Заменяет ли структурное состояние историю диалога у кодового агента?

Terminal-Bench 2.0 · GLM-5.3-flash · 87 задач × 4 условия × 3 прогона = 1044 запуска · **Alex Markov-Buturskiy**

Интерактивная версия и данные
dadadevelopment.github.io/skill.state-showcase

Пик промпта, только состояние	Стоимость, только состояние	Решено задач	С 6 последними ходами
0,47×	2,57×	61% → 35%	59% vs 61%

Вопрос и постановка

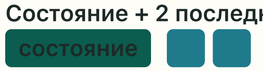
SKILL.state заменяет историю диалога инструкцией, структурным состоянием и текущим наблюдением: контекст перестаёт расти. Дешевле ли от этого решать задачи?

Переносим идею на терминального агента: полная история против состояния плюс 0, 2 или 6 последних ходов. Меряем долю решённых задач и стоимость с учётом кэша.

Что видит модель на каждом шаге

Полная история (контроль)


Только состояние


Состояние + 2 последних хода


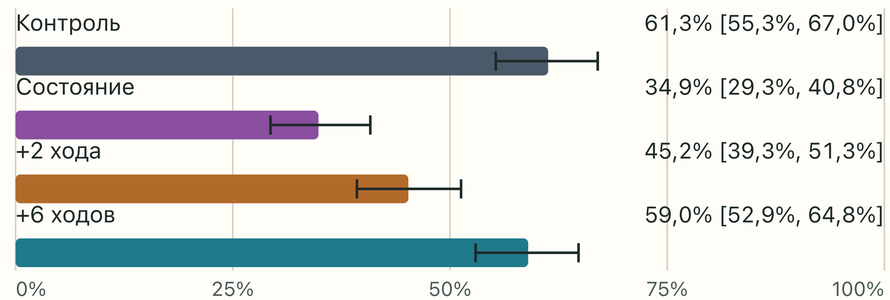
Состояние + 6 последних ходов


Эксперимент

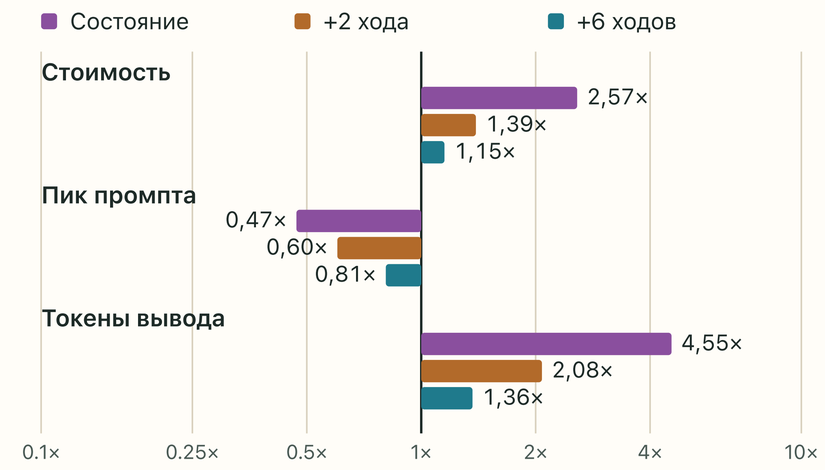
Модель	**zai/glm-5.3-flash**
Бенчмарк	**Terminal-Bench 2.0 · 87 / 89**
Запусков	**1 044 (3×87×4)**
Тариф, USD за млн	**0.15 вход · 0.03 кэш · 0.50 вывод**

Результаты

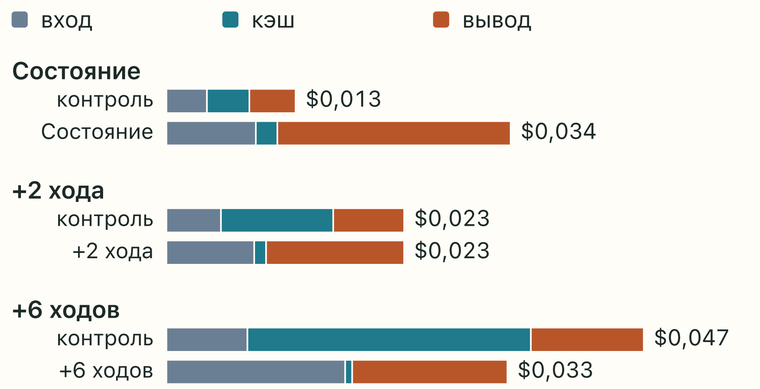
Доля решённых задач с 95% интервалом



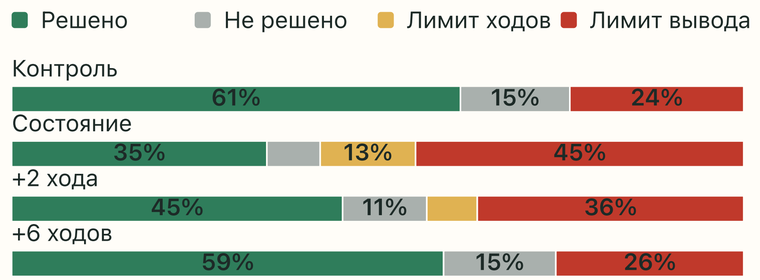
Отношение к контролю (задачи, решённые обоими)



Куда уходят деньги



Что мешает верить



- Одна модель, один бенчмарк, 87 из 89 задач.
- Часть данных потеряна вместе с сервером.

Выводы

1. Меньше токенов не значит дешевле: пик промпта 0,47×, а стоимость 2,57× от контроля на тех же решённых задачах.
2. Шесть последних ходов возвращают качество: 59,0% против 61,3%, разрыв не значим (p = 0,488).
3. Вывод ограничен одной моделью и одним бенчмарком, а 45% запусков «только состояния» упёрлись в лимит вывода.

Исходная идея: Badhe, Tiwari, Chung. SKILL.state. arXiv:2608.26263.

Данные: data/e1-release репозитория skill.state-paper, ревизия 54c0102. Все числа пересчитаны из сырых запусков.