

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

## **ЛЕКЦІЯ 9. Великі мовні моделі**

---

**Львів -- 2025**

# Лекція зі штучного інтелекту 2025-09

---

## Вступ

---

На цьому занятті ми розглянемо великі мовні моделі (Large Language Models, LLM) — передові системи штучного інтелекту, які були розроблені для розуміння та генерації людської мови. Ці моделі, такі як GPT-4, Claude, Gemini та інші, здатні виконувати різноманітні завдання, від написання текстів і програмного коду до вирішення математичних задач та надання відповідей на запитання. Ми дослідимо базові принципи роботи цих моделей, їхню архітектуру, процес навчання та застосування, а також розглянемо їхні обмеження та етичні аспекти використання.

## Теми, що розглядаються

---

1. Концепція великих мовних моделей та їх еволюція
2. Процес попереднього навчання (pre-training)
3. Токенізація тексту
4. Архітектура трансформерів
5. Процес виведення (inference) у мовних моделях
6. Навчання з інструкціями (supervised fine-tuning)
7. Навчання з підкріпленням від людського зворотного зв'язку (RLHF)
8. Психологія ВММ: галюцинації та когнітивні особливості
9. Мультимодальні моделі та інструменти
10. Майбутнє ВММ та їхні обмеження

## Попереднє навчання (Pre-training)

---

### Збір та обробка даних

Перший етап створення великої мовної моделі — це збір та обробка текстових даних з інтернету. Для прикладу розглянемо процес створення набору даних Fine Web, який використовується багатьма розробниками ВММ:

1. **Збір даних з інтернету:** Основним джерелом даних є Common Crawl — організація, яка індексує веб-сторінки з 2007 року. Станом на 2024 рік, Common Crawl містить близько 2,7 мільярда веб-сторінок.
2. **Фільтрація URL:** Застосовуються спеціальні списки блокування для видалення небажаних джерел, таких як сайти зі спамом, шкідливим кодом, маркетингові сайти, сайти з дорослим контентом тощо.
3. **Вилучення тексту:** Із HTML-сторінок видаляється розмітка, скрипти, CSS та інші нетекстові елементи, залишаючи лише корисний текстовий контент.

4. **Фільтрація за мовою:** Застосовуються класифікатори мови для визначення основної мови кожної сторінки. Наприклад, Fine Web зберігає сторінки, які містять понад 65% англійської мови.
5. **Видалення дублікатів:** Повторювані тексти видаляються для зменшення розміру даних та уникнення перенавчання на однакових прикладах.
6. **Видалення персональної інформації:** Тексти, що містять персональну інформацію (адреси, номери соціального страхування тощо), видаляються або анонімізуються.

У результаті цього процесу створюється високоякісний набір текстових даних. Наприклад, набір даних Fine Web займає близько 44 терабайт дискового простору і містить приблизно 15 трильйонів токенів.

## Токенізація тексту

Перед обробкою нейронною мережею текст потрібно перетворити на послідовність токенів — окремих одиниць, з якими працює модель.

### Процес токенизації:

1. **UTF-8 кодування:** Спочатку текст перетворюється у бінарне представлення за допомогою UTF-8 кодування.
2. **Групування бітів у байти:** Кожні 8 бітів групуються в один байт, що дає 256 можливих символів.
3. **Byte-pair encoding (BPE):** Застосовується алгоритм BPE для подальшого зменшення довжини послідовностей шляхом об'єднання найбільш частих пар символів у нові токени.

Сучасні ВММ використовують словники розміром близько 100 000 токенів. Наприклад, GPT-4 має словник з 100 277 токенів.

### Приклад токенизації:

Фраза "Привіт світ" може бути розбита на декілька токенів, залежно від моделі та алгоритму токенизації. Важливо розуміти, що токени не обов'язково відповідають окремим словам чи символам — вони можуть представляти частини слів, цілі слова або навіть кілька слів разом.

Кожен токен має унікальний числовий ідентифікатор у словнику моделі. Наприклад, токен "Привіт" може мати ID 15339, а "світ" — ID 1917.

## Архітектура нейронної мережі

Великі мовні моделі базуються на архітектурі трансформерів (Transformer) — типі нейронної мережі, спеціально розробленому для обробки послідовностей.

### Основні компоненти трансформера:

1. **Вбудовування токенів (Token Embedding):** Кожен токен перетворюється у вектор фіксованої довжини, який представляє його в багатовимірному просторі.
2. **Блоки уваги (Attention Blocks):** Механізм самоуваги дозволяє моделі встановлювати зв'язки між різними токенами в послідовності, незалежно від їхньої відстані один від одного.

3. **Багатошаровий перцептрон (MLP):** Шари нейронів, які додатково обробляють інформацію після механізму уваги.
4. **Нормалізація шарів (Layer Normalization):** Стабілізує навчальний процес, нормалізуючи активації нейронів.

Сучасні ВММ можуть мати мільярди параметрів. Наприклад, GPT-2 (2019) мав 1,5 мільярда параметрів, а сучасні моделі можуть мати сотні мільярдів або навіть трильйони параметрів.

## Процес навчання нейронної мережі

Навчання ВММ передбачає оптимізацію параметрів нейронної мережі для вирішення задачі передбачення наступного токена в послідовності:

1. **Вхідні дані:** Модель отримує послідовність tokenів (контекст) від початку до певного моменту.
2. **Передбачення:** На основі цього контексту модель передбачає ймовірності для кожного можливого наступного токена зі свого словника.
3. **Обчислення втрати:** Обчислюється розбіжність між передбаченнями моделі та фактичним наступним токеном у тренувальній послідовності.
4. **Оновлення параметрів:** Параметри моделі оновлюються для зменшення цієї розбіжності за допомогою алгоритму градієнтного спуску.

### Приклад процесу передбачення:

Припустимо, модель обробила контекст "Київ є столицею". На виході вона видасть розподіл ймовірностей для всіх можливих наступних tokenів. У цьому випадку token "України" матиме найвищу ймовірність.

## Обчислювальні ресурси для навчання

Навчання сучасних ВММ вимагає значних обчислювальних ресурсів:

1. **Графічні процесори (GPU):** Основний тип обладнання для тренування ВММ. Сучасні моделі тренуються на тисячах GPU, таких як NVIDIA H100.
2. **Розподілене навчання:** Процес навчання розподіляється між багатьма GPU та серверами для прискорення обчислень.
3. **Вартість навчання:** Вартість навчання великих моделей може становити мільйони доларів. Наприклад, у 2019 році вартість навчання GPT-2 оцінювалася в \$40,000, а сучасні моделі можуть коштувати десятки мільйонів.

За останні роки ефективність навчання значно покращилася завдяки вдосконаленню апаратного та програмного забезпечення. Наприклад, сьогодні GPT-2 можна було б навчити за \$100-600 і лише за один день, порівняно з початковими оцінками.

## Базові моделі та інференс

---

## Базові моделі (Base Models)

Результатом етапу попереднього навчання є базова модель — ВММ, яка вміє передбачати наступний токен у послідовності, але ще не налаштована на взаємодію з людиною.

Базову модель можна розглядати як "симулятор інтернет-тексту", який здатний продовжувати текст у стилі, подібному до текстів з інтернету, але не відповідає спеціально на запитання.

### Приклади базових моделей:

- GPT-2 (OpenAI, 2019): 1,5 млрд параметрів
- LLaMA 3 (Meta, 2024): 8-405 млрд параметрів

## Процес виведення (Inference)

Після навчання моделі застосовується процес виведення для генерації нового тексту:

1. **Введення префіксу:** Користувач або система надає початковий текст (промпт).
2. **Авторегресивне передбачення:** Модель передбачає ймовірності для наступного токена.
3. **Вибір токена:** Наступний токен обирається на основі цих ймовірностей (зазвичай з використанням температурного семплування).
4. **Повторення:** Обраний токен додається до контексту, і процес повторюється для генерації наступного токена.

### Стохастичність генерації:

Процес виведення є стохастичним — навіть із однаковим вхідним текстом модель може генерувати різні відповіді при кожному запуску через випадковість вибору токенів.

## Можливості базових моделей

Хоча базові моделі не налаштовані відповідати на запитання, вони містять значні знання, отримані під час попереднього навчання:

1. **Пасивні знання:** Моделі запам'ятовують багато фактів із тренувальних даних, але їх виклик вимагає правильного формулювання запиту.
2. **Ненадійність інформації:** Інформація, що міститься в моделі, є статистичним відображенням тренувальних даних і може бути неточною або неповною.
3. **Дублювання контенту:** Моделі можуть дослівно відтворювати фрагменти частих або важливих документів із тренувальних даних (наприклад, статті з Вікіпедії).
4. **Контекстне навчання:** Базові моделі демонструють здатність до навчання в контексті (in-context learning) — вони можуть розпізнавати патерни в запиті та продовжувати їх, навіть якщо не бачили точно такого патерну під час тренування.

## Навчання з інструкціями (Supervised Fine-Tuning)

---

## Перетворення базової моделі в асистента

Після попереднього навчання отримана базова модель здатна лише імітувати інтернет-текст, але не взаємодіяти з користувачем як корисний асистент. Для перетворення базової моделі в асистента застосовується процес навчання з інструкціями (Supervised Fine-Tuning, SFT).

### Основні етапи процесу SFT:

1. **Створення набору даних розмов:** Збирається набір даних, що містить діалоги між користувачем і асистентом.
2. **Структура розмов:** Кожен діалог складається з послідовності повідомлень користувача та відповідей асистента.
3. **Продовження навчання:** Базова модель продовжує навчатися, але на наборі даних розмов замість інтернет-текстів.

## Створення набору даних розмов

Набір даних для SFT створюється кількома способами:

1. **Ручне створення людьми-анотаторами:** Професійні анотатори пишуть ідеальні відповіді асистента на різноманітні запити.
2. **Керівництво з анотування:** Компанії розробляють детальні інструкції для анотаторів, які визначають, як повинен поводитися асистент (бути корисним, правдивим, нешкідливим).
3. **Синтетичне генерування:** У сучасних системах часто використовуються вже існуючі ВММ для допомоги у створенні або редагуванні даних розмов.

### Приклад інструкцій для анотаторів:

- Відповідати корисно, точно та правдиво
- Визнавати незнання, якщо інформація невідома
- Уникати шкідливого, незаконного або неетичного контенту
- Дотримуватися ввічливого та професійного тону

## Токенізація розмов

Для навчання моделі розмови перетворюються у послідовності токенів з використанням спеціальних маркерів, які вказують на ролі учасників:

1. **Спеціальні токени:** Вводяться нові токени для позначення ролей (наприклад, "користувач:", "асистент:").
2. **Формат розмови:** Кожне повідомлення форматується за певним протоколом, що дозволяє моделі розрізняти ролі та межі повідомлень.

### Приклад токенованої розмови:

```
<IM_START>користувач<IM_SEP>Що таке штучний інтелект?<IM_END>  
<IM_START>асистент<IM_SEP>Штучний інтелект (ШІ) – це галузь комп'ютерних наук...<IM_END>
```

Де `<IM_START>`, `<IM_SEP>`, `<IM_END>` є спеціальними токенами, що позначають початок повідомлення, розділення ролі та контенту, та кінець повідомлення відповідно.

## Психологія моделей після SFT

Після навчання з інструкціями модель набуває певних рис, які можна розглядати як її "психологію":

- Імітація анотаторів:** Модель статистично імітує поведінку анотаторів, які створювали відповіді. Коли користувач взаємодіє з моделлю, він фактично взаємодіє з "симуляцією середньостатистичного анотатора".
- Програмування через приклади:** Модель вчиться формату відповідей, тону, стилю та підходу до розв'язання задач на основі прикладів у наборі даних SFT.
- Відсутність справжнього розуміння:** Незважаючи на впевнений тон відповідей, модель не має справжнього розуміння інформації, а лише відтворює статистичні шаблони з тренувальних даних.

## Порівняння з попереднім навчанням

Навчання з інструкціями відрізняється від попереднього навчання кількома ключовими аспектами:

- Дані:** Замість випадкових інтернет-текстів використовуються спеціально створені діалоги.
- Тривалість і вартість:** SFT значно швидше та дешевше (години замість місяців).
- Мета:** Метою є не загальне моделювання мови, а навчання специфічній поведінці у діалозі.
- Розмір даних:** Набори даних SFT значно менші (мільйони прикладів проти трильйонів токенів у попередньому навчанні).

## Обмеження моделей після SFT

Моделі, які пройшли лише етап SFT, мають певні обмеження:

- Галюцинації:** Тенденція генерувати фактично невірну інформацію з упевненим тоном.
- Нездатність визнавати незнання:** Моделі часто надають відповіді навіть коли не мають достатньо інформації.
- Обмежені навички міркування:** Складні задачі, що вимагають покрокового мислення, можуть бути проблематичними.

## Навчання з підкріпленням від людського зворотного зв'язку (RLHF)

### Концепція навчання з підкріпленням

Навчання з підкріпленням від людського зворотного зв'язку (Reinforcement Learning from Human Feedback, RLHF) — це третій етап навчання BMM, який дозволяє моделі вдосконалити свої навички та краще відповідати людським очікуванням.

Цей етап можна порівняти з розв'язанням практичних задач у навчанні:

- Попереднє навчання — читання загальної теорії
- Навчання з інструкціями — вивчення розв'язаних прикладів
- RLHF — самостійне розв'язання практичних задач з підказками

## Процес RLHF у верифікованих доменах

У верифікованих доменах (математика, кодування, логіка), де можна об'єктивно оцінити правильність відповіді, RLHF має наступні етапи:

1. **Генерація варіантів розв'язання:** Для певного запиту (наприклад, математичної задачі) модель генерує багато різних розв'язків.
2. **Оцінка розв'язків:** Розв'язки порівнюються з правильною відповіддю для визначення, які з них коректні.
3. **Навчання на успішних розв'язках:** Модель додатково навчається на успішних розв'язках, посилюючи стратегії, які привели до правильних відповідей.

### Приклад застосування RLHF для математичної задачі:

Задача: "Олена купила 3 яблука і 2 апельсини. Кожен апельсин коштує 2 грн. Загальна вартість 13 грн. Скільки коштує одне яблуко?"

1. Модель генерує різні підходи до розв'язання.
2. Оцінюються підходи, які дають правильну відповідь (3 грн).
3. Модель навчається на успішних розв'язках, розвиваючи кращі стратегії міркування.

## Моделі мислення (Reasoning Models)

Як результат RLHF у верифікованих доменах з'являються "моделі мислення" — моделі, які демонструють:

1. **Ланцюжки міркувань:** Розбиття складної проблеми на простіші кроки.
2. **Перевірку власних розв'язків:** Переосмислення задачі різними способами для підтвердження результату.
3. **Самокорекцію:** Виявлення та виправлення помилок у процесі міркування.
4. **Метакогніцію:** Міркування про власний процес мислення.

Такі моделі виробляють довші, але значно точніші відповіді, особливо для складних задач.

## RLHF у неверифікованих доменах



У неверифікованих доменах (творче письмо, узагальнення, рекомендації) неможливо об'єктивно оцінити правильність відповіді. Тут застосовується інший підхід:

1. **Створення моделі винагороди:** Навчається окрема нейронна мережа, яка оцінює якість відповідей на основі людських преференцій.
2. **Збір людських порівнянь:** Людям-анотаторам показуються пари відповідей на один запит, і вони вибирають кращу.
3. **Навчання моделі винагороди:** Модель винагороди навчається прогнозувати людські преференції.
4. **Оптимізація відповідей:** Основна модель навчається максимізувати оцінку від моделі винагороди.

## Обмеження RLHF

Незважаючи на значні переваги, RLHF має певні обмеження:

1. **Ігрові стратегії (Gaming):** У неверифікованих доменах моделі можуть знаходити способи отримувати високі оцінки від моделі винагороди, генеруючи безглузді відповіді, які експлуатують слабкості моделі винагороди.
2. **Обмежена оптимізація:** Через проблему ігрових стратегій, RLHF у неверифікованих доменах можна застосовувати лише обмежений час, на відміну від верифікованих доменів, де можлива тривала оптимізація.
3. **Залежність від якості даних:** Якість результатів RLHF безпосередньо залежить від якості людських оцінок та порівнянь.

## Аналогія з AlphaGo

Процес RLHF у верифікованих доменах можна порівняти з навчанням системи AlphaGo для гри в го:

1. **Навчання з інструкціями (Supervised Learning):** Імітація ходів людей-експертів дозволяє досягти високого, але обмеженого рівня.
2. **Навчання з підкріпленням (Reinforcement Learning):** Гра проти себе та відкриття нових стратегій дозволяє перевершити людський рівень.

Подібно до "Ходу 37" AlphaGo (неочікуваний хід, який виявився геніальним), моделі навчені з RLHF можуть розвивати унікальні стратегії міркування, які відрізняються від типового людського підходу.

## Психологія та когнітивні особливості ВММ

---

### Галюцинації та управління знаннями

**Галюцинації** — це генерація фактично неправильної інформації моделлю з упевненим тоном. Основні причини галюцинацій:

1. **Стиль навчальних даних:** У наборах даних SFT відповіді на фактичні запитання зазвичай надаються з упевненим тоном, навіть якщо інформація рідкісна чи складна.
2. **Статистична природа:** Модель видає найбільш імовірні токени на основі тренувальних даних, а не на основі фактичної правди.

#### Способи зменшення галюцинацій:

1. **Визнання незнання:** Додавання прикладів, де модель чесно визнає відсутність інформації.
2. **Емпіричне тестування знань:** Систематичне тестування моделі для визначення меж її знань.
3. **Використання інструментів:** Надання моделі доступу до веб-пошуку або інших джерел інформації.

## Модель пам'яті ВММ

Великі мовні моделі мають два типи "пам'яті":

1. **Параметрична пам'ять:** Знання, закодовані в параметрах моделі під час навчання. Це подібно до довготривалої пам'яті людини, але з нечіткими, статистичними спогадами.
2. **Контекстна пам'ять:** Інформація у поточному контекстному вікні, доступна для прямого використання. Це аналог робочої пам'яті людини.

#### Практичні наслідки:

- Модель краще відповідатиме на запитання, якщо необхідна інформація надана в контексті, а не якщо вона має спиратися лише на свою параметричну пам'ять.
- Для точних відповідей краще надати моделі конкретні дані, а не очікувати відтворення з параметричної пам'яті.

## Потреба в токенах для міркування

Через обмежену кількість обчислень на кожен токен, моделі потребують достатньої кількості токенів для проведення складних міркувань:

1. **Розподіл обчислень:** Складні обчислення мають бути розподілені на кілька токенів для отримання правильного результату.
2. **Проміжні результати:** Модель має генерувати проміжні кроки для складних задач, а не намагатися видати відповідь одразу.
3. **Обмеження обчислень на токен:** Кожен токен має обмежену кількість обчислювальних операцій, що пояснює, чому моделі можуть помилятися у задачах, які здаються простими (наприклад, підрахунок кількості літер).

## Використання інструментів (Tool Use)

Для подолання обмежень і розширення можливостей, сучасні ВММ інтегруються з зовнішніми інструментами:

1. **Веб-пошук:** Дозволяє моделі отримувати актуальну інформацію з інтернету, зменшуючи галюцинації та надаючи доступ до свіжих даних.
2. **Виконання коду:** Модель може писати та виконувати код для обчислень, аналізу даних або інших задач, що потребують точності.
3. **Взаємодія з API:** Дозволяє моделі отримувати інформацію з різних сервісів або керувати зовнішніми системами.

#### Механізм використання інструментів:

1. **Спеціальні токени:** Вводяться спеціальні токени, які позначають початок і кінець використання інструмента.
2. **Протокол взаємодії:** Модель навчається структурувати запити до інструментів у певному форматі.
3. **Інтеграція результатів:** Результати роботи інструментів додаються до контексту моделі для подальшого використання.

#### Приклад використання веб-пошуку:

Користувач: Хто виграв Оскар за найкращий фільм у 2024 році?  
Асистент: <search\_start>Оскар 2024 найкращий фільм переможець</search\_start>  
[Результати пошуку додаються до контексту]  
Асистент: За даними офіційного сайту Кіноакадемії, Оскар за найкращий фільм 2024 року отримав ф

Використання інструментів дозволяє моделям:

- Надавати точніші відповіді
- Виконувати більш складні завдання
- Долати обмеження власних знань
- Працювати з даними, що з'явилися після дати навчання

## Мультимодальні моделі

### Концепція мультимодальності

Мультимодальні моделі здатні працювати з різними типами даних, а не лише з текстом:

1. **Обробка зображень:** Здатність "бачити" та аналізувати візуальний контент.
2. **Обробка аудіо:** Здатність "чути" та розуміти мовлення або інші звуки.
3. **Генерація зображень/аудіо:** Створення візуального або звукового контенту.

#### Принцип роботи мультимодальних моделей:

Додаткові модальності інтегруються у ВММ шляхом:

- Токенізації даних інших модальностей (наприклад, патчі зображень або сегменти спектрограм)
- Створення спільного простору представлень для всіх модальностей

- Навчання моделі на змішаних даних різних типів

## Мультиmodalьні можливості

Поєднання різних модальностей відкриває нові можливості:

1. **Відповіді на запитання щодо зображень:** Модель може аналізувати візуальний контент і відповідати на запитання про нього.
2. **Створення зображень за описом:** Генерація візуального контенту на основі текстового опису.
3. **Транскрибування аудіо:** Перетворення мовлення на текст.
4. **Інтерактивні розмови:** Використання голосового введення та виведення для більш природної взаємодії.

## Приклади мультиmodalьних моделей

Сучасний ландшафт мультиmodalьних BMM включає:

1. **GPT-4V:** Розширення GPT-4 з можливістю обробки зображень.
2. **Claude 3 Vision:** Мультиmodalьна модель від Anthropic.
3. **Gemini:** Мультиmodalьна модель від Google.
4. **DALL-E, Midjourney:** Спеціалізовані моделі для генерації зображень на основі текстових описів.

## Майбутнє великих мовних моделей

---

### Найближчі тенденції розвитку

В найближчі роки очікується кілька ключових тенденцій у розвитку BMM:

1. **Повна мультиmodalьність:** Інтеграція тексту, зображень, аудіо та відео в єдині моделі стане стандартом.
2. **Агентні системи:** Розвиток моделей, здатних виконувати довготривалі завдання з мінімальним наглядом людини.
3. **Інтеграція з комп'ютерними системами:** Моделі отримають більше можливостей керувати комп'ютером (мишею, клавіатурою) для виконання завдань.
4. **Збільшення контекстного вікна:** Розширення можливостей моделей працювати з дуже довгими контекстами (мільйони токенів).

### Дослідницькі напрямки

Актуальні напрямки досліджень включають:

1. **Навчання під час виведення (Test-Time Training):** Дозволить моделям адаптуватися та навчатися під час використання, а не тільки на етапі попереднього навчання.
2. **Покращення ефективності:** Зменшення обчислювальних вимог без втрати якості.

3. **Підвищення надійності та зменшення галюцинацій:** Розробка методів для більш надійних та правдивих відповідей.
4. **Багатоагентні системи:** Взаємодія кількох моделей для виконання складних завдань.

## Етичні аспекти та виклики

Розвиток ВММ пов'язаний з рядом етичних та практичних викликів:

1. **Дезінформація та генерація шкідливого контенту:** Ризик використання для створення фейків або матеріалів, що вводять в оману.
2. **Питання приватності:** Моделі можуть відтворювати чутливу інформацію з тренувальних даних.
3. **Авторські права:** Питання щодо використання захищених матеріалів для навчання моделей.
4. **Соціальний вплив:** Автоматизація завдань може вплинути на робочі місця та соціальні структури.
5. **Контроль та регулювання:** Визначення належного рівня нагляду за розробкою та застосуванням ВММ.

## Висновки

Великі мовні моделі представляють значний прорив у розвитку штучного інтелекту, дозволяючи комп'ютерам розуміти та генерувати людську мову на рівні, що раніше вважався неможливим. Процес їхнього створення включає три основні етапи:

1. **Попереднє навчання:** Набуття загальних знань та мовних навичок з великих обсягів текстів.
2. **Навчання з інструкціями:** Адаптація до діалогової взаємодії та корисних відповідей.
3. **Навчання з підкріпленням:** Вдосконалення здатності міркувати та відповідати відповідно до людських переваг.

Ці моделі мають унікальні когнітивні характеристики:

- Здатність зберігати та використовувати знання з тренувальних даних
- Потреба в послідовності токенів для складних міркувань
- Схильність до галюцинацій при перевищенні меж їхніх знань

Включення інструментів та мультимодальних можливостей розширює функціональність ВММ, дозволяючи їм виконувати ширший спектр завдань з більшою точністю.

Майбутній розвиток ВММ обіцяє ще більшу інтеграцію з комп'ютерними системами, розширені мультимодальні можливості та поліпшену надійність. Однак, важливо враховувати етичні аспекти та потенційні виклики при розробці та впровадженні цих потужних технологій.

## Література

1. Radford, A., et al. (2019). Language Models are Unsupervised Multitask Learners. OpenAI.
2. Brown, T. B., et al. (2020). Language Models are Few-Shot Learners. NeurIPS 2020.

3. Ouyang, L., et al. (2022). Training language models to follow instructions with human feedback. NeurIPS 2022.
4. Christiano, P., et al. (2017). Deep Reinforcement Learning from Human Preferences. NeurIPS 2017.
5. Ziegler, D. M., et al. (2019). Fine-tuning language models from human preferences. arXiv preprint.
6. Vaswani, A., et al. (2017). Attention Is All You Need. NeurIPS 2017.
7. Silver, D., et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587).