2. Контролируемый кодовый контекст

Для каждого instance использовался замороженный retrieval из набора `princeton-nlp/SWE-bench_bm25_27K` ревизии `d4667abc...` [7]. BM25 ранжирует документы по совпадениям терминов с поправкой на частоту и длину документа; в данном эксперименте ranking не пересчитывался после просмотра outcome. Из официального артефакта извлекались только блоки `<code>`; legacy-поле `text` не передавалось, поскольку содержало общий демонстрационный patch.

Поиск дал от двух до восьми исходных файлов и от 71 369 до 111 192 символов на задачу; всего 245 файлов и 5 691 570 символов. Полученный manifest использовался всеми методами. Независимое сопоставление каждого файла с raw-содержимым нужного base commit дало 245/245 точных совпадений. В матрице сохранялся SHA-256 контекста по каждой задаче. Поэтому различие результата нельзя объяснить разным набором предоставленных файлов.

Граница утечки

Generation runner мог читать только `instance_id`, repository, base commit, issue и извлечённые source-файлы. Gold patch, test patch, `FAIL_TO_PASS`, `PASS_TO_PASS`, difficulty и selection rank находились в отдельном sealed-файле, доступном evaluator только после сохранения predictions.

4. Методы и экспериментальная матрица

4.1. Четыре метода основной серии

Таблица 2: Контролируемое различие между сравниваемыми методами.

Метод	Последовательность	Вызовов	Задач
Direct	немедленная генерация patch	1	60
Один план	структурированный план, затем patch	2	60
BVC	4 роли, адаптивная критика, синтез, patch	6 или 10	60
Полный совет	4 роли, обязательная критика, синтез, patch	10	60

Все методы используют одинаковую версию модели, temperature $= 0$, общий контекст и одинаковую финальную patch-инструкцию. Один план и оба совета передают patch-генератору совместимую структуру плана. Полный совет и BVC имеют одинаковые роли, оси, синтезатор и максимальный бюджет. Следовательно, адаптивность критики не смешана с наличием JSON-схемы либо самого многоролевого разложения.

4.2. Модель, лимиты и порядок исполнения

Таблица 3: Замороженная конфигурация generation runner.

Параметр	Значение
Провайдер / модель	Yandex Cloud AI Studio; requested `deepseek-v4-flash`; returned `gpt://deepseek-v4-flash/latest`
Температура / seed	0 / provider seed недоступен
Режим рассуждения	отдельный параметр thinking/non-thinking не передавался; использован provider default, поле `reasoning_content` сохранено
Prompt-версия	`main60-3.0.0`
Лимит роли	4 096 output-токенов
Лимит плана/синтеза	4 096 output-токенов
Лимит patch	16 384 output-токена; один repository-root unified diff
Порядок методов	детерминированная ротация `bvc-main60-schedule-v1`
Primary guard	не более 1 380 успешных вызовов, 1 518 попыток и 45 млн токенов

Каждый API-ответ сохранялся целиком вместе с requested limit, usage, finish reason, длительностью, prompt, plan/patch и SHA-256. Один primary patch-вызов после повторяющихся transport-timeout был выполнен с read timeout 600 вместо 180 секунд; запрос не изменялся и завершился за 195,688 секунды. Это отклонение касается транспорта и не меняет 60-задачные outcomes. Доступность DeepSeek V4 Flash в Yandex Cloud подтверждается официальным сообщением провайдера [10]. Однако алиас `latest` не является неизменяемым идентификатором весов: сохранённый returned model URI точно документирует обслуженный запрос, но сам по себе не гарантирует будущий запуск той же серверной ревизии.

4.3. BVC как заранее включаемый режим

Активационная политика BVC — `human_enabled_before_first_edit`. Сначала четыре независимые роли выдают решения по локализации причины, стратегии исправления, зависимостям и тестовому покрытию. Затем вычисляются D_{vote} и D_{cov}. При превышении порогов $0{,}30$ и $0{,}25$, либо при отсутствии оси с двумя валидными голосами, выполняется один раунд критики. Эта схема продолжает проектную модель, описанную в дипломе [13], но в эксперименте реализована как полностью исполняемый и хэшируемый протокол. Пороги были зафиксированы как инженерные значения до просмотра outcomes; они не настраивались на 60 тестовых задачах и не заявляются оптимальными.

4.4. Три слоя эксперимента

Основная матрица имеет размер

$$60 \text{ задач} \times 4 \text{ метода} = 240 \text{ строк}.$$

Для 15 заранее выделенных задач выполнены два дополнительных повторения трёх методов:

$$15 \times 3 \times 2 = 90 \text{ дополнительных строк}.$$

В сочетании с соответствующими 45 строками первой репетиции это даёт по три outcome на задачу и метод. Наконец, для тех же 15 задач создан один контролируемый разговорный paraphrase:

$$15 \times 3 = 45 \text{ строк}.$$

Полный объём официальной оценки:

$$240 + 90 + 45 = 375.$$

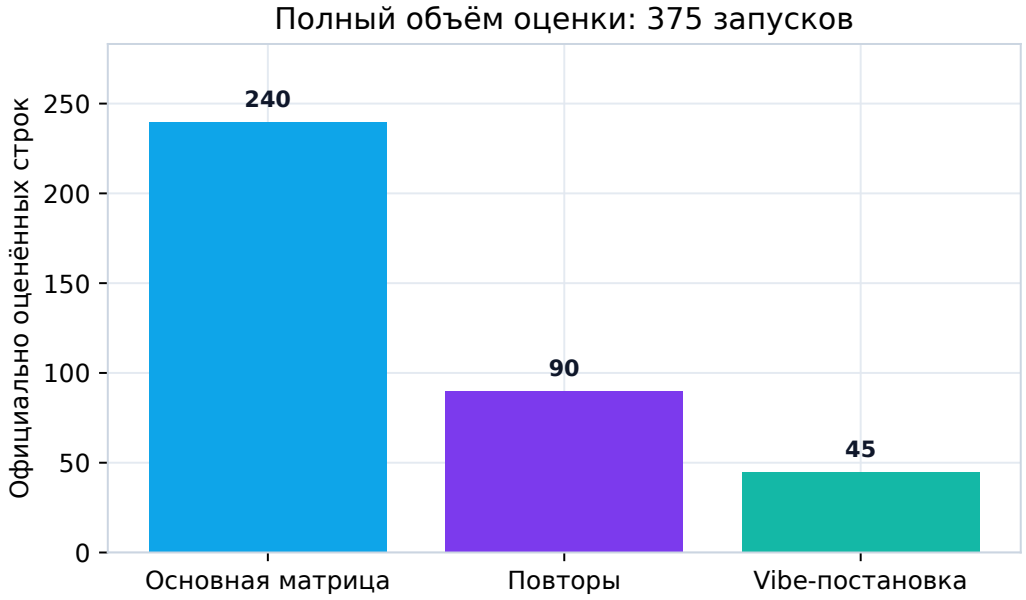


Рис. 2: Три слоя сохранённых официальных результатов.

5. Цепочка аудируемости и воспроизводимости

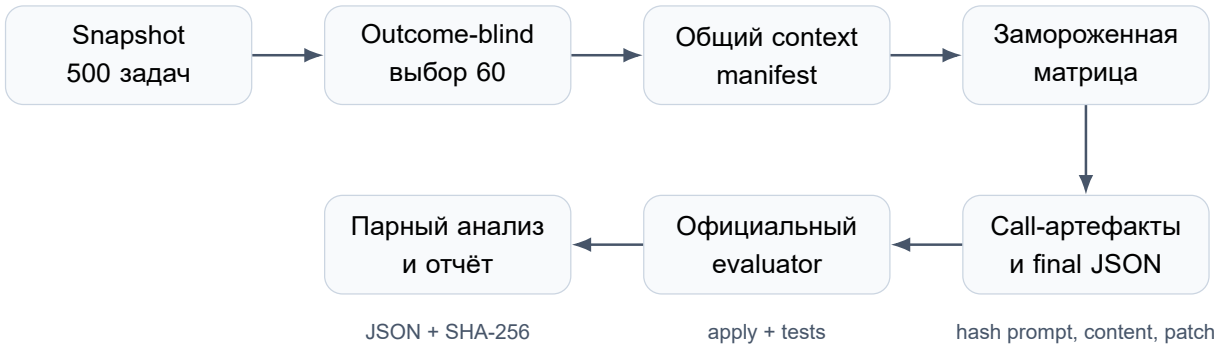


Рис. 3: Цепочка данных от snapshot до статистического вывода.

5.1. Заморозка до outcome

До основной генерации фиксируются:

- список задач и страты;
- контекстные manifests;
- prompts и их hash;
- version runner и модель;
- лимиты токенов, timeout и stop-rule;
- пороги gate;
- первичный contrast, bootstrap seed и правило множественных сравнений.

Это отделяет изменение исследовательского дизайна от исправления технической ошибки исполнения. Любое допустимое транспортное отклонение сохраняется в отдельном журнале и не удаляет уже наблюдавшиеся результаты.

Первичный contrast был заранее задан как BVC минус один план; margin практически важной потери составлял -5 п. п.. Bootstrap seed `bvc-main60-bootstrap-v1`, 10 000 реплик и Holm-коррекция для двух secondary-contrast также были зафиксированы до evaluation. Шесть других реальных задач служили development-набором для проверки JSON, diff и harness; они не входят ни в одну inferential-таблицу.

5.2. Сохранение всех исходов

Provider-successful вызов не гарантирует корректный diff. В основной серии 59/60 direct и 59/60 full-council patch были синтаксически разбираемы; BVC и один план дали 60/60. Два некорректных результата сохранены и оценены как сгенерированные outcomes. Результат не исключался по причине ошибки применения patch, изменения тестового файла или нулевого успеха.

5.3. Исполняемая оценка

Evaluator последовательно проверял существование patch, синтаксическую разбираемость, применимость к нужному checkout и прохождение тестов SWE-bench Verified. Только официальный статус `resolved=true` засчитывался как успех. Правдоподобное объяснение модели без проходящих тестов не считалось решением.

Использован официальный harness commit `f7bbbb2ccdf479001d6467c9e34af59e44a840f9`. Patch применялся к чистому base commit в контейнере; затем должны были пройти все требуемые `FAIL_TO_PASS` и `PASS_TO_PASS` тесты. Официальные per-instance reports были созданы для 18 BVC, 14 direct, 18 full-council и 20 single-plan predictions. Для неприменимых непустых patches официальный лог содержал буквальный статус `Patch Apply Failed`; такие строки сохранялись как `resolved=false`. Один пустой direct-patch и один

пустой full-council patch также остались неуспехами. Для бинарной метрики отсутствие report по любой другой причине также кодировалось как 0 и отдельно помечалось как инфраструктурный случай, а не исключалось из знаменателя.

Таблица 4: Переход от применимого patch к официально решённой задаче.

Метод	Применимо	Решено	Решено среди применимых
Direct	14	4	28,6%
Один план	20	7	35,0%
BVC	18	5	27,8%
Полный совет	18	6	33,3%

Условная доля «решено среди применимых» описывает воронку после генерации и не является отдельным причинным эффектом качества плана: применимость patch сама зависит от метода. Таблица нужна для локализации потерь между синтаксисом, применением и тестами, а не для post-hoc ранжирования алгоритмов.

Шесть primary-patch изменяли тестовые файлы. Все шесть официально неуспешны и сохранены, поэтому test-edit не создаёт ни одного функционального успеха и не объясняет сравнительный результат.

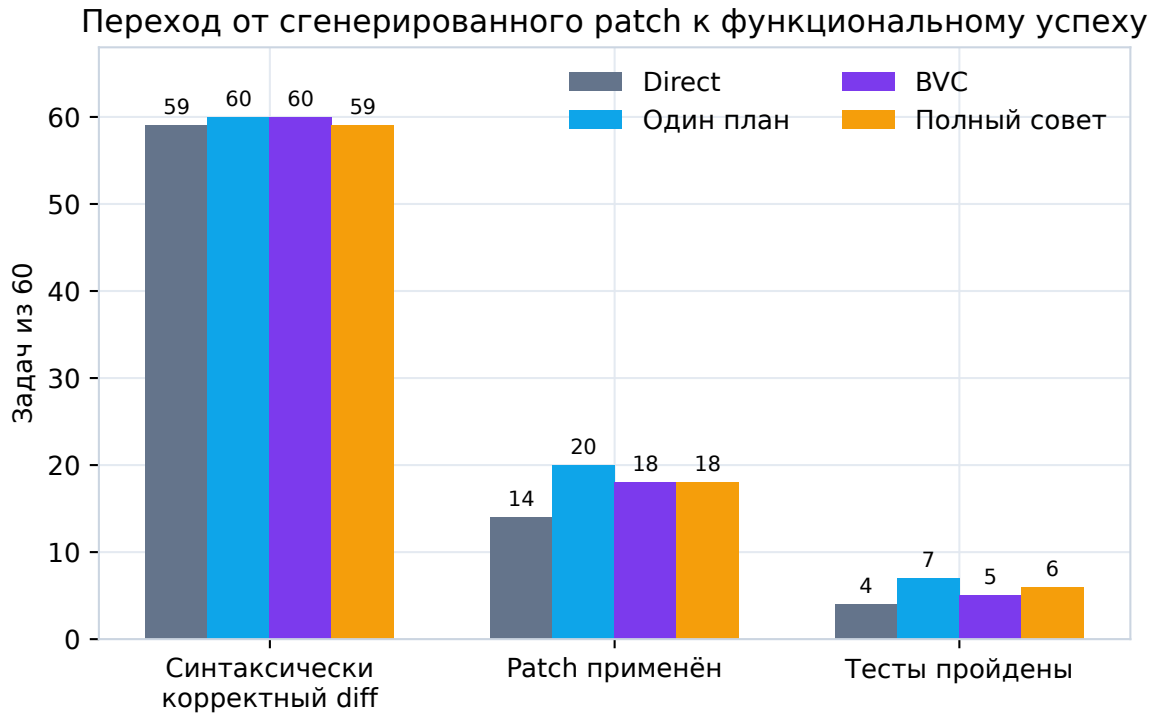


Рис. 4: Число результатов на трёх стадиях исполняемой проверки.

6. Проверки целостности

6.1. Полнота матриц

Таблица 5: Автоматические проверки трёх экспериментальных слоёв.

Слой	Ожидалось	Получено	Пропуски	Дубликаты	Статус
Основная матрица	240	240	0	0	passed
Дополнительные повторы	90	90	0	0	passed
Разговорная постановка	45	45	0	0	passed
Всего	375	375	0	0	complete

Во всех трёх слоях число несовпадений hash prompt равно нулю; число несовпадений hash content также равно нулю. В основной серии сохранено 1 268 call-артефактов, из них 1 255 успешных вызовов и 13 неуспешных попыток. В принятых финальных маршрутах 240 строк содержится 1 252 успешных вызова: $60 + 120 + 472 + 600$ для direct, одного плана, BVC и полного совета. Оставшиеся три успешных артефакта — плановые вызовы из вытесненных попыток 1–3 для `sympy__sympy-12489`; соответствующие patch-вызовы прерывались транспортно, а принятый результат получен в попытке 4. Эти три вызова входят в аудит фактического расхода, но не в число вызовов финального маршрута. Аудит не подменяет функциональные тесты, но подтверждает, что итоговая таблица относится к замороженным inputs. Совпадение hash проверяет буквальное содержимое сохранённых prompt и context; оно не доказывает семантическую корректность произвольной metadata-метки.

6.2. Контроль расхода

Общий cost-guard по всем созданным артефактам зарегистрировал 1 848 успешных вызовов, 54 533 372 токенов и 7 266,84 руб. расчётной стоимости при stop-пороге 11 800 руб. Здесь объём шире 375 официальных строк: он включает технические и незавершённые попытки, которые нужны для честного учёта фактического расхода.

7. Статистический план

7.1. Парная разность долей

Пусть $Y_{im} \in \{0, 1\}$ — официальный outcome задачи i для метода m. Для двух методов A и B

$$\widehat{\Delta}_{A-B} = \frac{1}{n} \sum_{i=1}^{n} (Y_{iA} - Y_{iB}).$$

Такое определение сохраняет task-level pairing. Например, одинаковый успех обоих методов даёт нулевой вклад, а успех только BVC — $+1/n$.

7.2. Точный критерий Мак-Немара

Пусть b — число задач, решённых только A, а c — только B. При нулевой гипотезе направления $b + c$ дискордантных пар равновероятны. Двустороннее точное значение:

$$p_{\text{exact}} = \min\left(1,\ 2 \sum_{k=0}^{\min(b,c)} \binom{b+c}{k} 2^{-(b+c)}\right).$$

Критерий использует именно те задачи, на которых методы различились [4].

7.3. Стратифицированный paired bootstrap

Для каждой из трёх страт сложности задачи повторно выбираются с возвращением внутри своей страты; затем страты объединяются с исходными размерами 30, 27 и 3. Для каждой из 10 000 детерминированных

реплик вычисляется $\widehat{\Delta}^*$. Интервал — 2,5-й и 97,5-й процентили. Так одновременно сохраняются pairing и состав по сложности [1].

Внутри одной bootstrap-реплики выбирается именно *задача*, а не отдельный outcome метода: вся пара (Y_{iA}, Y_{iB}) переносится вместе. Иначе корреляция методов на одной задаче была бы разрушена, а интервал отвечал бы на другой, непарный вопрос.

7.4. Интервалы отдельных долей и множественные сравнения

Для визуализации каждой доли используется Wilson-интервал [8]. Он не заменяет парный анализ. Сравнение BVC с одним планом объявлено первичным. Два secondary-contrast — с direct и полным советом — корректируются методом Holm [2]. Ни одно значение p не интерпретируется отдельно от числа дискордантных пар и интервала эффекта.

Для x успехов из n, $\widehat{p} = x/n$ и квантиля $z = 1{,}96$ центр Wilson-интервала равен

$$\widetilde{p} = \frac{\widehat{p} + z^2/(2n)}{1 + z^2/n},$$

а полуширина —

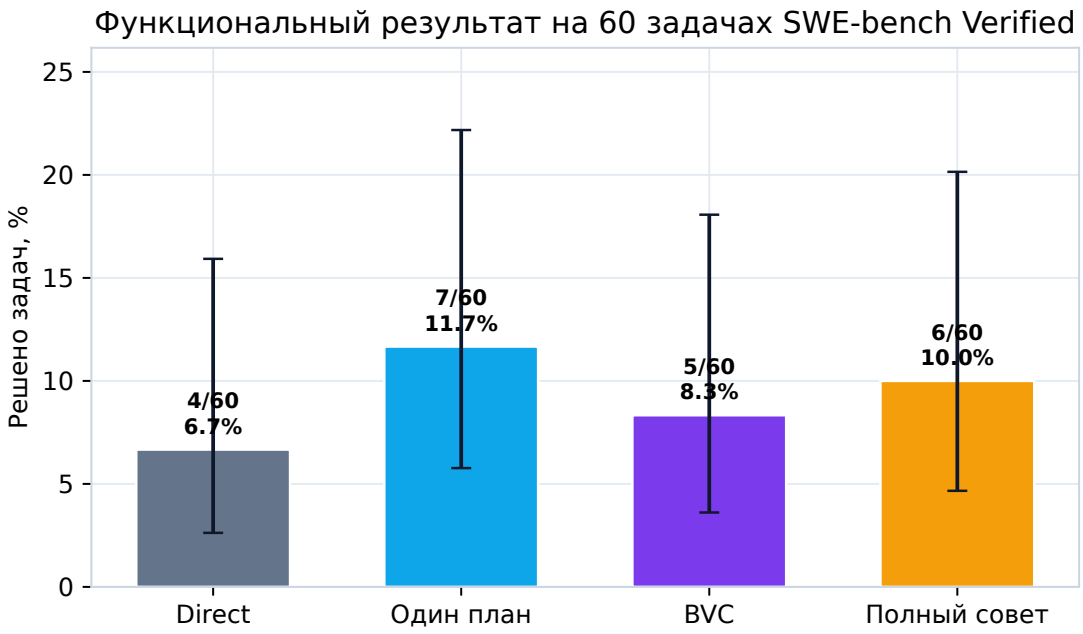
$$h = \frac{z}{1 + z^2/n}\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n} + \frac{z^2}{4n^2}}.$$

Показывается $[\widetilde{p} - h; \widetilde{p} + h]$. Для Holm два secondary-p сортируются; меньшее сравнивается с $\alpha/2$, следующее — с α. В текущих данных оба raw-p равны 1, поэтому adjusted-p также равны 1.

8. Результаты основной матрицы

Таблица 6: Функциональный результат и 95% Wilson-интервал.

Метод	Решено	Доля	95% Wilson CI
Direct	4/60	6.7%	[2.6; 15.9]%
Один план	7/60	11.7%	[5.8; 22.2]%
BVC	5/60	8.3%	[3.6; 18.1]%
Полный совет	6/60	10.0%	[4.7; 20.1]%



Вертикальные интервалы: 95% Wilson CI для каждой доли; сравнение методов является парным.

Рис. 5: Доли решённых задач; интервалы отдельных долей показаны для масштаба неопределённости.

Таблица 7: Парные сравнения BVC с тремя режимами.

Contrast	BVC-only	Baseline-only	Разность, п. п.	95% bootstrap	p
BVC — один план (primary)	1	3	$-3{,}3$	$[-10{,}0; +3{,}3]$	0,625
BVC — direct	1	0	$+1{,}7$	$[0{,}0; +5{,}0]$	1,000
BVC — полный совет	0	1	$-1{,}7$	$[-5{,}0; 0{,}0]$	1,000

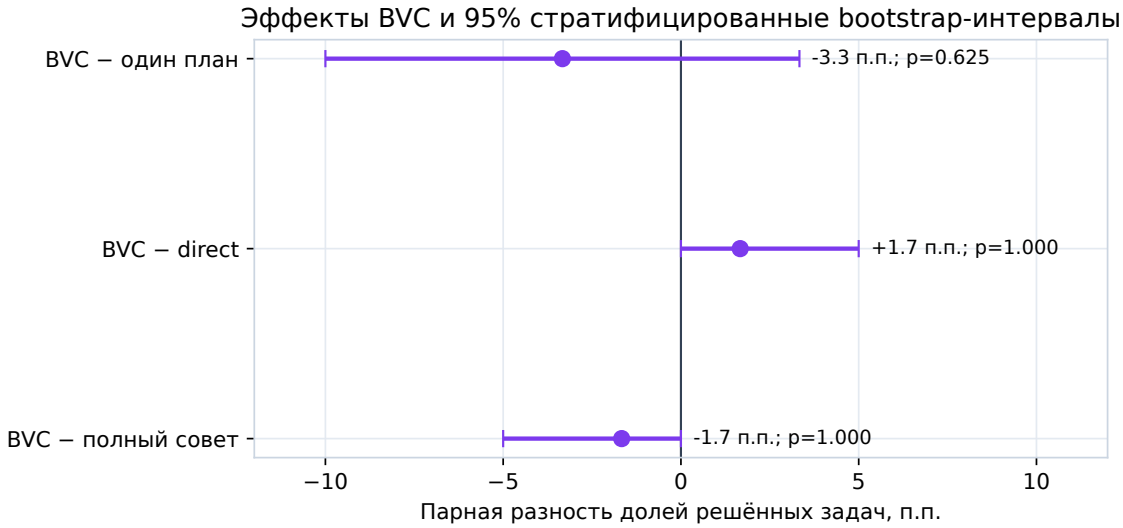


Рис. 6: Точечные оценки и замороженные bootstrap-интервалы парных эффектов.

Для двух secondary-contrast одна клетка дискордантности пуста. Поэтому процентильные интервалы вырождены по знаку: каждая bootstrap-реплика BVC–direct неотрицательна, а BVC–полный совет неположительна. Границы 0,0 в интервалах $[0{,}0; +5{,}0]$ и $[-5{,}0; 0{,}0]$ отражают структуру этой матрицы, а не являются самостоятельным подтверждением знака истинного эффекта.

Основная заранее заявленная гипотеза преимущества BVC над одним планом не поддержана: наблюда-

емая разность отрицательна, интервал включает ноль, точный тест не значим. С direct получен положительный описательный сигнал: пять решений против четырёх, причём одна задача решена только BVC. Но единственная дискордантная пара даёт $p = 1{,}0$; это основание для следующего эксперимента, а не доказательство превосходства. Заранее заданная неухудшаемость с margin -5 п. п. также не установлена: нижняя граница 95% интервала первичного эффекта равна -10 п. п..

9. Ресурсные метрики

Таблица 8: Фактическая эффективность в основной матрице.

Метод	Вызовы	Токены	Цена, руб.	Руб./задачу	Руб./решённую
Direct	60	1 867 093	312,44	5,21	78,11
Один план	120	3 585 838	522,84	8,71	74,69
BVC	472	14 282 809	1 838,46	30,64	367,69
Полный совет	600	18 365 172	2 402,61	40,04	400,44

Наиболее корректная ресурсная пара — BVC и полный совет, потому что они совпадают по ролям и максимальному маршруту. BVC сократил вызовы на 21,3%, токены на 22,2% и стоимость на 23,5%. На карте стоимости и качества точки показаны без попытки объединить рубли и success rate в произвольный единый score.

Парный расширенный расчёт по каждой задаче даёт среднюю экономию 2,13 вызова, 68 039 токенов и 9,40 руб. на задачу. Соответствующие 95% bootstrap-интервалы равны $[1{,}60; 2{,}67]$ вызова, $[51\,980; 83\,480]$ токенов и $[6{,}75; 12{,}05]$ руб. Эти интервалы относятся к среднему ресурсному эффекту на замороженной смеси страт и добавлены как прозрачный описательный анализ; primary inference по качеству от них не меняется.

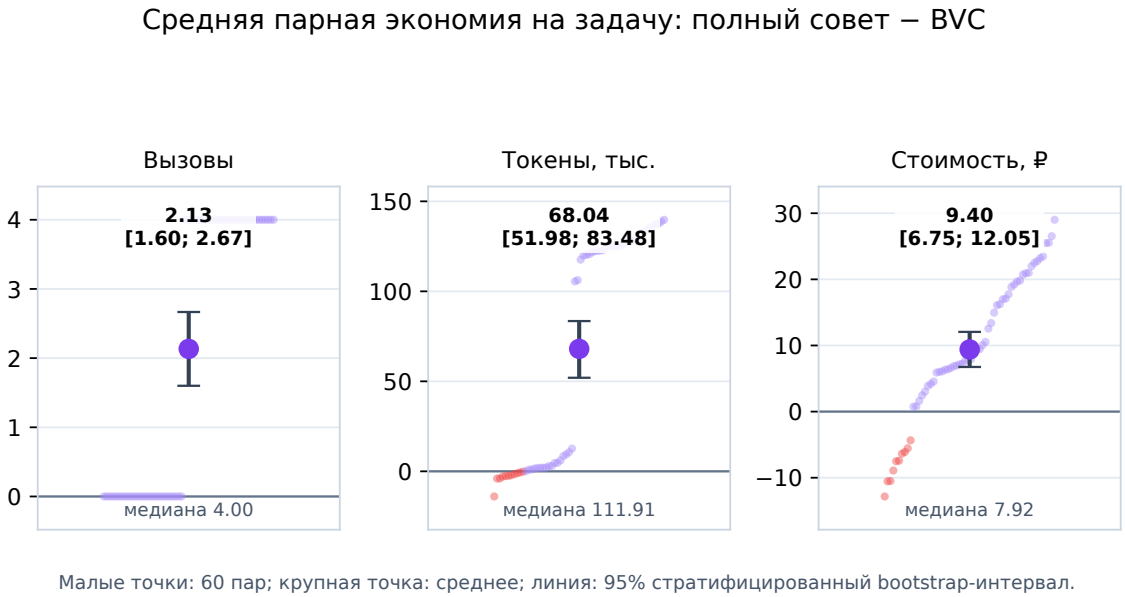


Рис. 7: Средняя парная экономия полного совета относительно BVC; 95% интервалы получены 10 000-кратным bootstrap внутри страт.



Рис. 8: Наблюдаемая стоимость на задачу и функциональный результат.

10. Повторяемость

Temperature = 0 не делает полный pipeline детерминированным: возможны различия провайдера, окружения и генерации. Для 15 задач сравнивались три формальных исхода каждого из direct, одного плана и BVC.

Таблица 9: Функциональный outcome в трёх повторах.

Метод	Повтор 1	Повтор 2	Повтор 3	Все три совпали	Попарное согласие
Direct	0/15	2/15	1/15	13/15	91,1%
Один план	1/15	1/15	2/15	13/15	91,1%
BVC	0/15	2/15	0/15	13/15	91,1%

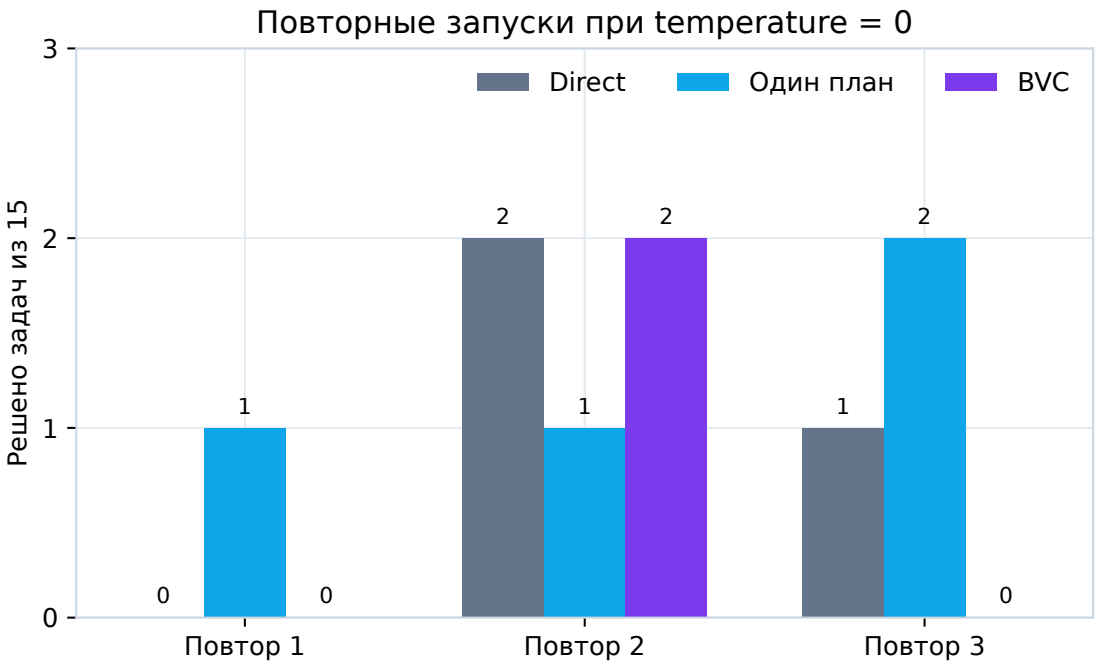


Рис. 9: Число решённых задач в каждом повторе при temperature $= 0$.

Для каждого метода все три бинарных исхода совпали на 86,7% задач, среднее попарное согласие равно 91,1%. Эти значения характеризуют наблюдаемый pipeline и не подменяют primary-оценку на 60 задачах. Majority outcome BVC равен 0/15, одного плана — 2/15; точный парный $p = 0{,}5$.

Этот вспомогательный 15-задачный слой не является прямой оценкой дисперсии 60-задачного contrast и не аннулирует замороженную основную матрицу. Однако колебание числа успехов от 0 до 2 из 15 при неизменном nominal setup показывает, что одиночные различия в одну–две задачи нельзя считать устойчивым свидетельством качества. Для такой оценки нужны как минимум три и более запуска каждого метода на всей сравниваемой выборке либо отдельная модель вероятности успеха на instance.

В repeat-слое транспортное окно также временно увеличивалось до 600 секунд после повторных timeout. Два поздних успешных API-ответа заняли более 180 секунд. Все затронутые задачи официально остались unresolved, поэтому ни один наблюдаемый успех или majority-success не создан расширенным timeout; контрфактический ответ при строгом 180-секундном обрыве неизвестен.

11. Устойчивость к разговорной формулировке

Для тех же 15 instances фактическое требование и кодовый контекст сохранены, но формальная инструкция переформулирована разговорно. Это контролируемая проверка одного paraphrase, а не обобщение на весь класс vibe coding.

Финальный файл разговорных входов имеет отдельный SHA-256; все 15 текстов прошли model-based fidelity check и последующее ручное сопоставление acceptance criteria. В per-row metadata по технической причине осталось `prompt_variant=formal`, однако это поле не участвовало в построении prompt. Сохранённые API-prompts содержат именно разговорный текст. Регенерация ради косметической метки не выполнялась, чтобы не выбирать новый outcome при недоступном provider seed.

Таблица 10: Формальная и разговорная постановки.

Метод	Формально	Разговорно	Согласие по задачам	p_{McNemar}
Direct	0/15	2/15	86,7%	0,5
Один план	1/15	1/15	100,0%	1,0
BVC	0/15	1/15	93,3%	1,0

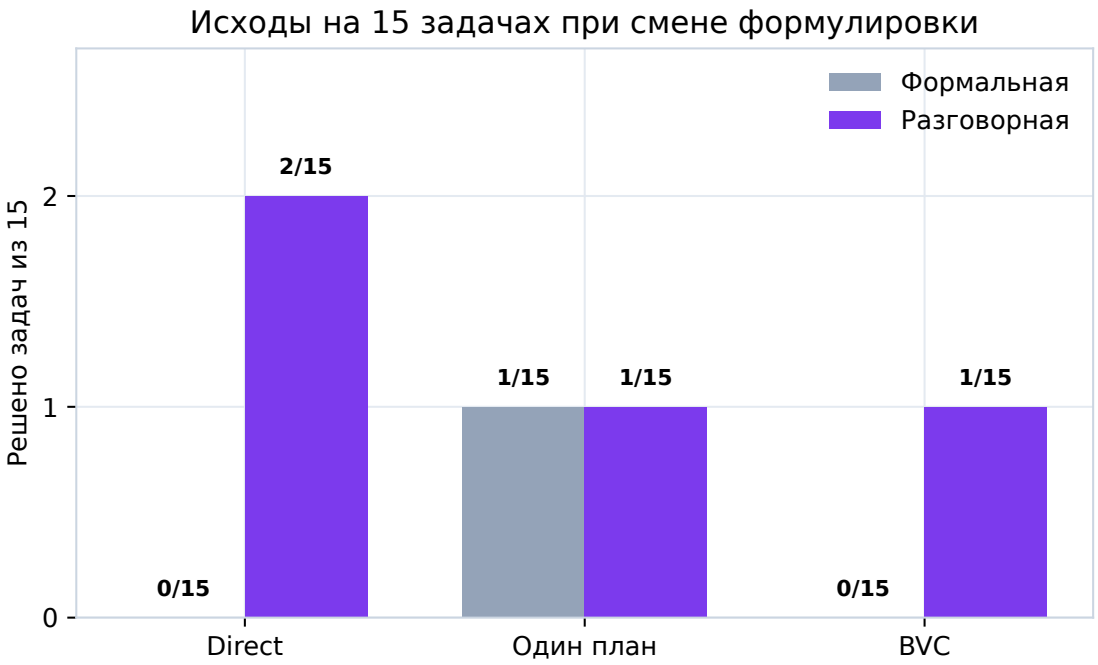


Рис. 10: Исходы методов после контролируемой смены стиля запроса.

BVC дал один успех в разговорной серии и сохранил одинаковый бинарный outcome на 14/15 задачах. Это документированный исход одного paraphrase, но не доказательство общей устойчивости: изменения direct и BVC находятся в пределах колебания 0–2 успеха, уже наблюдавшегося в формальных повторах.

12. Машиночитаемый пакет и доступность результатов

Ключевые входы анализа имеют следующие SHA-256:

Проверяемые источники

Основные 240 строк

`02b8fbcc05512aac41bae0fd4d0013912ef0b2d14436eb74d94f72bfae6892e3`

90 дополнительных повторов

`124e689cdb45cb79a550179c858cfc70a9a612b93d93101dd3abf309f8197220`

45 разговорных запусков

`65c4c6992fbbc5a658828f5c672f4c6016d690997587f2bd6d1ab38cc3817d37`

Замороженный robustness-протокол

`6aa01afcdcebd095b264f58865b5fb6f173bb93bf2b5a136110845eacddf3371`

Сопроводительный пакет содержит исходные JSONL основной, repeat- и разговорной серий, результаты integrity-проверок, frozen protocols, manifests, скрипты анализа и исходники статей. На момент подготовки рукописи пакету не присвоены публичный DOI или постоянный URL. Поэтому приведённые SHA-256 позволяют проверить уже полученную копию, но сами по себе не обеспечивают к ней доступ. Точный статус результата — аудируемый и пересчитываемый из сопроводительного пакета; независимая публичная воспроизводимость потребует публикации архива в постоянном репозитории.

Воспроизводимость здесь означает возможность проверить неизменность входов, повторно выполнить агрегацию и увидеть все строки, включая неуспехи. Она не означает гарантированного воспроизведения тех же модельных токенов внешним провайдером в будущем.

13. Связь с дипломом и практическая ценность

Диплом [13] задаёт продуктовую архитектуру Xynapse и идею ограниченного многоролевого анализа. Настоящий контур добавляет к ней исследовательский слой: замороженную выборку, baselines, официальный evaluator, парную статистику и ресурсный аудит. Поэтому в последующих публикациях диплом корректно цитируется как источник архитектуры и метода, а числовые claims ссылаются на экспериментальные JSONL и анализ.

Контур можно повторно использовать при замене модели. Для корректного сравнения достаточно создать новую матрицу с теми же task IDs и context hashes, сохранить полный набор финальных строк и выполнить тот же анализ. Изменение тарифа не требует повторной генерации: стоимость пересчитывается из трёх сохранённых категорий токенов.

Особенно важна эта воспроизводимость для продукта: при обновлении модели команда может не спорить по отдельным красивым примерам, а провести тот же парный тест и увидеть, изменились ли функциональный outcome, решение gate и цена маршрута. В результате BVC получает накопительную экспериментальную историю, а каждое следующее улучшение становится сопоставимым с предыдущим.

14. Границы вывода

Выборка содержит 60 задач и одну основную модель; интервалы success rate остаются широкими. Не проверен equal-budget best-of-k или self-consistency baseline, поэтому эксперимент изолирует адаптивный gate относительно его собственной неадаптивной версии, но не доказывает оптимальность многоролевого расходования бюджета. Контекст ограничен BM25-выборкой файлов, patch создаётся одним вызовом без инструментов, запуска тестов и обратной связи, тогда как современные coding workflows используют поиск и итерации [9, 11]. Разговорная серия использует один paraphrase и 15 задач. Повторы оценивают бинарный outcome, а не семантическое сходство планов. Алиас модели `latest` и отсутствие provider seed дополнительно ограничивают точное повторение генерации. После проведения эксперимента опубликован аудит, согласно которому SWE-bench Verified больше не подходит для измерения актуальной frontier coding capability из-за остаточных дефектов тестов и контаминации [6]. Поэтому настоящая работа не трактует 8,3% или 11,7% как абсолютную оценку способности модели. Парный дизайн остаётся проверкой четырёх workflows на одной фиксированной матрице и одном evaluator, но вывод ограничен именно этим тестом. Эти границы точно определяют масштаб утверждений и не позволяют смешать описательный сигнал с подтверждённой гипотезой.

Три сопроводительные статьи используют одну и ту же основную матрицу 240 запусков, а не три независимых эксперимента. Они разделяют вопросы: адаптивный бюджет, аудируемый контур и чувствительность к стилю запроса. Совпадение исходных данных между статьями поэтому раскрывается явно и не может интерпретироваться как независимая репликация.

15. Заключение

Главный результат статьи — BVC получил полноценный доказательный контур, а не только архитектурное описание. Создана матрица из 375 официально оценённых строк: 240 основных, 90 дополнительных повторов и 45 разговорных запусков. Целостность всех матриц подтверждена, несовпадения prompt/content hash отсутствуют. Основная гипотеза преимущества BVC над одним планом не поддержана; положительный сигнал против direct равен одной дополнительной решённой задаче и не является статистически значимым; неухудшаемость относительно одного плана также не установлена. Одновременно адаптивная экономия относительно полного совета измерена напрямую. Тем самым проект Xynapse/BVC, исходно описанный в дипломе [13], получил проверяемую экспериментальную основу, пригодную для следующих моделей, выборок и статей. С инженерной точки зрения это сильное преимущество метода: его поведение, цена и границы качества можно не предполагать, а последовательно измерять и улучшать.

Декларации

Финансирование и конфликт интересов отсутствуют. LLM использовались для генерации экспериментальных ответов и языковой редакции; функциональный outcome определён официальным исполняемым

evaluator, а все числовые значения статей вычислены из сохранённых JSON/JSONL. Диплом [13] цитируется как источник архитектуры Xynapse/BVC, но не как источник экспериментальных результатов настоящей работы.

Литература

[1] B. Efron и R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993. DOI: 10.1201/9780429246593.

[2] S. Holm. «A Simple Sequentially Rejective Multiple Test Procedure». В: *Scandinavian Journal of Statistics* 6.2 (1979), с. 65—70. DOI: 10.2307/4615733.

[3] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press и K. Narasimhan. «SWE-bench: Can Language Models Resolve Real-World GitHub Issues?» В: *International Conference on Learning Representations*. 2024. URL: https://arxiv.org/abs/2310.06770.

[4] Q. McNemar. «Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages». В: *Psychometrika* 12.2 (1947), с. 153—157. DOI: 10.1007/BF02295996.

[5] OpenAI. *Introducing SWE-bench Verified*. 13 авг. 2024. URL: https://openai.com/index/introducing-swe-bench-verified/ (дата обр. 28.07.2026).

[6] OpenAI. *Why SWE-bench Verified No Longer Measures Frontier Coding Capabilities*. 23 февр. 2026. URL: https://openai.com/index/why-we-no-longer-evaluate-swe-bench-verified/ (дата обр. 28.07.2026).

[7] S. Robertson и H. Zaragoza. «The Probabilistic Relevance Framework: BM25 and Beyond». В: *Foundations and Trends in Information Retrieval* 3.4 (2009), с. 333—389. DOI: 10.1561/1500000019.

[8] E. B. Wilson. «Probable Inference, the Law of Succession, and Statistical Inference». В: *Journal of the American Statistical Association* 22.158 (1927), с. 209—212. DOI: 10.1080/01621459.1927.10502953.

[9] C. S. Xia, Y. Deng, S. Dunn и L. Zhang. «Agentless: Demystifying LLM-based Software Engineering Agents». В: *arXiv preprint arXiv:2407.01489* (2024). URL: https://arxiv.org/abs/2407.01489.

[10] Yandex Cloud. *Миллион токенов контекста: запускаем DeepSeek V4 Flash в Yandex AI Studio*. 28 мая 2026. URL: https://yandex.cloud/ru/blog/yandex-ai-studio-deepseek-v4-flash (дата обр. 03.08.2026).

[11] J. Yang, C. E. Jimenez, A. Wettig, K. Lieret, S. Yao, K. Narasimhan и O. Press. «SWE-agent: Agent–Computer Interfaces Enable Automated Software Engineering». В: *arXiv preprint arXiv:2405.15793* (2024). URL: https://arxiv.org/abs/2405.15793.

[12] Д. Э. Халилов. *Xynapse: интеллектуальная среда разработки с интегрированным AI-ассистентом*. GitHub. 2026. URL: https://github.com/jabrailkhalil/xynapse (дата обр. 03.08.2026).

[13] Д. Э. Халилов. «Интеллектуальная среда разработки с интегрированным AI-ассистентом для российского рынка». Выпускная квалификационная работа. Москва: Национальный исследовательский университет «Высшая школа экономики», 2026.