© Bondarovets S., Okhrimenko T. Method for Enhancing the Robustness of Large Language Models against Adversarial Attacks // *Ukrainian Scientific Journal of Information Security, 2026, vol. 32, issue 2, pp. 60-71. © This is an open access article under the CC BY 4.0 International license.*

DOI: 10.18372/2225-5036.32.21659

МЕТОД ПІДВИЩЕННЯ СТІЙКОСТІ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ДО ЦІЛЕСПРЯМОВАНИХ АТАК

Сергій Бондаровець, Тетяна Охріменко

Національний університет «Київський авіаційний інститут», Київ, Україна



БОНДАРОВЕЦЬ Сергій Сергійович, аспірант
Рік та місце народження: 1995 рік, м. Біла Церква, Україна.
Освіта: Національний авіаційний університет, 2018 рік.
Посада: аспірант Державного університету «Київський авіаційний інститут»
Наукові інтереси: штучний інтелект, інформаційна безпека.
Публікації: 2 наукові статті.
E-mail: 23933@stud.kai.edu.ua
Orcid: 0009-0007-2205-6768



ОХРІМЕНКО Тетяна Олександрівна, к.т.н., ст. дослідник
Рік та місце народження: 1990 рік, м. Вінниця, Україна.
Освіта: Національний авіаційний університет, 2012 рік
Посада: заступник декана з наукової роботи Факультету комп'ютерних наук та технологій КАІ.
Наукові інтереси: інформаційна безпека, реагування на інциденти інформаційної безпеки, квантова криптографія
Публікації: більше 100 наукових публікацій, серед яких монографії, наукові статті, матеріали та тези доповідей на конференціях, патенти та авторські свідоцтва.
E-mail: t.okhrimenko@npp.kai.edu.ua
Orcid: 0000-0001-9036-6556

Анотація. *У статті представлено математично обґрунтований метод підвищення стійкості великих мовних моделей (LLM) до маніпуляцій на рівні вхідних даних, систематизуючи вектори цілеспрямованих атак та механізми їхньої превентивної нейтралізації. Вступна частина окреслює архітектурну вразливість авторегресійних трансформерів – злиття операційних інструкцій та неперевірених даних у єдиному просторі пам'яті, що генерує системні ризики компрометації та формує ядро такої загрози, як ін'єкція запитів. Основна частина дослідження пропонує комплексний апарат, який діє на фундаментальному рівні тензорної механіки. Для запобігання злиттю градієнтів розроблено модуль геометричної проекції, що ортогонально розділяє вбудовування запиту та rag–контексту. Задля усунення ризику вичерпання системної ліквідності пам'яті застосовано 4–бітне асиметричне квантування матриць трансформації. Паралельно впроваджено динамічний адаптер ентропії, що алгоритмічно коригує змінну коефіцієнта штрафу λ на етапі пре–softmax генерації, ліквідуючи ризик надмірного блокування легітимних запитів. Додатково проведений розширений порівняльний аналіз, який узагальнює переваги запропонованої архітектури над існуючими реактивними методами та ресурсоємними тіньовими моделями. Математично доведено, що запропонований зсув логітів робить оптимізаційний градієнт виродженим, гарантуючи експоненційне згасання ймовірності успішної атаки. У висновках підсумовано ключові результати та визначено перспективні напрями для подальших досліджень: розробку формальних гарантій безпеки для мультимодальних середовищ, стандартизацію бенчмарків та розвиток комплексних систем моніторингу.*

Ключові слова: *штучний інтелект; кібербезпека; захист штучного інтелекту; великі мовні моделі; цілеспрямовані атаки.*

Вступ

Сучасні системи штучного інтелекту, зокрема великі мовні моделі (LLM), демонструють експоненційне зростання обчислювальних можливостей. Проте їхнє розгортання у сферах, чутливих до безпеки, супроводжується пропорційним зростанням ризиків маніпуляції. Зловмисники застосовують цілеспрямовані атаки: умисні зміни вхідних даних, що змушують алгоритм генерувати некоректні або небезпечні результати. На відміну від класичних вразливостей програмного забезпечення, ці атаки експлуатують виключно математичну природу нейронних мереж, створюючи семантичні ілюзії для алгоритмів.

Критичною точкою відмови є генеративна поверхня атаки: авторегресійні трансформери є принципово стохастичними та мають структурну вразливість. Ця вразливість визначається наступним чином: операційні інструкції та неперевірені зовнішні дані обробляються спільно в єдиному безперервному просторі пам'яті. Архітектурна нездатність моделі апаратно чи алгоритмічно розмежувати легітимні команди від зловмисних навантажень формує математичне ядро такої вразливості, як ін'єкція запитів. Такі маніпуляції не потребують доступу до вихідного коду чи ваг моделі – вони змушують систему ігнорувати первинні обмеження.

Наслідки успішної експлуатації включають прямі загрози: несанкціонований доступ, витік даних (data exfiltration) та каскадну компрометацію суміжних модулів. Непряма ін'єкція запитів (IPI) через отруєні зовнішні документи дозволяє зловмисникам перехоплювати контроль над процесом генерації автономних агентів, тригеруючи каскадні збої під час виконання [2]. Якщо не застосовувати строгі методи пом'якшення, атаки підривають довіру та неминуче призводять до

© Bondarovets S., Okhrimenko T. Method for Enhancing the Robustness of Large Language Models against Adversarial Attacks // *Ukrainian Scientific Journal of Information Security, 2026, vol. 32, issue 2, pp. 60-71. © This is an open access article under the CC BY 4.0 International license.*

катастрофічних наслідків: серйозні операційні та фінансові збитки, порушення конфіденційності, а також швидке вичерпання ліквідності через вимушену зупинку критичних бізнес-процесів.

Існуючі механізми захисту, такі як змагальне навчання (adversarial training) чи захисна дистиляція, не є універсальними. Головний недолік існуючих рішень полягає у наступному: їхня вразливість до адаптивних атак та експлуатаційні проблеми, такі як надмірне споживання обчислювальних ресурсів, жорсткі обмеження ліквідності при масштабуванні та деградація базових метрик точності. Для побудови відмовостійких систем необхідний порівняльний аналіз та покращення існуючих математичних методів що гарантуватимуть стійкість до збурень без суттєвої втрати корисності моделі. Це вимагає строгої формалізації проблеми в термінах просторової геометрії токенів та ймовірнісних розподілів під час декодування з урахуванням системних ризиків компрометації.

Постановка проблеми

Основна архітектурна вразливість великих мовних моделей полягає у структурному змішуванні операційних інструкцій та неперевірених користувацьких даних у єдиному безперервному просторі пам'яті авторегресійних трансформерів[15]. Ця технічна вада робить системи нездатними внутрішньо розрізняти легітимні системні команди від зловмисних навантажень, що формує математичну основу для такої критичної загрози, як ін'єкція запитів. В умовах найгірших сценаріїв успішна експлуатація цієї вразливості ініціює каскадну компрометацію підключених баз даних та API, що генерує системні ризики та може спровокувати різке виснаження ліквідності компанії через зупинку інфраструктурних процесів.

Аналіз останніх досліджень і публікацій

Галузеві безпекові фреймворки застосовуються для формалізації, операціоналізації та квантифікації загроз.

- **OWASP Top 10 для LLM** визначає категорію LLM01 (ін'єкція запитів) як найбільш критичний вектор загрози, що потребує першочергової апаратної чи алгоритмічної ізоляції[14].
- **MITRE ATLAS** деталізує цілеспрямовані атаки через стандартизовану матрицю, операціоналізуючи ризики через профілі AML.T0054 (LLM Jailbreak) та AML.T0056 (непряма ін'єкція запитів).
- **NIST AI RMF 1.0** вимагає чіткого формулювання заяв про прийнятність ризиків підприємства та імперативно зобов'язує використовувати стандартизовані безпекові бенчмарки (JailbreakBench, AdvGLUE, WildGuardMix) для емпіричного вимірювання відхилень [13].

З математичної точки зору, сучасні white-box атаки (наприклад, Greedy Coordinate Gradient – GCG) формулюють злам як дискретну неопуклу оптимізаційну задачу. Мета алгоритму: знайти такий суфікс **S**, який при конкатенації зі шкідливим запитом x алгоритмічно мінімізує втрати перехресної ентропії L для цільової ствердної відповіді **y**. Даний підхід відхиляє логіти моделі від попередньо налаштованого безпечного домену D^*. Непряма ін'єкція запитів (IPI) функціонує ще небезпечніше: вона приховує шкідливі інструкції у зовнішніх документах, уникаючи прямої взаємодії зловмисника з інтерфейсом [8, 9].

Аналіз останніх досліджень та існуючих методів захисту виявляє експлуатаційні компроміси, що потребують врахування обмежень ліквідності та системних ризиків. У табл. 1 наведено порівняння існуючих методів захисту та основних ризиків їх використання.

Таблиця 1

Порівняльний аналіз методів захисту великих мовних моделей

Метод захисту	Механізм ізоляції	Експлуатаційний ризик
SafeDecoding	обчислює безпекові коригування логітів шляхом застосування штрафного	агресивні налаштування генерують високий ризик надмірного блокування
SelfDefend	використовує дублюючий стек LLM для крос-верифікації латентної генерації	тіньова генерація фактично подвоює обчислювальні витрати та затримку
Геометричний контроль доступу	проектує користувацькі векторні вбудовування в ізоморфні підпростори	жорсткість ізоморфних підпросторів погіршує повноту (recall)
Зовнішні класифікатори (Semalith v1.4)	ультракомпактні мережі (184M параметрів) досягають FPR 0.000 на	зменшена кількість параметрів обмежує семантичну глибину проти складних
StruQ, IPIGuard, SecRAG	використовують детерміновані спрямовані ациклічні графи (DAG) або	підтримка багатозв'язкових графів у реальному часі створює високі витрати
Jailbreak Rapid Response	динамічно доналаштовує шари через генерацію синтетичних варіантів	притаманно реактивна позиція означає, що zero-day jailbreak повинен успішно
Representation Engineering	застосовує PCA до різниці активацій Δh_l для вилучення головного геометричного	припускає чисту лінійну роздільність концептів, що є хибним при сильно

Відповідно, перспективним напрямком є інтеграція топологічних обмежень (за аналогією з графами знань **secrag**) та динамічного коригування ймовірностей на етапі декодування (за аналогією із **safedecoding**). Цей підхід дозволяє керувати простором вбудовувань без модифікації базових ваг. Така оптимізація вимагає математичного вирішення для превентивного блокування наступних загроз.

- ризики компрометації: у випадку непрямої ін'єкції запитів (ipi) шкідливі інструкції конкатенуються з легітимним масивом даних. Без строгих геометричних бар'єрів зловмисник отримує можливість ініціювати каскадний витік даних (data exfiltration). У багатоагентних середовищах це призводить до несанкціонованого підвищення привілеїв (privilege escalation), де автономні агенти виконують шкідливий граф інструкцій замість оригінальних директив.
- математична ізоляція векторів: топологічне маркування простору математично фіксує ваги ребер, запобігаючи злиттю шкідливих і легітимних векторів навіть за умови застосування жадібного координатного градієнта (gcg). Це створює строгу ізоморфну ізоляцію, стійку до оптимізаційних алгоритмів.
- обчислювальна ліквідність: замість звернення до багатопараметричних класифікаторів (llama guard 3), оптимізація алгоритму безпосередньо на рівні ймовірнісного розподілу токенів (safedecoding) зводить затримку до мінімуму. Штрафні коефіцієнти детерміновано та математично блокують відхилення генерації до шкідливого домену D^*.

Обраний напрямок формує бар'єр проти цілеспрямованих маніпуляцій, превентивно блокуючи ризики компрометації до моменту генерації першого небезпечного токена, водночас повністю задовольняючи жорсткі обмеження операційної ліквідності.

Мета даного дослідження полягає у покращенні та математичному обґрунтуванні методу підвищення стійкості великих мовних моделей до цілеспрямованих атак, з фокусом на такий вектор, як ін'єкція запитів. Цей метод має гарантувати стабільність систем генерації з доповненням пошуком (rag) та архітектур відкритої ваги (open–weights) в умовах високодинамічних загроз без необхідності повного перенавчання базової моделі.

Основна частина дослідження

Опис математичного апарату запропонованого покращеного методу підвищення стійкості LLM

Сучасні великі мовні моделі (LLM), особливо в рамках архітектур відкритої ваги (open–weights) та систем генерації з доповненням пошуком (rag), є вразливими до маніпуляцій на рівні вхідних даних [15, 21].

Існуючі підходи до захисту демонструють суттєві концептуальні та експлуатаційні недоліки: вони або покладаються на реактивне блокування, що унеможливлює захист від атак нульового дня, або використовують ресурсоємні паралельні тіньові моделі, що подвоюють обчислювальну складність та порушують обмеження операційної ліквідності систем реального часу.

Для подолання цих обмежень виникає необхідність покращення існуючих математичних моделей. Вибір запропонованої математичної моделі базується на наступній гіпотезі: замість того, щоб намагатися виявити шкідливий намір постфактум за допомогою зовнішніх класифікаторів, необхідно математично унеможливити саме злиття легітимного запиту та шкідливого контексту на рівні векторного простору моделі.

Саме тому в основу нашого методу покладено синтез двох інноваційних підходів: топологічної ізоляції векторів (геометричного розділення просторів вбудовувань) та динамічного коригування логітів на етапі пре–softmax декодування. Такий апарат дозволяє створити суворий детермінований бар'єр проти цілеспрямованих маніпуляцій, превентивно нейтралізуючи системні ризики компрометації до моменту повної генерації першого небезпечного токена.

В основі авторегресійної генерації в системах rag лежить обробка двох ключових інформаційних масивів, які формалізуються через базові тензори x та c. математична сутність цих параметрів визначається наступним чином: x та c є матрицями вбудовувань токенів (token embeddings), що належать до просторів $r^{l_x \times d}$ та $r^{l_c \times d}$.

Відповідно, l_x позначає довжину токенізованої послідовності системного або користувацького запиту, l_c відповідає за довжину токенізованої послідовності зовнішнього контексту, а d репрезентує загальну вимірність латентного простору моделі.

Тензор x містить довірений, легітимний запит, тоді як тензор c формується із зовнішнього тексту, отриманого через пошукову видачу. Головна загроза (непряма ін'єкція запитів) інкапсулюється виключно в межах матриці c. Під час стандартної операції генерації алгоритм атаки намагається перехопити керування моделлю через базовий механізм багатонаправленої уваги (multi–head self–attention), змішуючи градієнти c із градієнтами x. Для усунення цієї структурної вади пропонується ввести матриці трансформації w_c та w_x. Математична сутність трансформації полягає в наступному: це лінійні проекції $\in r^{d \times d_{iso}}$, де d_{iso} позначає звужену (стиснуту) вимірність ізольованого підпростору.

Розглянемо контекст захисту в цьому випадку: ці матриці примусово проектують оригінальні багатовимірні вектори токенів у заздалегідь визначені неперетинні площини. завдяки цьому створюється нездоланний фізичний та геометричний

бар'єр між потоками даних. матриці контексту c та запиту x геометрично проектуються у взаємно ізольовані ізоморфні підпростори, унеможливлюючи їхнє семантичне злиття на глибоких шарах нейромережі.

Для того, щоб геометрична ізоляція була математично гарантованою, вводиться умова суворої ортогональності. ця умова вимагає мінімізації норми Фробеніуса для скалярного добутку спроектованих матриць під час розрахунку багатонаправленої уваги. Формалізовано це рівняння (ортогональний штраф) виглядає так:

$$|| (cw_c)(xw_x)^t ||_f^2 \approx 0$$

У даному випадку оптимізація спрямована на зведення до нуля квадрата норми Фробеніуса для скалярного добутку між геометрично спроектованою матрицею зовнішнього контексту (cw_c) та геометрично спроектованою (і транспонованою) матрицею легітимного запиту $(xw_x)^t$. Нульовий скалярний добуток означає ідеальну ортогональність, тобто повну лінійну незалежність векторних підпросторів. Векторний підпростір неперевіреного контексту стає строго перпендикулярним до векторного підпростору системного запиту. Захисна дія цього рівняння полягає у блокуванні перенесення градієнтів шкідливого контексту на легітимний запит. Вплив отруєного контексту на системний запит при розрахунку уваги (attention) математично дорівнює нулю, що робить неможливим перехоплення контекстного вікна алгоритмами оптимізації атаки (такими як gcg).

Після того, як модуль геометричної проекції успішно забезпечує ортогональну сепарацію на рівні векторів уваги, наступний етап захисту розгортається безпосередньо під час генерації тексту (на етапі пре–softmax декодування). Основним завданням цього етапу є нейтралізація залишкових аномалій, які могли б ініціювати генерацію шкідливого контенту, якщо витончено сконструйована ін'єкція запитів зможе частково обійти первинні фільтри.

Для цього вводиться механізм топологічної оцінки прихованих станів та динамічної модифікації логітів: на кожному кроці генерації t нейромережа формує прихований стан, який позначається як h_t. Це вектор $\in r^d$, що інкапсулює багатовимірну семантику всього попереднього тексту (включаючи граматику, тональність та базові наміри запиту). багатовимірність означає, що цей числовий масив містить тисячі параметрів (наприклад, 4096 вимірів у сучасних архітектурах), які утворюють повне «розуміння» контексту моделлю.

Прихований стан h_t слугує динамічною координатою для безперервного обчислення просторової близькості поточного процесу генерації до зони відомих атак. Для визначення цієї близькості введемо поняття шкідливого домену d^* через обчислення його еталонного центру.

Рівняння центру мас:

$$c^* = \frac{1}{|d^*|}\sum_{i=1}^{|d^*|} h_{mal}^{(i)}$$

c^* розраховує математичний центр мас (центроїд) векторів відомих атак $h_{mal'}^{(i)}$, акумульованих у базі заборонених відповідей d^*. Обчислення цього центроїда дозволяє звести задачу виявлення загрози до вимірювання кутової відстані.

Далі обчислюється функція топологічного штрафу $p(h_t, d^*)$, що спирається на косинусну відстань.

Рівняння скалярної функції, що вимірює векторну близькість поточної генерації h_t до шкідливого центроїда c^*:

$$p(h_t, d^*) = max(0, cos\ cos\ (h_t, c^*) - \tau)$$

Функція max із порогом τ створює так звану «мертву зону». Якщо поточне відхилення вектора генерації від легітимного курсу є меншим за емпірично встановлений поріг τ, система ігнорує цю незначну флуктуацію. Це гарантує, що обчислювальний опір та додаткова затримка (latency) для абсолютно безпечних запитів дорівнюють нулю, зберігаючи високу пропускну здатність моделі.

Для застосування топологічного штрафу до розподілу ймовірностей вводиться змінна коефіцієнта штрафу λ – скалярний мультиплікатор сили зсуву логітів. Використання статичного значення λ неминуче провокує ризик надмірного блокування легітимних запитів (over–refusal), оскільки модель може помилково класифікувати нейтральні токени, що мають частковий семантичний збіг із забороненим словником. Тому обов’язковою умовою оптимізації є те, що значення коефіцієнта штрафу повинно динамічно адаптуватися до ентропії поточного вікна контексту.

Інформаційна ентропія $h(p(y_t))$ виступає як міра розсіювання ймовірностей генерації. Ентропія обчислюється на основі поточного розподілу ймовірностей $p(y_t)$. Цей параметр є тригером активації захисту. аномальне та різке зростання ентропії свідчить про конфлікт прихованих станів, коли модель вагається між виконанням системної інструкції та нав'язаної шкідливої команди (що є типовим маркером того, що ін'єкція запитів намагається перехопити керування).

Рівняння динамічної адаптації коефіцієнта виглядає наступним чином:

$$\lambda_t = \frac{\lambda_{max}}{1 + exp(\beta \cdot (h(p(y_{t-1})) - \mu))}$$

Логістична (сигмоїдна) функція математично пов'язує силу штрафу λ_t з ентропією попереднього кроку генерації $h(p(y_{t-1})$. β відповідає за крутизну функції (швидкість реакції), а μ встановлює базове математичне очікування нормальної ентропії.

Інтенсивність захисту масштабується строго пропорційно до системної невпевненості. Якщо ентропія зростає (модель фіксує конфлікт інструкцій), коефіцієнт λ_t експоненційно наближається до λ_{max}, активуючи жорстке придушення. Якщо ентропія низька (впевнена легітимна генерація), λ_t прямує до мінімуму, запобігаючи деградації та блокуванню безпечного тексту.

Остаточний вплив захисного механізму відбувається на рівні векторів логітів z_t та $\tilde{z}_t$ – це вектори $\in r^v$ (де v – загальний об'єм словника нейромережі), що репрезентують ненормалізовані ймовірності появи кожного можливого токена. z_t – це оригінальний вихід останнього лінійного шару (linear projection layer). $\tilde{z}_t$ – це скоригований вектор, який подається на вхід функції softmax для фінального вибору токена.

Для точкового впливу вводиться бінарна маска m_{mask} – булевий вектор $\in \{0,1\}^v$, цей математичний фільтр гарантує, що штраф застосовується виключно до токенів, які належать до емпірично визначеного небезпечного словника (наприклад, слів, що формують команди несанкціонованого доступу чи розкриття системних промптів). маска зберігає пропускну здатність нейтральних та легітимних генерацій.

Скоригований вектор логітів розраховується за фінальним рівнянням:

$$\tilde{z}_t = z_t - \lambda_t \cdot p(h_t, d^*) \cdot m_{mask}$$

Від оригінальних ненормалізованих ймовірностей z_t віднімається добуток динамічного коефіцієнта, топологічної косинусної близькості до загрози та бінарної маски. Математичним наслідком цього віднімання є те, що ймовірності всіх небезпечних токенів примусово зсуваються у глибокий хвіст розподілу. навіть якщо змагальний алгоритм намагається максимізувати вагу шкідливого токена, ця модифікація динамічно нівелює його зусилля, гарантуючи безпечний вибір при генерації.

Наведений апарат динамічного коригування створює підґрунтя для розрахунку теоретичної ефективності всього методу, яка доводиться через аналіз оптимізаційних втрат та виродження градієнтів.

Ефективність розробленого методу захисту математично доводиться через аналіз оптимізаційних втрат алгоритмів, що генерують змагальні атаки (наприклад, greedy coordinate gradient – gcg). У стандартному вразливому середовищі цільові функції оптимізації (cross–entropy) позначаються як L_{adv} та L_{orig}.

Дані показники репрезентують міри алгоритмічної розбіжності між поточною генерацією моделі та цільовим текстом. L_{orig} показує втрати для легітимної відповіді, тоді як L_{adv} відповідає за втрати для шкідливої генерації.

Атакуючий алгоритм ітеративно підбирає змагальний суфікс S, щоб мінімізувати L_{adv} і змусити модель видати цільову шкідливу відповідь y_{mal}. Інтеграція розробленого методу перетворює L_{adv} на математично негладку функцію, унеможливлюючи її подальшу спрямовану оптимізацію.

Очікувана функція втрат для шкідливої відповіді після застосування геометричної ізоляції та динамічного штрафування модифікується наступним чином:

$$E[L_{adv}(S \oplus X, C, y_{mal})] \geq L_{orig} + \gamma||||_f^2 + \lambda p(h_t, d^*)$$

У цьому рівнянні вводиться змінна запасу стійкості γ – константа мінімально допустимої ортогональної відстані між просторами. Вона діє як математичний гарантійний буфер проти стохастичного шуму нейромережі. оскільки під час інференсу флуктуації обчислень можуть випадково зблизити площини зовнішнього контексту та системного запиту, γ діє як жорсткий запобіжник, що унеможливлює їхню перехресну активацію.

Головним наслідком модифікованого рівняння втрат є вплив на градієнт $\nabla_S L_{adv}$ – це вектор часткових похідних $\frac{\partial L_{adv}}{\partial S_i}$ по параметрах змагального суфікса S, які генерує атакуючий алгоритм.

Оскільки оптимізація gcg вимагає безперервного та диференційованого латентного простору, динамічний зсув логітів та топологічна ізоляція розривають цей оптимізаційний шлях. Вектор похідних стає виродженим, тобто $\nabla_S L_{adv} \approx 0$.

Описаний підхід математично позбавляє атакуючий алгоритм напрямку для подальшого градієнтного спуску, що робить цілеспрямований підбір шкідливих токенів неможливим.

Обчислювальна складність

Інтеграція будь–якого механізму захисту безпосередньо впливає на апаратну пропускну здатність. Базова обчислювальна складність механізму багатонаправленої уваги (multi–head attention) становить[11]:

$$O((L_x + L_c)^2 \cdot d).$$

Ізольована складність з інтегрованими матрицями проекцій розраховується як

$$O(L_x^2 d_{iso} + L_c^2 d_{iso} + (L_x + L_c) d \cdot d_{iso})$$

Контекст захисту розкриває найгірші сценарії (worst–case scenarios): при квадратичному зростанні довжини отриманого пошукового контексту L_c (наприклад, у системах rag з великими масивами документів) системна затримка зростає на константний фактор $O(d^2)$, що генерує ризик вичерпання системної пам'яті, що може призвести до відмови в обслуговуванні легітимних користувачів.

Для вирішення цієї апаратної проблеми та запобігання системним відмовам пропонується механізм 4–бітного асиметричного квантування матриць трансформації:

$$\widehat{W}_c = quantize(W_c, 4 - bit)$$

Дана апаратна операція стискає матриці, експоненційно знижуючи споживання системної пам'яті. У такому випадку очікувана втрата математичної точності під час стиснення обов'язково компенсується пропорційним збільшенням параметра γ, що гарантує збереження ортогонального бар'єра при мінімальних витратах пам'яті.

Фінальна ефективність методу оцінюється через показник ASR (attack success rate) – це ймовірнісна міра того, що згенерована послідовність належатиме до шкідливого домену D^* за наявності легітимного запиту X та цілеспрямовоно модифікованого контексту C_{adv}. Формалізується як:

$$ASR = P(y_{1:T} \in D^* | X, C_{adv})$$

Завдяки виродженню градієнта та розриву оптимізаційного шляху, для розрахунку верхньої межі цієї ймовірності застосовується нерівність Маркова. Якщо оптимізатор не здатний мінімізувати L_{adv} нижче безпечного порогу, ASR отримує наступну математичну оцінку:

$$ASR \leq exp\left(-\frac{\gamma||||_f^2 + \lambda p}{\sigma^2}\right),$$

де σ^2 репрезентує дисперсію стохастичного шуму при генерації.

Дана експоненційна залежність математично доводить, що ймовірність успіху такої загрози, як ін'єкція запитів, експоненційно спадає та наближається до нуля зі зростанням λ та γ.

Сформований та теоретично обгрунтований апарат превентивної протидії забезпечує повний контроль над латентним простором авторегресійних трансформерів без втручання в їхні базові ваги. Поєднання строгих топологічних обмежень, геометричної ізоляції векторів rag та динамічного згладжування декодування формує універсальний підхід, здатний працювати в умовах обмежених обчислювальних ресурсів.

Архітектура реалізації запропонованого методу

Концептуальна архітектура запропонованого методу є логічним інженерним втіленням попередньо розробленого математичного апарату. Вона складається з чотирьох послідовних модулів, інтегрованих у конвеєр інференсу open–weights моделей без необхідності перенавчання базової архітектури. Архітектурна стратегія полягає у наступному: створення детермінованого, прозорого та обчислювально легкого механізму, що функціонує як додатковий шар абстракції між блоками багатонаправленої уваги (multi–head attention) та фінальним лінійним класифікатором (lm_head) авторегресійних трансформерів. Ця ешелонована структура дозволяє превентивно ідентифікувати та блокувати цілеспрямовані маніпуляції, зокрема такий вектор, як ін'єкція запитів.

На відміну від монолітних підходів, що вимагають дороговартісного тонкого налаштування (fine–tuning) мільярдів параметрів, модульна природа запропонованої архітектури спирається на маніпуляції виключно у латентному просторі. Загальна структура функціонує за наступним алгоритмом: послідовне перехоплення вхідних ембедингів, їхня геометрична трансформація, обчислення топологічної близькості до кластерів атак, динамічний розрахунок ентропії та математична модифікація фінальних логітів. Такий конвеєр гарантує, що оптимізаційні алгоритми зловмисників (наприклад, greedy coordinate gradient) стикаються зі штучно створеним градієнтним розривом, тоді як легітимна маршрутизація токенів залишається непорушною.

Модуль геометричної проекції

Першим та найважливішим етапом захисного конвеєра є модуль геометричної проекції. Його операційна логіка безпосередньо спирається на введену в математичному апараті сувору умову ортогональної сепарації. Модуль перехоплює вхідні токени запиту користувача x та отриманого контексту rag c. На цьому етапі матриці вхідних вбудовувань $r^{l_x \times d}$ та $r^{l_c \times d}$ проходять через спеціалізований механізм сепарації.

Наступним кроком відбувається ортогональна трансформація їхніх вбудовувань у просторово розділені ізоморфні підпростори за допомогою матриць трансформації w_c та $w_x \in r^{d \times d_{iso}}$.

Матриці трансформації повинні залишатися лінійно незалежними для унеможливлення злиття градієнтів шкідливого контексту з системними інструкціями. З інженерної точки зору це реалізується шляхом множення вхідних тензорів на оптимізовані матриці w_c та w_x, що стискають вимірність простору. Математичний контролер модуля постійно відстежує умову ортогональності, яка вимагає мінімізації норми фробеніуса для їхнього

скалярного добутку під час розрахунку багатонаправленої уваги:

$$|| (cw_c)(xw_x)^t ||_f^2 \approx 0$$

Це створює жорсткий фізичний бар'єр між потоками даних: якщо ін'єкція запитів прихована в зовнішньому документі, її математичний вектор стає перпендикулярним до вектора системного запиту, зводячи вплив шкідливого градієнта $\nabla_S L_{adv}$ до нуля.

Аналізатор топологічної відстані

Після того як вхідні масиви пройшли геометричну сепарацію і були оброблені прихованими шарами трансформера, ініціюється робота другого архітектурного блоку. Аналізатор топологічної відстані функціонує на етапі пре–softmax генерації, де він обчислює просторову евклідову (або косинусну) відстань між поточним вектором логітів та емпірично визначеним кластером шкідливих інструкцій d^*.

Архітектурно цей кластер представлений як центроїд шкідливих векторів, який акумулює багатовимірну семантику тисяч відомих змагальних суфіксів. на кожному кроці генерації t нейромережа формує прихований стан h_t, що репрезентує вектор поточної семантики. завдання аналізатора зводиться до обчислення функції топологічного штрафу $p(h_t, d^*)$ що обчислює косинусну близькість h_t до центроїда шкідливих векторів.

Ця функція використовує математичну формулу $max(0,cos\ cos\ (h_t, c^*) - \tau)$, де поріг τ формує мертву зону. Якщо косинусна близькість поточного стану генерації до бази атак є меншою за τ, модуль залишається у пасивному стані, ігноруючи незначні флуктуації. Проте, якщо змагальний алгоритм намагається змістити логіти моделі від безпечного домену d^*, відстань скорочується, і аналізатор топологічної відстані миттєво генерує ненульовий скалярний сигнал p. Цей сигнал передається наступним модулям конвеєра як індикатор того, що поточний вектор $\tilde{z}_t$ містить високий ризик компрометації.

Відсутність складних класифікаторів у цьому вузлі (наприклад, багатопараметричних нейромереж) дозволяє виконувати розрахунок евклідової та косинусної метрик зі швидкістю, що майже дорівнює нулю мілісекунд, підтримуючи високу продуктивність системи.

Динамічний адаптер ентропії

Після завершення ортогональної сепарації та розрахунку векторної близькості до зони загрози, архітектурний конвеєр передає керування третьому модулю – динамічному адаптеру ентропії. Базова проблема статичних захисних механізмів полягає в їхній лінійній жорсткості: якщо штрафний коефіцієнт є незмінним, система неминуче деградує. Якщо коефіцієнт налаштовано занадто агресивно та не адаптовано до контексту, виникає ризик надмірного блокування легітимних запитів (over–refusal). Для вирішення цієї проблеми динамічний адаптер ентропії в реальному часі алгоритмічно коригує змінну коефіцієнта штрафу λ на основі дисперсії ймовірностей розподілу наступного токена.

Модуль функціонує як стохастичний балансувальник. Умовою оптимізації є те, що значення λ має динамічно адаптуватися до ентропії контексту $h(p(y_t))$. Логіка процесу полягає у безперервному моніторингу інформаційної ентропії $h(p(y_t))$, яка виступає математичною мірою розсіювання ймовірностей генерації.

Під час нормальної роботи (коли легітимний запит опрацьовується без втручання зловмисників) модель демонструє впевненість у передбаченні наступного токена, що відображається у низькій ентропії. Проте, якщо зловмисник ініціює атаку (наприклад, коли прихована ін'єкція запитів вступає в конфлікт із системним промптом), у прихованих станах h_t виникає семантичний конфлікт, який призводить до аномального зростання ентропії, оскільки ймовірнісний розподіл розсіюється між легітимними та шкідливими токенами.

Адаптер реєструє цю флуктуацію та розраховує нове значення коефіцієнта за логістичною функцією:

$$\lambda_t = \frac{\lambda_{max}}{1 + exp(\beta \cdot (h(p(y_{t-1})) - \mu))}$$

У цій ланці логістична функція математично пов'язує силу штрафу з ентропією попереднього кроку. Дія модуля масштабується пропорційно до системної невпевненості: чим вища ентропія (сильніший конфлікт), тим ближче значення λ_t піднімається до свого максимуму λ_{max}, забезпечуючи агресивне придушення атаки.

Якщо ж генерація є стабільною, значення λ_t асимптотично наближається до нуля, повністю нейтралізуючи ризик надмірного блокування легітимних запитів (over–refusal) та гарантуючи високу пропускну здатність безпечного тексту.

Модуль застосування штрафу

завершальним етапом захисного конвеєра є модуль застосування штрафу, який фізично імплементує розраховані обмеження безпосередньо у вихідний тензор нейромережі. цей компонент є термінальною точкою, де математична теорія трансформується у захисну дію на рівні інференсу. модуль застосування штрафу застосовує нелінійний логіт–штраф до ймовірнісного розподілу, що математично блокує генерацію токенів, які тяжіють до небезпечного домену d^*.

Архітектурно цей процес інтегрується перед фінальною активаційною функцією softmax.

Результат останнього лінійного шару формалізується як вектор $\tilde{z}_t \in r^v$.

Для того, щоб санкції застосовувалися виключно до семантично релевантних загрозі напрямків, модуль використовує спеціалізований фільтр. Скоригований вектор логітів $\tilde{z}_t$ розраховується як:

$$\tilde{z}_t = z_t - \lambda_t \cdot p(h_t, d^*) \cdot m_{mask},$$

де λ_t – змінна коефіцієнтів штрафу, а m_{mask} - бінарна маска проекції, що є булевим вектором, що виконує роль селективного бар'єра. Вона гарантує, що розрахована вага штрафу буде віднята виключно від тих логітів, які відповідають токенам із заздалегідь визначеного небезпечного словника (наприклад, системні команди ОС, ідентифікатори обходу обмежень або деструктивні директиви). Для всіх інших, нейтральних токенів, маска дорівнює нулю, що дозволяє зберегти їхні оригінальні ймовірності недоторканими.

Взаємодія всіх чотирьох модулів формує єдину архітектуру, яка ефективно руйнує математичні передумови для успішних gradient–based атак. Ефективність доводиться через зближення втрат оптимізатора greedy coordinate gradient (gcg). Алгоритм gcg та його аналоги покладаються на неперервність латентного простору для мінімізації функції втрат шкідливої генерації L_{adv}.

Проте, очікувана функція втрат для шкідливої відповіді модифікується:

$$E[L_{adv}(S \oplus X, C, y_{mal})] \geq L_{orig} + \gamma|||_f^2 + \lambda p(h_t, d^*),$$

де γ – змінна запасу стійкості. Введення динамічного нелінійного віднімання на рівні логітів створює штучні розриви (дисконтинуїтети) в оптимізаційному просторі. Зсув логітів та ізоляція роблять градієнт $\nabla_S L_{adv}$ виродженим, розриваючи шлях оптимізації. Коли градієнт стає виродженим (наближається до нуля), атакуючий алгоритм втрачає вектор напрямку: він більше не здатний визначити, яка комбінація вхідних токенів змагального суфікса S наблизить модель до генерації шкідливого домену d^*.

Даний підхід призводить до того, що ймовірність успішної атаки (attack success rate) зазнає математично доведеного колапсу. Експоненційне спадання ймовірності гарантує, що навіть масивні обчислювальні витрати з боку зловмисника не призведуть до успішного підбору токенів, що обходять систему захисту.

Порівняльний аналіз існуючих методів

Згідно з актуальними стандартами безпеки (такими як NIST AI RMF та MITRE ATLAS), ізольовані захисні механізми чи прості промпт–фільтри не здатні самостійно нейтралізувати всі вектори атак, особливо коли зловмисники застосовують алгоритми безперервної градієнтної оптимізації. У таблиці 2 наведені результати порівняння існуючих методів підвищення стійкості LLM і запропонованого нами.

Таблиця 2

Порівняльний аналіз існуючих методів підвищення стійкості LLM до цілеспрямованих атак

Метод (джерело)	Архітектурні переваги	Недоліки та системні ризики	Порівняльний аналіз із запропонованим методом
smoothllm	пом'якшує джейлбрейки, використовуючи крихкість змагальної оптимізації через випадкові збурення вхідних символів.	генерує системний ризик деградації номінальної продуктивності, оскільки важкі збурення руйнують семантику.	наш метод не застосовує стохастичні збурення тексту; натомість ми детерміновано ізолюємо векторні простори, зберігаючи 100% оригінальної семантики.
struq	розділяє промпти та дані на два канали, використовуючи безпековий фронтенд–протокол роздільників [5].	залишається частково вразливим до високоскладних атак безперервної оптимізації градієнтів, що здатні долати токен–розділювачі.	замість введення додаткових токенів–роздільників (які оптимізатор може обійти), наша архітектура математично блокує злиття градієнтів на рівні вбудовувань.
advprompter	архітектура автоматизованого red–teaming, що генерує змагальні суфікси за секунди зі збереженням низької перплексії [16].	інструмент подвійного призначення: ставить під сумнів фільтри, які покладаються на високу перплексію як маркер атаки.	запропонований модуль динамічної ентропії реагує на розсіювання ймовірностей, а не на перплексію тексту, що дозволяє блокувати навіть такі згладжені атаки.
інженерія представлень (representation engineering)	маніпулює високорівневими когнітивними явищами всередині мережі, застосовуючи PCA до різниці активацій Δh_l [20]	сувора умова: припускає чисту лінійну роздільність концептів, що є хибним при глибоко заплутаних перекриттях ознак.	наша функція топологічної відстані p діє на фінальні приховані стани через косинусну близькість, не вимагаючи ідеальної лінійної сепарації (PCA) всіх концептів.

Продовження табл. 2

llama guard 3 vision	багатомодальний класифікатор безпеки, доналаштований для захисту вхідних і вихідних даних візуального міркування [1].	інтенсивні багатомодальні матричні обчислення; gcg обходить класифікацію промптів у 72% випадків.	ми уникаємо ресурсоємних проходів через окремі класифікатори, виконуючи швидкий розрахунок логіт–штрафу безпосередньо в конвеєрі базової моделі.
semalith v1.4	ультракомпактний бар'єр (184m) для агентних середовищ із нульовим показником fpr (false positive rate) на безпечних запитах [1].	радикально зменшена кількість параметрів критично обмежує семантичну глибину моделі проти складних загальних загроз.	наш метод використовує всю семантичну глибину базової llm, коригуючи лише її фінальний ймовірнісний вихід, а не покладаючись на усічену проксі–модель.
jailbreak rapid response	динамічно доналаштовує шари через генерацію синтетичних варіантів постфактум спостережуваної атаки [10].	притаманно реактивна позиція: загроза нульового дня (zero–day) повинна успішно пробити систему щонайменше один раз.	розроблений геометричний бар'єр є суто превентивним; він нейтралізує градієнти ще до того, як атака нульового дня (zero–day) зможе реалізуватися.
паралельні тіньові llm (selfdefend)	використовує дублюючий стек llm для крос–верифікації латентної генерації в реальному часі [18].	провокує ризик виснаження ліквідності: тіньова генерація фактично подвоює обчислювальні витрати та затримку (latency).	запропонований метод діє на рівні пре–softmax логітів з обчислювальною складністю $\mathcal{O}(d_{iso})$, що повністю усуває необхідність паралельного інференсу.
геометричний контроль доступу (geometric access control)	замінює текстову фільтрацію геометричним захистом, проектуючи вбудовування в ізоморфні підпростори[3, 10].	жорсткість фіксованих ізоморфних підпросторів здатна погіршувати повноту (recall) багатопрофільних запитів.	ми застосовуємо гнучкий геометричний поділ та 4–бітне квантування для векторів rag–контексту, зберігаючи загальну повноту пам'яті моделі.
динамічна модифікація декодування (safedecoding)	коригує логіти, застосовуючи штрафний коефіцієнт до токенів, що наближаються до небезпечного домену d^*[17]	агресивні статичні налаштування генерують високий ризик надмірного блокування легітимних запитів (over–refusal).	наш модуль динамічного адаптера ентропії автоматично балансує коефіцієнт λ_t, повністю ліквідуючи недолік over–refusal.
spotlighting	метод інженерії промптів, що застосовує безперервні алгоритмічні трансформації (наприклад, рандомізацію регістру) до контексту.	глибоко обізнані з алгоритмом зловмисники здатні легко створити адаптивні промпти для ухилення від захисту.	наш захист спирається на незворотний математичний розрив оптимізаційного шляху (виродження градієнта), який неможливо обійти інженерією слів.
secrag	інтегрує топологічні графи знань (kg) у конвеєр пошуку, різко знижуючи успішність змагального перекриття [2, 5].	підтримка багатозв'язкових графів у реальному часі створює високі витрати на інженерію даних та обчислення.	замість побудови важких графів, метод використовує скалярний добуток спроєктованих матриць, що мінімізує навантаження на конвеєр даних.
reflect–guard	змушує модель генерувати структуровані, логічні саморефлексії для протидії складним рольовим атакам [12].	генерація проміжних логічних токенів жорстко серіалізує процес модерації, викликаючи високу затримку (latency).	коригування ймовірностей відбувається на рівні існуючого прихованого стану $\hbar_t$ без генерації додаткових токенів саморефлексії, зберігаючи швидкість.
not what you've signed up for	встановлює фундаментальне емпіричне доведення проблеми непрямої ін'єкції запитів та крадіжки даних [9].	слугує виключно як комплексна модель загроз та ілюстрація вразливостей, а не як самостійне алгоритмічне рішення.	наш метод є прямим інженерним та алгоритмічним вирішенням проблеми конкатенації неперевірених масивів, описаної у цій моделі загроз.
граф–залежні розділювачі (ipiguard)	детерміновано блокує несанкціоноване підвищення привілеїв в автономних агентних графах (dag)[2].	примусове виконання жорстких шляхів виконання обмежує автономні та емерджентні можливості llm.	замість фіксованих графів виконання застосовано ймовірнісне згладжування, що дозволяє зберегти емерджентну автономність агентів при повній безпеці.
запропонований метод	математично гарантує розрив градієнтного шляху оптимізатора gcg; ліквідує over–refusal; зберігає обчислювальну ліквідність vram.	вимагає точного емпіричного калібрування змінної запасу стійкості γ та порогу мертвої зони τ перед фінальним розгортанням на нових архітектурах.	синтезує найкращі риси геометричного поділу та логіт–штрафування, позбуваючись їхніх недоліків. забезпечує захист від атак нульового дня (zero–day) без перенавчання.

Проведений розширений аналіз доводить, що сучасний ландшафт безпеки LLM страждає від критичної фрагментації. Більшість методів (такі як semalith v1.4, llama guard 3) покладаються на модулі постобробки або паралельної класифікації, що неминуче призводить до деградації швидкості реакції системи (latency).

Інші підходи (наприклад, jailbreak rapid response) є принципово реактивними, дозволяючи зловмисникам здійснити каскадний витік даних (data exfiltration) під час першої успішної ітерації атаки.

Запропонована нами концептуальна архітектура покращена за рахунок того, що вона діє на фундаментальному рівні тензорної механіки. Інтегрувавши модуль геометричної проекції, ми математично відокремили простір системних директив від неперевіреного простору rag–документів, де найчастіше ховається ін'єкція запитів. Додавання динамічного адаптера ентропії вирішило ключову проблему методу safedecoding – ризик надмірного блокування легітимних запитів (over–refusal). Завдяки зворотному зв'язку між інформаційною ентропією та коефіцієнтом штрафу, система залишається гнучкою, блокуючи генерацію лише за наявності математично доведеного семантичного конфлікту.

Висновки та перспективи подальших досліджень

Проведений порівняльний аналіз та теоретичне обґрунтування розробленого методу підвищення стійкості великих мовних моделей дозволяють зробити кілька ключових висновків.

По–перше, ландшафт загроз для систем штучного інтелекту зазнав безпрецедентної еволюції. З появою великомасштабних генеративних архітектур фокус зловмисників змістився на атаки, що експлуатують логіку та взаємодію моделі із зовнішнім середовищем. Така вразливість, як ін'єкція запитів, розмиває традиційну межу між даними та системними інструкціями, перетворюючи будь–яке неперевірене джерело в rag–системах на потенційний вектор компрометації.

По–друге, найбільш критичними для сучасних інфраструктур є приховані та персистентні загрози, що підривають цілісність на структурному рівні. В умовах найгірших сценаріїв (worst–case scenarios) успішна алгоритмічна атака провокує каскадний витік даних (data exfiltration) та несанкціоноване підвищення привілеїв (privilege escalation) в автономних агентних графах. Ці системні ризики компрометації вимагають застосування суворих математичних бар'єрів, оскільки класичні евристичні фільтри є вразливими до методів безперервної оптимізації градієнтів (gcg).

По–третє, у сфері захисних механізмів існує гострий розрив між методами з доказовими гарантіями (які руйнують продуктивність) та слабкими практичними рішеннями. Запропонований нами апарат успішно долає цей бар'єр, доводячи, що ефективна безпека ші є емерджентною характеристикою багаторівневого підходу «захисту в глибину» (defense–in–depth). Інтеграція ортогональної геометричної проекції та динамічного адаптера ентропії забезпечує виродження градієнта $\nabla_S L_{adv} \approx 0$, математично гарантуючи розрив шляху атаки. При цьому застосування 4–бітного квантування матриць ліквідує ризик вичерпання системної пам'яті, підтримуючи високу пропускну здатність моделі.

На основі виявлених архітектурних обмежень та невирішених питань у галузі кібербезпеки LLM окреслено наступні перспективні напрями для майбутніх наукових досліджень:

- формальні гарантії для мультимодальних компонентів: оскільки візуальні та аудіо–моделі мають інакшу топологічну природу, дослідження мають зосередитися на адаптації змінної запасу стійкості γ для крос–модальних матриць багатонаправленої уваги, щоб запобігти злиттю візуальних атак із текстовими інструкціями.

- стандартизація та бенчмарки для оцінювання безпеки: наразі відсутні єдині стандарти для кількісного вимірювання стійкості rag–застосунків; необхідна розробка комплексних інструментів для об'єктивного розрахунку ймовірності asr (attack success rate) при інтеграції нашого геометричного контролю в розподілені системи федеративного навчання (federated learning).

- комбіновані та адаптивні захисти в rag–ланцюгах: складні ланцюги агентів (agent chains) потребують вивчення синергетичного ефекту від поєднання нашого динамічного логіт–штрафування з іншими алгоритмічними фільтрами на етапі векторного пошуку та валідації виводу.

- методики системного моніторингу та виявлення аномалій: автоматизація калібрування порогу τ та максимального коефіцієнта λ_{max} через алгоритми мета–навчання (meta–learning), що дозволить системі динамічно розпізнавати складні, багатоетапні атаки в реальному часі та миттєво адаптуватися до нових open–weights архітектур.

Література

[1]. Addagada, T. C. (2026). Semalith v1.4: a calibrated 184M safety classifier achieving state–of–the–art prompt–injection detection at 44x fewer parameters than Llama–Guard–3–8B. arXiv preprint arXiv:2607.22545.

[2]. An, H., Zhang, J., Du, T., Zhou, C., Li, Q., Lin, T., & Ji, S. (2024). IPIGuard: a novel tool dependency graph–based defense against indirect prompt injection in LLM agents. arXiv preprint arXiv:2508.15310.

[3]. Arditi, A., et al. (2024). Refusal in language models is mediated by a single direction. arXiv preprint arXiv:2406.11717.

[4]. Chao, P., Debenedetti, E., Robey, A., Andriushchenko, M., Croce, F., Sehwag, V., Dobriban, E., Flammarion, N., & J., George. (2024). JailbreakBench: an open robustness benchmark for jailbreaking large language models. NeurIPS 2024.

[5]. Chen, S., Piet, J., Sitawarin, C., & Wagner, D. (2024). StruQ: defending against prompt injection with structured queries. arXiv preprint arXiv:2402.06363.

[6]. Cloudflare. (2023). OWASP top 10 risks for LLMs. Cloudflare Learning Center. У контексті імплементації MITRE ATLAS.

[7]. Debenedetti, E., Zhang, J., Balunovic, M., Beurer–Kellner, L., Fischer, M., & Tramèr, F. (2024). Agentdojo: a dynamic environment to evaluate prompt injection attacks and defenses for llm agents. NeurIPS 2024, 37, 82895–82920.

[8]. Greshake, K. et al. (2023). AI injections: direct and indirect. Опубліковано в рамках OWASP LLM Risk 2023–24.

[9]. Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: compromising real–world LLM–integrated applications with indirect prompt injection. arXiv preprint arXiv:2302.12173.

[10]. Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Lambert, N., Choi, Y., & Dziri, N. (2024). WildGuard: open one–stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. NeurIPS 2024 Camera Ready / arXiv:2406.18495.

[11]. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.–M., Wang, W.–C., Xiao, G., Dang, X., Gan, C., & Han, S. (2024). AWQ: activation–aware weight quantization for LLM compression and acceleration. MLSys 2024.

[12]. Luo, H. et al. (2026). Reflect–Guard: enhancing LLM safeguards against adversarial prompts via logical self–reflection. arXiv preprint arXiv:2605.24834.

[13]. NIST. (2023). Artificial intelligence risk management framework (AI RMF 1.0). National Institute of Standards and Technology.

[14]. OWASP. (2023). OWASP top 10 for large language model applications.

[15]. Pant, D., Lohani, S., & Kumar, A. (2026). On the inseparability of instructions and data in shared–embedding sequence models. arXiv preprint arXiv:2606.27567.

[16]. Paulus, A., Zharmagambetov, A., Guo, C., Amos, B., & Tian, Y. (2024). AdvPrompter: fast adaptive adversarial prompting for LLMs. ICML 2024 / arXiv preprint arXiv:2404.16873.

[17]. Xu, Z., Jiang, F., Niu, L., Jia, J., Lin, B. Y., & Poovendran, R. (2024). SafeDecoding: defending against jailbreak attacks via safety–aware decoding. ACL 2024 / arXiv preprint arXiv:2402.08983.

[18]. Zhang, Y., Ding, Z. (2024). SelfDefend: llms can defend themselves against jailbreaking in a practical manner. arXiv preprint arXiv:2606.16527v1.

[19]. Zhu, S., Zhang, R., An, B., Wu, G., Barrow, J., Wang, Z., Huang, F., Nenkova, A., & Sun, T. (2023). AutoDAN: interpretable gradient–based adversarial attacks on large language models. First Conference on Language Modeling (COLM) / arXiv:2310.15140.

[20]. Zou, A., Phan, L., Chen, S., Campbell, J., Popov, P., Ren, A., ... & Hendrycks, D. (2023). Representation Engineering: a top–down approach to AI transparency. arXiv preprint arXiv:2310.01405.

[21]. Zou, A., Wang, Z., Kolter, J. Z., & Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models. ICML 2023.

[22]. Zou, W., Geng, R., Wang, B., & Jia, J. (2024). PoisonedRAG: knowledge poisoning attacks to retrieval–augmented generation of large language models. arXiv preprint arXiv:2402.07867

УДК 003.26:004.056.55:621.39(045)

Bondarovets S., Okhrimenko T. Method for Enhancing the Robustness of Large Language Models against Adversarial Attacks

Abstract. The paper presents a mathematically grounded method for enhancing the robustness of large language models (LLMs) against input-level manipulations, systematizing adversarial attack vectors and mechanisms for their preventive neutralization. The introductory section outlines the fundamental architectural vulnerability of autoregressive transformers – the conflation of operational instructions and unverified data within a single memory space, which generates systemic compromise risks and forms the core of threats such as prompt injection. The main part of the study proposes a comprehensive apparatus that operates at the fundamental level of tensor mechanics. To prevent gradient merging, a geometric projection module is developed, which orthogonally separates the query and RAG–context embeddings. To eliminate the risk of systemic memory liquidity depletion, 4–bit asymmetric quantization of transformation matrices is applied. In parallel, a dynamic entropy adapter is introduced, which algorithmically adjusts the penalty coefficient variable at the pre–softmax generation stage, eliminating the risk of over–refusal of legitimate

queries. Additionally, an extended comparative analysis is conducted, which summarizes the advantages of the proposed architecture over existing reactive methods and resource-intensive shadow models. It is mathematically proven that the proposed logit shift renders the optimization gradient degenerate, guaranteeing an exponential decay in the probability of a successful attack. The conclusions summarize the key results and outline promising directions for further research: development of formal security guarantees for multimodal environments, standardization of benchmarks, and evolution of comprehensive monitoring systems.
Keywords: artificial intelligence; cybersecurity; artificial intelligence protection; large language models; adversarial attacks.

БОНДАРОВЕЦЬ Сергій Сергійович, аспірант, Національний університет «Київський авіаційний інститут».
BONDAROVETS Serhii, PhD Student, National University «Kyiv Aviation Institute».

ОХРІМЕНКО Тетяна Олександрівна, к.т.н., старший дослідник, заступник декана з наукової роботи Факультету комп'ютерних наук та технологій, Національний університет «Київський авіаційний інститут».
OKHRIMENKO Tetiana, Ph.D., Senior Researcher, Deputy Dean for Scientific Work, Faculty of Computer Science and Technology, National University «Kyiv Aviation Institute».

Дата надходження:	Дата прийняття:	Дата публікації:
30.06.2026	24.07.2026	21.08.2026