

Научное саммари

GEPA-Curator: отбор задач для самосовершенствующихся агентов

Оригинальное название: GEPA-Curator: Curated Learning for Self Improving Agents

Авторы: Zack Yu, Nicholas Lee, Minjae Lee, Lakshya Agarwal, Luca Manolache, Wonjun Kang, Matei Zaharia, Michael W. Mahoney, Kurt Keutzer, Sehoon Kim, Amir Gholami

Дата публикации: 2026-10-27

Источник: arXiv

Основа анализа: полный текст открытой версии статьи

Направления: Защита нейронных сетей и больших языковых моделей от атак

Первоисточник: <https://arxiv.org/abs/2610.37651v1>

1 Резюме

Статья описывает GEPA-Curator, надстройку над рефлексивными оптимизаторами запросов к модели (GEPA и др.) для агентов на основе больших языковых моделей. Она добавляет два компонента: отбор задач для мини-выборка (отбор задач) и подбор прошлых успешных траекторий того же задания в качестве контекста для Reflector (извлечение ранее успешных траекторий). Авторы сообщают об улучшениях на GoBrowse (WebArena: GitLab, Shopping, Shopping Admin, Reddit), ALFWorld, AppWorld и -bench при двух моделях-оптимизаторах: Gemini 3.1 Pro Preview и Claude Opus 5. Текст, переданный для анализа, обрывается в подписи к Таблице 2, поэтому раздел с результатами по абляциям (раздел 4.3) и приложения в анализ не вошли.

2 Проблема и контекст

По утверждению авторов, в агентных задачах неудача может быть вызвана не слабостью запроса к модели, а внешними факторами: недоступностью инструментов, ограничениями тестовый контур, стохастичностью выполнения, шумом LLM-судьи, неоднозначностью или невыполнимостью задач. В таком случае Reflector может приписать сбой промпу и внести ненужные или вредные изменения. Авторы также указывают, что валидационная фильтрация (как в GEPA и других работах) действует только после генерации мутации, зависит от агрегированных оценок, которые могут быть шумными, и не улучшает качество самой ре-

флексии. Пример в тексте: кнопка покупки временно не работает, и Reflector может сделать вывод о необходимости искать обходные пути.

3 Цель исследования

Показать, что выбор того, какие задачи и какой опыт подаются оптимизатору запросов к модели, влияет на качество оптимизации. Для этого предложить GEPA-Curator, который отбирает задачи с полезным сигналом и подбирает эталонные успешные траектории, не меняя сам оптимизатор.

4 Методы и модели

Оптимизатор формализован как интерфейс «запрос к модели + мини-выборка задач + метаданные → обновлённый запрос к модели, траектории, оценки». GEPA-Curator работает как внешний цикл поверх оптимизатора и хранит историю по каждой задаче (траектории и оценки прошлых итераций). (1) отбор задач: задачи с недостаточной историей приоритетны; задачи пропускаются при устойчивом успехе, повторяющихся сбоях из-за внешних факторов или противоречивых оценках. Пропуск временный, и задачи пересматриваются на каждой эпохе. (2) извлечение ранее успешных траекторий: Curator выбирает одну успешную траекторию той же задачи, отдавая предпочтение успехам с согласованными оценками; при отсутствии надёжного примера он воздерживается. Оба механизма выполняются в одном вызове Curator. Для сравнения используется упрощённый жадный вариант GEPA без Pareto-отбора: одна линия, где каждая мутация становится следующим контрольной точкой. Авторы утверждают, что для их экспериментов разница с Pareto-вариантом мала (по их абляции).

5 Данные и экспериментальная постановка

GoBrowse/WebArena: GitLab (720 задач), Shopping (585), Shopping Admin (750), Reddit (585); разбиение 3:1:1 на train/validation/test; 5000 траектории сбора опыта на сайт и метод. ALFWorld: 150 обучающих и 50 валидационных задач из official train и valid seen, оценка на 134 valid unseen; 2000 траектории сбора опыта. AppWorld: официальные сплиты, 1500 траектории сбора опыта, метрика Task Goal Completion (Test-N и Test-C). -bench: из retail test взято 115 сценариев, пользовательское взаимодействие удалено (схлопнуто в одиночные запросы), разбиение 48/32/35; 800 траектории сбора опыта и валидация каждые 4 итерации. Актор: gpt-oss-120b. мини-выборка: 16. Валидация каждые 16 итераций (кроме -bench). Результаты усреднены по 5 запускам. Сравнение с GEPA и

ACE (Таблица 2), с начальное значение генератора случайных чисел-промптом написанным вручную.

6 Результаты

Результаты авторов (в таблицах, как представлено в тексте). WebArena, Gemini 3.1 Pro: начальное значение генератора случайных чисел 72.4, GEPA 78.7 $\pm$ 0.6, GEPA-Curator 82.1 $\pm$ 1.7 (разница 3.4 п.п. с GEPA). Claude Opus 5: начальное значение генератора случайных чисел 73.1, GEPA 79.5 $\pm$ 1.2, GEPA-Curator 84.5 $\pm$ 1.0 (разница 5.0 п.п.). По доменам разница с GEPA составляет от 0.6 до 7.4 п.п. Авторы отмечают, что прирост над начальное значение генератора случайных чисел у Curator в 1.54 $\times$ (Gemini) и 1.78 $\times$ (Claude) больше, чем у GEPA. Таблица 2, Gemini 3.1 Pro: ALFWorld GEPA 82.4, ACE 90.1, Curator 91.8; AppWorld Test-N 38.0 / 55.7 / 58.7; Test-C 27.8 / 34.5 / 46.2; -bench 75.4 / 80.6 / 84.0. Claude Opus 5: ALFWorld GEPA 95.8, ACE 89.7, Curator 98.4; AppWorld Test-N 52.9 / 61.2 / 68.8; Test-C 32.5 / 33.4 / 46.4; -bench 82.9 / 61.7 / 82.9. Таким образом, Curator не уступает по среднему значению в представленных строках, но на -bench при Claude Opus 5 результат совпадает с GEPA (82.9), а разброс (std) у многих пар перекрывается. Заявление в аннотации «до 20.7% над GEPA» в представленных таблицах напрямую не воспроизводится: максимальная абсолютная разница с GEPA в таблицах около 18.4 п.п. (AppWorld Test-C, Gemini), а разница 21.2 п.п. над ACE соответствует -bench (Claude Opus 5: 82.9 – 61.7). Авторы не уточняют, что означают эти проценты (абсолютные п.п. или относительные величины).

7 Ограничения

Авторы в тексте прямо не выделяют раздел ограничений (он может быть в обрезанной части). Из текста видны следующие ограничения: (а) текст обрывается на Таблице 2, раздел 4.3 с абляциями и обсуждение не представлены, поэтому вклад каждого компонента (отбор задач и поиск успехов) в анализе не подтверждён; (б) использован один актор (gpt-oss-120b) и две модели-оптимизатора; (в) GEPA-Curator построен на упрощённом жадном варианте GEPA, а не на полном GEPA, что усложняет прямое сравнение; (г) -bench модифицирован: пользовательское взаимодействие удалено, что меняет задачу; (д) малые обучающие выборки (например, 48 задач -bench) и разброс результатов; (е) отсутствуют данные о стоимости дополнительных вызовов Curator, хотя он выполняет дополнительную работу; (ж) заявленная «робастность» основана на ограниченном наборе доменов и моделей.

8 Воспроизводимость

Частично воспроизводимо по описанию. Указаны: размеры и разбиения датасетов, число траектории сбора опыта, размер мини-выборка (16), частота валидации (каждые 16 итераций; каждые 4 для -bench), максимальный контекст (131072), число запусков (5), модели (gpt-oss-120b, Gemini 3.1 Pro Preview, Claude Opus 5). Не приведены в тексте: сами начальное значение генератора случайных чисел-промпты (написаны вручную), полный запрос к модели для Curator (сказано, что он в Приложении В.2, которого в тексте нет), детали хранения истории, критерии пропуска задач в числовом виде, код и версии среды. Открытость кода в тексте не упоминается.

9 Связь с тематическими направлениями

Тема связана с самосовершенствованием агентов на основе больших языковых моделей через рефлекссию и оптимизацию запросов к модели, а также с качеством обучающего сигнала. Идея фильтровать шумные задачи и подавать эталонные успешные примеры относится к практическим вопросам надёжной оптимизации агентов. Для прикладной работы важны: независимость от конкретного оптимизатора (как утверждают авторы) и применимость к разным средам (веб, бытовые задачи, API, диалоговые сценарии). Однако в тексте не показано, что такой подход безопасен в смысле устойчивости к злонамеренным данным; тема сопоставлена в метаданных с «защитой нейронных сетей и LLM от атак», но статья этой тематике не посвящена.

10 Что не удалось установить

1) Текст обрезан на Таблице 2, поэтому не проверены остальные результаты и выводы раздела 4.3. 2) Метаданные указывают тему «защита от атак», которая не соответствует содержанию статьи (оптимизация запросов к модели и отбор задач). 3) Цифры «до 20.7% над GEPA» и «21.2% над ACE» в аннотации: 21.2 совпадает с разницей над ACE на -bench (Claude), но 20.7 в представленных таблицах не найдено; способ расчёта не описан. 4) Многие сравнения имеют перекрывающиеся std (например, ALFWorld у GEPA и Curator для Claude; -bench для Claude), поэтому формулировки «consistently outperforms» требуют осторожности. 5) Вывод о том, что curation «материально» увеличивает прирост, опирается на отношение приростов над начальное значение генератора случайных чисел ($1.54\times$ и $1.78\times$), это интерпретация авторов по малому числу доменов. 6) Причинность между отбором задач и качеством рефлексии авторами напрямую не измерена в представленной части: наблюдаемые результаты согласуются с гипотезой, но не доказывают её. 7) Статья по-

мечена «Under Review», версия v1; дата и идентификатор arXiv указаны в метаданных как 2026 год, что стоит проверить по источнику.

Ссылки

Первоисточник: <https://arxiv.org/abs/2610.37651v1>

PDF статьи: <https://arxiv.org/pdf/2610.37651v1>