**‘НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**
**імені ІГОРЯ СІКОРСЬКОГО»**
**Навчально-науковий інститут прикладного системного аналізу**
**Кафедра штучного інтелекту**

До захисту допущено:
В. о. завідувачки кафедри
_____ Ірина ДЖИГИРЕЙ
«___»_____20__ р.

# Дипломна робота

**на здобуття ступеня бакалавра**
**за освітньо-професійною програмою**
**«Системи і методи штучного інтелекту»**
**спеціальності 122 «Комп'ютерні науки» на тему:**
**«Прогнозування ефективності рекламних оголошень з використанням методів мультимодального штучного інтелекту»**

Виконала:
студентка IV курсу, групи КІ-21
До Данг Ха Мі _____

Керівник:
професорка кафедри штучного інтелекту, д.ф.-м.н.,
професор Купенко Ольга Петрівна _____

Консультант з нормоконтролю:
фахівець І категорії кафедри штучного інтелекту, к.т.н,
доцент Комариста Богдана Миколаївна _____

Рецензент:
професорка кафедри математичних методів системного
аналізу, д.т.н., доцент, Недашківська Надія Іванівна _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студентка _____

Київ – 2026 року

**Національний технічний університет України**
**«Київський політехнічний інститут імені Ігоря Сікорського»**
**Навчально-науковий інститут прикладного системного аналізу**
**Кафедра штучного інтелекту**

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ
В. о. завідувачки кафедри
_____ Ірина ДЖИГИРЕЙ
«___» _____ 2026 р.

**ЗАВДАННЯ**
**на дипломну роботу студенту**
**До Данг Ха Мі**

1. Тема роботи «Прогнозування ефективності рекламних оголошень з використанням методів мультимодального штучного інтелекту», керівник роботи Купенко Ольга Петрівна, професор кафедри ШІ, затверджені наказом по НН ІПСА від «27» травня 2026 р. № 1944-с.

2. Термін подання студентом роботи «05» червня 2026 року.

3. Вихідні дані до роботи: рекламні матеріали платформи Meta Ads

4. Зміст роботи: аналіз предметної галузі та огляд існуючих підходів до скорингу рекламних матеріалів; розробка методології формування цільової змінної; проектування та навчання мультимодальної моделі EfficientNet-B3 + XLM-RoBERTa з крос-модальною увагою; порівняльний аналіз архітектурних конфігурацій; розробка системи пояснення рішень; розгортання веб-інтерфейсу та оцінка практичної ефективності системи.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): електронна презентація.

6. Дата видачі завдання «02» лютого 2026 року.

Календарний план

| No з/п | Назва етапів виконання дипломної роботи | Термін виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Огляд літератури за темою | 20.04.2026 | Виконано |
| 2 | Підготовка першого розділу | 27.04.2026 | Виконано |
| 3 | Підготовка другого розділу | 04.05.2026 | Виконано |
| 4 | Розробка програмного продукту | 18.05.2026 | Виконано |
| 5 | Підготовка третього розділу | 25.05.2026 | Виконано |
| 6 | Оформлення дипломної роботи | 01.06.2026 | Виконано |
| 7 | Підготовка презентації доповіді | 08.06.2026 | Виконано |

Студент Ха Мі ДО ДАНГ

Керівник Ольга КУПЕНКО

# РЕФЕРАТ

Дипломна робота: 108 ст., 6 рис., 21 табл., 45 посилань, 2 додатки.

МУЛЬТИМОДАЛЬНЕ НАВЧАННЯ, СКОРИНГ РЕКЛАМНИХ КРЕАТИВІВ, ЕФЕКТИВНІСТЬ РЕКЛАМИ, EFFICIENTNET-B3, XLM-ROBERTA, CROSS-ATTENTION FUSION, GRAD-CAM++, META ADS, CTR, ВАЛОВИЙ ПРИБУТОК.

Об’єктом дослідження є процес автоматичної оцінки ефективності рекламних креативів на основі аналізу їхнього візуального та текстового змісту. Предметом дослідження є методи та архітектури мультимодального глибокого навчання для передзапускового скорингу рекламних матеріалів у системі Meta Ads. Метою роботи є розробка інтелектуальної системи, яка на основі зображення рекламного креативу та тексту оголошення прогнозує ймовірність його успішності до запуску рекламної кампанії та надає інтерпретовані пояснення прийнятого рішення.

Актуальність роботи зумовлена зростанням витрат на цифрову рекламу та значними втратами бюджету внаслідок запуску неефективних креативів. Задача передзапускової оцінки рекламних матеріалів для платформи Meta Ads залишається недостатньо дослідженою, тоді як існуючі роботи переважно орієнтовані на інші рекламні платформи та великомасштабні набори даних.

У роботі сформовано датасет із 6614 мультимовних рекламних матеріалів Meta Ads із підтвердженими метриками ефективності. Запропоновано методологію формування цільової змінної на основі profit-weighted score, що поєднує нормалізовані показники CTR і валового прибутку. Розроблено мультимодальну архітектуру на базі EfficientNet-B3 та XLM-RoBERTa з механізмом двонаправленої крос-модальної уваги.

Фінальна модель досягла показників Test AUC = 0,810, Accuracy = 0,739 та F1 = 0,739. Проведено порівняльний аналіз дев’яти конфігурацій моделей, за результатами якого встановлено, що використання комбінованих CTR+GP-лейблів підвищує якість класифікації, тоді як збільшення обсягу даних без контролю їхньої якості може погіршувати результати. Реалізовано систему пояснення рішень на основі Grad-CAM++, зонального аналізу та текстової уваги, а також вебінтерфейс на базі Gradio.

Практична цінність роботи полягає в можливості підвищення ефективності відбору рекламних креативів до їх запуску та скорочення витрат на A/B-тестування. Запропоновану систему рекомендовано до використання в задачах передзапускового оцінювання рекламних матеріалів у середовищі Meta Ads.

# ABSTRACT

Bachelor thesis: 108 p., 6 figures, 21 tables, 45 references, 2 appendices.

MULTIMODAL LEARNING, ADVERTISING CREATIVE SCORING, AD EFFECTIVENESS, EFFICIENTNET-B3, XLM-ROBERTA, CROSS-ATTENTION FUSION, GRAD-CAM++, META ADS, CTR, GROSS PROFIT.

The object of the study is the process of automatic evaluation of advertising creative effectiveness based on the analysis of their visual and textual content. The subject of the study is methods and architectures of multimodal deep learning for pre-launch scoring of advertising materials in the Meta Ads environment. The aim of the thesis is to develop an intelligent system that predicts the probability of an advertisement's success before the launch of a campaign using the creative image and ad text, while also providing interpretable explanations of the decision-making process.

The relevance of this work is driven by the growing costs of digital advertising and the substantial budget losses associated with launching ineffective creatives. The task of pre-launch evaluation of advertising materials for the Meta Ads platform remains insufficiently explored, whereas existing studies have primarily focused on other advertising platforms and large-scale datasets.

In this study, a dataset consisting of 6,614 multilingual Meta Ads creatives with verified performance metrics was compiled. A methodology for constructing the target variable based on a profit-weighted score combining normalized click-through rate (CTR) and gross profit indicators was proposed. A multimodal architecture based on EfficientNet-B3 and XLM-RoBERTa with a bidirectional cross-modal attention mechanism was designed and trained.

The final model achieved a Test AUC of 0.810, an Accuracy of 0.739, and an F1-score of 0.739. A comparative analysis of nine model configurations demonstrated that the use of combined CTR+GP labels improves classification performance, whereas increasing the dataset size without ensuring data quality may reduce model effectiveness. In addition, an explainability framework based on Grad-CAM++, region-based analysis, and textual attention was implemented, together with a Gradio-based web interface.

The practical value of the study lies in improving the selection of advertising creatives before their deployment and reducing the costs associated with A/B testing. The proposed system is recommended for use in pre-launch evaluation of advertising materials within the Meta Ads platform.

**ЗМІСТ**

# ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

API – Application Programming Interface.

AUC – Area Under the Curve.

AUC-ROC – Area Under the Receiver Operating Characteristic Curve.

BERT – Bidirectional Encoder Representations from Transformers.

BLIP – Bootstrapping Language-Image Pre-training.

CAM – Class Activation Map.

CLIP – Contrastive Language-Image Pre-training.

CLS – Classification token.

CNN – Convolutional Neural Network.

COCO – Common Objects in Context.

CTA – Call to Action.

CTR – Click-Through Rate.

CVR – Conversion Rate.

EfficientNet – Efficient Neural Network.

F1 – F1-score.

FN – False Negative.

FP – False Positive.

GP – Gross Profit.

GPU – Graphics Processing Unit.

Grad-CAM – Gradient-weighted Class Activation Mapping.

JSON – JavaScript Object Notation.

MLP – Multilayer Perceptron.

NLP – Natural Language Processing.

OCR – Optical Character Recognition.

PAD – Padding token.

PNG – Portable Network Graphics.

ROAS – Return on Ad Spend.

ROC – Receiver Operating Characteristic.

RoBERTa – Robustly Optimized BERT Pretraining Approach.

TN – True Negative.

TP – True Positive.

ViT – Vision Transformer.

VLM – Vision-Language Model.

XLM-R – XLM-RoBERTa (скорочена назва).

XLM-RoBERTa – Cross-lingual Language Model RoBERTa.

YOLO – You Only Look Once.

ZIP – ZIP archive.

ШІ – Штучний інтелект.

# ВСТУП

Сучасний[1] ринок цифрової реклами характеризується надзвичайно високою конкуренцією та стрімко зростаючими витратами на просування продуктів. За даними IAB та EMARKETER, глобальні витрати на цифрову рекламу перевищили 601 млрд доларів у 2023 році, демонструючи зростання понад 9,5% порівняно з попереднім роком [32][33]. Значна їх частина припадає на платні рекламні платформи, зокрема Meta (Facebook та Instagram) – один із провідних інструментів перформанс-маркетингу у сфері електронної комерції та прямого відгуку. У цьому контексті ефективність рекламних креативів – зображень та текстових повідомлень, що складають рекламні оголошення – є визначальним чинником успіху маркетингових кампаній.

Традиційним підходом до відбору ефективних рекламних матеріалів є паралельне A/B-тестування: маркетолог запускає декілька варіантів оголошення з реальним бюджетом і лише після накопичення статистики приймає рішення щодо переможного варіанту. Цей підхід має принциповий недолік – витрати на неефективні матеріали є неминучою платою за отримання статистики і передують будь-якому висновку про якість креативу. Емпіричні дослідження підтверджують, що спостережуваний показник повернення на рекламні витрати (ROAS) систематично завищується через гетерогенність аудиторії [35], а каузальний ефект реклами на продажі значно складніше виміряти, ніж це передбачають стандартні метрики платформи [36]. Це породжує нагальну потребу в системах передзапускової оцінки якості рекламних матеріалів.

Задача автоматичної оцінки рекламних креативів до запуску привертає увагу великих технологічних компаній – Alibaba [2, 3], ByteDance [1], JD.com [6, 7], Yahoo [9] – – які розробляють системи скорингу на основі власних

[1]Тут і нижче використано такий інструменти штучного інтелекту як чат-бот з генеративним штучним інтелектом Claude, виключно для коригування та редагування тексту, створеного автором цієї дипломної роботи, на основі автоматизованої перевірки граматики, структури та стилю, що відповідає Політиці використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського (протокол №11 Вченої ради КПІ ім. Ігоря Сікорського від 11 грудня 2023 р.).

промислових датасетів від сотень тисяч до мільярдів зразків. Проте всі відомі публічні дослідження зосереджені на платформах пошукової або контекстної реклами і не стосуються Meta Ads як основної платформи. Крім того, жодна з опублікованих робіт не використовує валовий прибуток як компоненту цільової змінної – обмежуючись CTR або парою CTR+CVR – що призводить до оптимізації за клікабельністю на шкоду реальній бізнес-цінності. Таким чином, задача передзапускового скорингу рекламних матеріалів для платформи Meta Ads з врахуванням економічної ефективності залишається відкритою.

Метою цієї роботи є розробка інтелектуальної системи скорингу рекламних креативів, яка на основі зображення та тексту оголошення прогнозує ймовірність його успішності до моменту запуску рекламної кампанії та надає маркетологу інтерпретовані пояснення прийнятого рішення.

Об’єктом дослідження є процес автоматичної оцінки ефективності рекламних креативів на основі аналізу їхнього візуального та текстового змісту. Предметом дослідження є методи та архітектури мультимодального глибокого навчання для передзапускового скорингу рекламних матеріалів у системі Meta Ads.

Наукова новизна роботи полягає у: по-перше, першому публічному академічному дослідженні задачі скорингу рекламних креативів на реальних даних Meta Ads з підтвердженими бізнес-метриками; по-друге, оригінальній методології формування цільової змінної на основі profit-weighted score – score = reliable_ctr_norm × 0,6 + gp_norm × 0,4, – яка захищає від clickbait-оптимізації і відсутня в жодній відомій публікації; по-третє, емпіричному підтвердженні того, що CLIP zero-shot (AUC = 0,512 ≈ random) принципово не здатний вирішити задачу скорингу без навчання на реальних рекламних метриках; по-четверте, емпіричному доведенні що якість навчальних даних є важливішою за їхній обсяг – збільшення датасету на 24,3% за рахунок AI-OCR тексту знизило AUC на 0,039.

Для досягнення поставленої мети в роботі вирішуються такі завдання:

– аналіз предметної галузі та систематичний огляд існуючих наукових підходів до задачі оцінки рекламних матеріалів;
– збір та підготовка реального мультимовного датасету з 6 614 рекламних матеріалів платформи Meta Ads з підтвердженими метриками CTR та валового прибутку;
– розробка та обґрунтування методології формування двокомпонентної цільової змінної на основі зваженого CTR та валового прибутку;
– проектування та навчання мультимодальної архітектури на базі EfficientNet-B3 та XLM-RoBERTa з механізмом двонаправленої крос-модальної уваги;
– проведення порівняльного дослідження дев’яти архітектурних і датасетних конфігурацій, включаючи аналіз впливу якості лейблювання та обсягу навчальних даних;
– розробка системи пояснення рішень на основі Grad-CAM++, зонального аналізу 3×3 та attention weights XLM-RoBERTa;
– розгортання системи у вигляді веб-інтерфейсу на основі Gradio з AI-рекомендаціями та аналіз практичної ефективності системи.

Практичне значення отриманих результатів визначається тим, що розроблена система дозволяє маркетологу отримати обґрунтовану оцінку потенційної ефективності рекламного матеріалу до витрати рекламного бюджету. Фінальна модель досягає Test AUC = 0,810. На тестовій вибірці при однаковому бюджеті система знаходить на 50,6% більше успішних матеріалів і скорочує марні запуски на 51,5% порівняно з випадковим відбором. Крім числового score, система надає маркетологу теплову карту Grad-CAM++, зональний аналіз та виділені ключові слова тексту – що перетворює статистичне передбачення на конкретні рекомендації щодо покращення креативу.

Структура роботи. Дипломна робота складається зі вступу, трьох розділів, висновків, переліку посилань та двох додатків. Загальний обсяг роботи – 108 сторінок, 6 рисунків, 21 таблиць, 45 посилань. У першому розділі проведено

аналіз предметної галузі та огляд існуючих підходів, сформульовано постановку задачі. У другому розділі описано методи розробки системи: збір та підготовку датасету і текстових даних, формування цільової змінної, архітектуру моделі, процес навчання, систему пояснення рішень та веб-інтерфейс. У третьому розділі представлено результати порівняльного аналізу архітектур, аналіз впливу якості лейблювання та обсягу даних, аналіз системи пояснення рішень, оцінку практичної ефективності та обмеження методу.

# РОЗДІЛ 1 ПІДХОДІВ ДО ПЕРЕДЗАПУСКОВОГО СКОРИНГУ РЕКЛАМНИХ КРЕАТИВІВ

## 1.1 Актуальність задачі оцінки рекламних креативів

Цифрова реклама є одним із найбільш динамічних секторів сучасної економіки. За даними IAB та PwC, обсяг ринку інтернет-реклами лише у Сполучених Штатах досяг 225 млрд доларів у 2023 році, з яких близько 64,9 млрд припадає на соціальні медіа [32]. У глобальному масштабі сукупні витрати на цифрову рекламу перевищили 601 млрд доларів, демонструючи зростання понад 9,5% у порівнянні з попереднім роком [33]. Перформанс-маркетинг – зокрема реклама в Meta Ads (Facebook та Instagram) – є одним із провідних інструментів залучення клієнтів і прямих продажів для компаній у сфері електронної комерції та прямого відгуку.

Рекламний креатив – зображення, відео або банер у поєднанні з текстом оголошення (headline та body copy) – є ключовим елементом, що визначає ефективність рекламної кампанії. Саме якість креативу безпосередньо впливає на клікабельність (CTR), конверсію та кінцевий прибуток. Платформа Meta Ads використовує власні системи ранжування якості та релевантності, які оцінюють кожен креатив після запуску кампанії [34], однак до моменту отримання статистики маркетолог вже витрачає частину рекламного бюджету на тестування.

Традиційним підходом до відбору ефективних рекламних матеріалів у перформанс маркетингу є паралельне тестування: креативи з різними візуальними образами, посилами та текстами запускаються одночасно з реальним рекламним бюджетом, після чого на підставі накопиченої статистики ті, що демонструють кращі показники, масштабуються, а решта зупиняється. Незважаючи на широке застосування, цей підхід має принциповий недолік – витрати на неефективні матеріали є неминучою платою за отримання статистики і передують будь-якому висновку про якість креативу. Емпіричні

дослідження свідчать, що спостережуваний показник повернення на рекламні витрати (ROAS) систематично завищується через гетерогенність аудиторії [35]. Крім того, польові експерименти підтверджують, що каузальний ефект реклами на продажі значно складніше виміряти, ніж це передбачають стандартні метрики платформи [36]. Це породжує нагальну потребу в системах попередньої оцінки якості рекламного контенту – до моменту витрати бюджету.

Дослідження в галузі маркетингових комунікацій встановили, що різні елементи рекламного повідомлення нерівномірно привертають увагу споживача: зображення, брендові елементи та текст конкурують за обмежений обсяг уваги, і їхній відносний розмір та розташування критично впливають на запам'ятовуваність та конверсію [37]. Комп'ютерні моделі зорової уваги дозволяють апроксимувати людське сприйняття сцени без проведення дорогих айтрекінгових досліджень [38], що відкриває можливість для автоматизованої оцінки візуальних характеристик рекламного матеріалу.

Паралельно з розвитком теорії уваги в рекламі, глибоке навчання забезпечило потужні інструменти для автоматичного аналізу зображень і тексту. Моделі прогнозування клікабельності на основі нейронних мереж застосовуються в промислових системах таких компаній, як Alibaba [2], ByteDance [1], JD.com [6] та Yahoo [9]. Проте більшість опублікованих промислових рішень орієнтовані на платформи пошукової або контекстної реклами і оперують датасетами від одного мільйона до кількох мільярдів зразків. Для платформи Meta Ads, яка є провідним інструментом перформанс-маркетингу у західному ринку, публічних академічних досліджень із застосуванням реальних рекламних метрик практично не існує.

Окремої уваги заслуговує задача прогнозування ефективності в умовах так званого cold-start: коли рекламний креатив є новим і не має накопиченої статистики показів. Саме в цьому сценарії системи на основі онлайн-статистики принципово непридатні, і єдиною альтернативою є аналіз контенту – зображення та тексту – ще до першого показу. Цю задачу вперше формалізовано

у роботі Zhao et al. (ByteDance, 2019) під назвою PEAC [1], яка заклала основи контентно-орієнтованого підходу до оцінки рекламних матеріалів до запуску.

Водночас ефективність рекламного креативу визначається не лише клікабельністю. Висока CTR при нульовому валовому прибутку характерна для «клікбейтних» матеріалів, що залучають кліки, але не забезпечують конверсій. Тому комплексна оцінка якості креативу має враховувати як поведінкові метрики (CTR), так і економічні результати (прибуток). Включення gross profit як компоненти цільової змінної дозволяє захистити систему від оптимізації за хибними сигналами і наблизити метрику оцінки до реальної бізнес-цінності рекламного матеріалу.

Таким чином, задача автоматичної оцінки рекламних креативів до запуску є актуальною як з наукової, так і з практичної точки зору. Вона поєднує виклики мультимодального навчання, прогнозування поведінкових метрик та роботи з реальними бізнес-даними в умовах обмеженого обсягу навчальної вибірки. Розробка системи, яка б дозволяла маркетологу отримати обґрунтовану оцінку потенційної ефективності креативу ще до витрати бюджету, має пряме прикладне значення для індустрії перформанс-маркетингу.

## 1.2 Огляд існуючих підходів до скорингу рекламних матеріалів

Задача автоматичної оцінки рекламних матеріалів привертає увагу дослідників переважно з боку великих технологічних компаній, які мають доступ до масштабних рекламних платформ і реальних метрик ефективності. Опубліковані роботи зосереджені навколо платформ Alibaba, ByteDance, JD.com, Yahoo та Gunosy; жодна з них не стосується Meta Ads як основної платформи дослідження.

Фундаментальною роботою у напрямку оцінки креативів до запуску є дослідження Zhao et al. (ByteDance, 2019), відоме під назвою PEAC [1]. Автори

формалізували задачу cold-start оцінки рекламних матеріалів: коли новий креатив ще не має накопиченої статистики показів, система має спрогнозувати його ефективність виключно на основі контенту – зображення та тексту. Запропонована архітектура поєднувала візуальні та текстові ознаки і навчалась передбачати ранг креативу відносно вже відомих матеріалів. Ця робота заклала концептуальну основу для всього напрямку content-based pre-launch scoring і є найближчим академічним аналогом задачі, що вирішується в цій роботі.

Alibaba представила серію досліджень у галузі ранжування рекламних матеріалів. Wang et al. (2021) запропонували модель Visual Attention Model (VAM) у поєднанні з гібридним бандитом для ранжування креативів у display-рекламі [2]. Особливістю цієї роботи є публічний датасет CreativeRanking обсягом 1,7 млн рекламних зображень, який використовується як зовнішній бенчмарк у галузі. Пізніша робота Sheng et al. (Taobao, 2024) представила дворівневу систему попереднього навчання та інтеграції мультимодальних представлень для ранжування реклами [3], яка продемонструвала приріст ROI на 2,9% у промисловому розгортанні. Важливо зазначити, що підхід Sheng et al. передбачає двофазне навчання – попереднє тренування енкодерів з подальшою інтеграцією – на відміну від класичного заморожування ваг.

Компанія JD.com опублікувала два взаємодоповнюючих дослідження. Liu et al. (2020) розробили Category-Specific CNN (CSCNN) – візуальну модель прогнозування CTR, що враховує категорію товару як умовний сигнал при обробці зображення [6]. Yang et al. (2024) розширили підхід до спільного ранжування оголошень та їхніх креативів у реальному часі [7], вирішуючи задачу узгодженості між рівнем оголошення і рівнем рекламного матеріалу в промисловій системі.

Серед досліджень на малих датасетах найближчою до умов цієї роботи є праця Kitada et al. (Gunosy, 2019) [4], яка застосовує мультизадачні умовні мережі уваги для одночасного прогнозування CTR і CVR на датасеті обсягом близько 14 тисяч зразків – масштаб, зіставний із 6 614 зразками, що

використовуються в цій роботі. Це підтверджує практичну можливість побудови ефективних моделей скорингу без доступу до промислових обсягів даних.

Yahoo запропонував підхід до генерації та ранжування рекламних матеріалів через мультимодальну систему, що поєднує оцінку якості з автоматичним вдосконаленням контенту [9]. Дослідження Lin et al. (Alibaba, SIGIR 2022) розглянули спільну оптимізацію ранжування оголошень і відбору креативів у каскадній архітектурі [8], що демонструє тенденцію до інтеграції задач оцінки та відбору в єдиному pipeline.

Окремим напрямком є оцінка якості рекламних текстів. Zhang et al. (CyberAgent, 2024) представили AdTEC – перший відкритий бенчмарк для автоматичної оцінки якості тексту у пошуковій рекламі [5], що охоплює п'ять задач, включаючи прогнозування CTR. Найкращий результат на задачі класифікації ефективності текстів складає Accuracy 0,711 та F1 0,688 при використанні виключно текстових ознак. Мультимодальна модель цієї роботи перевищує ці показники (Accuracy 0,739, F1 0,739), що свідчить про додаткову цінність візуальної модальності.

Спільною рисою переважної більшості розглянутих підходів є орієнтація на платформи пошукової або контекстної реклами (Alibaba, JD.com, ByteDance, Google) та використання великих промислових датасетів від сотень тисяч до мільярдів зразків. Жодна з опублікованих академічних робіт не досліджує платформу Meta Ads як основне джерело даних. Крім того, жодна з них не використовує валовий прибуток як компоненту цільової змінної, обмежуючись CTR або парою CTR+CVR.

Порівняльний аналіз розглянутих підходів наведено в таблиці 1.1.

Аналіз таблиці 1.1 підтверджує, що ця робота займає унікальну нішу в існуючому ландшафті досліджень: поєднання платформи Meta Ads, GP-зваженої цільової змінної, малого датасету та інтегрованого модуля пояснення рішень відрізняє її від усіх відомих опублікованих підходів.

**Таблиця 1.1** – Порівняльний аналіз існуючих підходів до скорингу рекламних матеріалів

| ***Робота*** | ***Платформа*** | ***Підхід*** | ***Дата сет*** | ***Метрика*** | ***Особливість*** |
|---|---|---|---|---|---|
| Zhao et al., PEAC [1] | ByteDance / Toutiao | Multimodal (image + text), cold-start | ~млн | AUC, CTR | Перша робота з pre-launch content scoring |
| Wang et al., CreativeRanking [2] | Alibaba Display | Visual Attention Model (VAM) + Hybrid Bandit | 1,7 млн | CTR rank | Публічний датасет CreativeRanking; image-only |
| Sheng et al. [3] | Alibaba / Taobao | Two-phase pretrain + multimodal integration | млрд | ROI +2.9% | Pretrained encoders; підтверджує ефект заморожених ваг |
| Kitada et al. [4] | Gunosy (Japan) | Multi-task attention (CTR + CVR) | ~14 тис. | CTR, CVR | Малий датасет; найближчий за масштабом до цієї роботи |
| Liu et al., CSCNN [6] | JD.com | Category-specific CNN (visual CTR) | ~млн | AUC, CTR | Категорійна умовна обробка зображень |
| Yang et al. [7] | JD.com | Parallel ranking (ads + creatives) | Промисл. | CTR rank | Спільне ранжування оголошень і креативів в реальному часі |
| Mishra et al. [9] | Yahoo | Multimodal generation + ranking | Промисл. | Quality score | Генерація та ранжування варіантів рекламного матеріалу |
| Zhang et al., AdTEC [5] | CyberAgent (Search) | Text-only benchmark (5 tasks) | Відкритий | Acc 0.711 F1 0.688 | Перший відкритий бенчмарк якості рекламних текстів |
| Ця робота | Meta Ads (Facebook) | EfficientNet-B3 + XLM-R + Cross-Attention | 6 614 | AUC 0.810 | Перша академічна робота на Meta Ads; GP-зважена мітка; Grad-CAM++ explainability |

## 1.3 Методи мультимодального навчання та архітектури злиття модальностей

Рекламний креатив є за своєю природою мультимодальним об’єктом: він поєднує візуальний компонент – зображення або відео – із текстовим посилом у вигляді заголовку (headline) та основного тексту (body copy). Кожна з цих модальностей несе самостійний інформаційний сигнал, і їхня взаємодія визначає загальне сприйняття оголошення споживачем. Відповідно, підходи до автоматичного аналізу рекламних матеріалів можна розділити на три категорії: image-only, text-only та мультимодальні методи злиття.

Підходи на основі виключно зображень (image-only) стали першим напрямком досліджень у галузі автоматичної оцінки реклами. Класичні згорткові мережі – ResNet [18], EfficientNet [16] та інші – дозволяють витягувати ознаки кольорової палітри, композиції, наявності облич і продуктів. Робота Wang et al. (Alibaba) демонструє, що візуальна увага до різних зон зображення корелює з клікабельністю [2].

Водночас image-only моделі принципово не здатні враховувати маркетинговий посил оголошення – pain point у заголовку, заклик до дії (CTA), цінову пропозицію – що є критичним сигналом для перформанс-маркетинг креативів.

Підходи на основі виключно тексту (text-only) отримали новий імпульс із появою трансформерних мовних моделей. BERT [21] та його похідні – RoBERTa [22], XLM-RoBERTa [20] – навчились представляти семантику тексту у вигляді щільних векторів, що дозволяє моделі виявляти ключові маркетингові конструкції: терміновість, соціальний доказ, конкретні обіцянки. Бенчмарк AdTEC [5] показав, що text-only підходи досягають Accuracy 0,711 та F1 0,688 на задачі оцінки ефективності рекламних текстів. Проте у форматі перформанс-маркетинг статичного зображення текст оголошення і візуал несуть різний контент: зображення може зображати продукт або ситуацію вживання,

тоді як текст формулює посил і CTA. Ігнорування однієї з модальностей призводить до систематичної втрати сигналу.

Мультимодальні підходи поєднують обидва джерела інформації. Концептуально найпростішим є пізнє злиття (late fusion або concat): вектори зображення і тексту конкатенуються і подаються на спільний класифікатор. Kiela et al. (MMBT) [10] показали, що навіть просте поєднання попередньо навчених енкодерів дає стрибок якості порівняно з одномодальними базовими лініями на задачах класифікації зображень і тексту. Водночас конкатенація не моделює взаємодію між модальностями: вектори просто склеюються, і класифікатор має самостійно виявити релевантні крос-модальні залежності, що ускладнює навчання при обмеженому обсязі даних.

Більш виразним підходом є крос-модальна увага (cross-attention fusion), де кожна модальність «запитує» інформацію в іншої. Теоретичною основою є механізм масштабованої скалярної уваги, запропонований Vaswani et al. [14], де запит (query), ключ (key) і значення (value) можуть належати різним послідовностям. У крос-модальному контексті це означає, що візуальне представлення може «запитувати» текстові ознаки, і навпаки – забезпечуючи направлений інформаційний обмін між модальностями.

ViLBERT [11] реалізував цю ідею у двопотоковій архітектурі з co-attentional трансформерними шарами між візуальним і текстовим потоками. ALBEF [12] вніс важливе вдосконалення: перед крос-модальним злиттям представлення вирівнюються у спільному просторі через контрастивне навчання, що усуває модальний розрив і забезпечує семантичну узгодженість векторів до їхньої взаємодії. BLIP-2 [13] представив якісно новий підхід: легкий Q-Former слугує мостом між замороженим візуальним енкодером і великою мовною моделлю. Цей підхід демонструє, що при обмеженому обсязі доменних даних заморожені енкодери у поєднанні з невеликим навчальним компонентом здатні досягати конкурентних результатів – що є принципово важливим для умов цієї роботи (~6 600 зразків).

Окремої уваги заслуговує підхід CLIP (Contrastive Language-Image Pretraining) [19], навчений на 400 мільйонах пар зображення-текст у контрастивному режимі. Модель вміє зіставляти зображення і текстові описи у спільному семантичному просторі без додаткового навчання (zero-shot). Однак у контексті перформанс-маркетингу CLIP демонструє принципово недостатню ефективність: у цій роботі zero-shot CLIP досягає AUC 0,512 – результат, статистично еквівалентний випадковому класифікатору.

Це узгоджується з висновками Chen et al. (JD.com, 2025), які показали, що великі візуально-мовні моделі без fine-tuning на реальних рекламних метриках досягають лише AUC ~0,50 – результат, еквівалентний випадковому класифікатору.

Трансформерні моделі з крос-модальною увагою вимагають значних обсягів навчальних даних для повного fine-tuning. В умовах малого датасету це обґрунтовує стратегію поступового розморожування [28]: спочатку навчається лише fusion-голова, потім поступово розморожуються верхні шари енкодерів – що дозволяє адаптувати представлення до рекламного домену без втрати узагальнюючої здатності [29, 45].

У цій роботі реалізовано симетричний варіант крос-модальної уваги: шар v2t формує запит з візуального вектора до текстового, шар t2v – навпаки. Результати об'єднуються конкатенацією і подаються на MLP-класифікатор; residual-з'єднання та LayerNorm стабілізують навчання [14]. Детальний опис архітектури наведено в підрозділі 2.5.

Емпіричне порівняння підтверджує теоретичні очікування: додавання текстової модальності забезпечує приріст AUC на 0,048, а мультимодальна модель з крос-модальною увагою досягає Test AUC 0,810 та F1 0,739 – модель виявляє не лише окремі візуальні чи текстові сигнали, але й їхню взаємодію. Порівняльний аналіз наведено в таблиці 1.2.

**Таблиця 1.2** – Порівняльний аналіз підходів до мультимодального злиття

| *Підхід* | *Переваги* | *Недоліки* | *Типові моделі* |
|---|---|---|---|
| Image-only | Не потребує текстових даних; швидке навчання; ефективний для брендових і продуктових зображень | Ігнорує посил оголошення; не враховує CTA і headline; втрачає семантичний контекст | ResNet [18], EfficientNet [16], ViT [17], VAM [2] |
| Text-only | Добре вловлює маркетинговий посил, pain point, CTA; ефективний для пошукової реклами | Ігнорує візуальні елементи; у перформанс-маркетингу зображення несе значну частину сигналу | BERT [21], RoBERTa [22], XLM-R [20], AdTEC [5] |
| Late fusion (concat) | Простота реалізації; незалежне навчання енкодерів; добре з попередньо навченими моделями | Відсутній явний механізм моделювання взаємодії модальностей; класифікатор навчає крос-модальні залежності самостійно | MMBT [10] |
| Cross-attention fusion | Явне моделювання крос-модальних залежностей; направлений інформаційний обмін; ефективно при малих датасетах | Складніше навчання; більша кількість параметрів; потребує якісних даних обох модальностей | ViLBERT [11], ALBEF [12], BLIP-2 [13], ця робота |
| CLIP zero-shot | Не потребує навчання на доменних даних; універсальний для загальних задач | AUC ~0.51 на перформанс-маркетинг (еквівалентно випадковому); не розуміє бізнес-метрики без fine-tuning | CLIP ViT-B/32 [19] |

Таким чином, мультимодальне злиття з крос-модальною увагою є обґрунтованим вибором для задачі скорингу рекламних креативів: воно дозволяє моделі враховувати як візуальний контент зображення, так і семантику рекламного тексту, виявляючи їхню взаємодію через механізм уваги. Kiela et al. [10] емпірично підтвердили перевагу attention-based fusion над простим concat

на задачах класифікації зображень і тексту, що є теоретичним обґрунтуванням архітектурного вибору цієї роботи. Конкретні рішення щодо вибору енкодерів та параметрів навчання описано в розділі 2.

### 1.4 Постановка задачі

На основі проведеного аналізу предметної галузі, огляду існуючих підходів та методів мультимодального навчання сформульовано задачу, яка вирішується в цій роботі.

**Об'єктом дослідження** є процес автоматичної оцінки ефективності рекламних креативів на основі аналізу їхнього візуального та текстового змісту.

**Предметом дослідження** є методи та архітектури мультимодального глибокого навчання для передзапускового скорингу рекламних матеріалів у системі Meta Ads.

**Метою роботи** є розробка інтелектуальної системи, яка на основі зображення рекламного креативу та тексту оголошення прогнозує ймовірність його успішності до моменту запуску рекламної кампанії, а також надає маркетологу інтерпретовані пояснення прийнятого рішення.

Формально задача визначається таким чином. Нехай дано рекламний креатив, що складається з двох компонентів: зображення та тексту оголошення. Потрібно побудувати функцію f, яка відображає пару (зображення, текст) у скалярну оцінку ймовірності успішності, формула (1.1):

$$f : (I, T) \rightarrow \hat{y} \in [0, 1], \tag{1.1}$$

де $I$ – зображення рекламного креативу, $T$ – текст оголошення (headline та body copy), $\hat{y}$ – передбачена ймовірність належності до класу «успішний креатив».

Цільова змінна (мітка) формується на основі реальних рекламних метрик: зваженого показника клікабельності та валового прибутку. Для кожного креативу обчислюється композитний score, формула (1.2):

$$score = reliable\_ctr\_norm \times 0.6 + gp\_norm \times 0.4, \quad (1.2)$$

де *reliable_ctr_norm* – перцентильний ранг зваженого CTR (CTR × impressions_weight), *gp_norm* – перцентильний ранг валового прибутку, *impressions_weight* = percentile_rank(impressions) – нейтральна вага, що відображає статистичну надійність CTR і не залежить від рішень медіабайєра.

Бінарна мітка визначається порівнянням score з медіанним значенням по навчальній вибірці, формула (1.3):

$$label = 1, \ якщо\ score \geq median(score), \ інакше\ label = 0. \quad (1.3)$$

Таке формулювання цільової змінної забезпечує збалансований розподіл класів (50%/50%) і захищає від оптимізації за хибними сигналами: креативи з високим CTR але нульовим прибутком (clickbait) отримують знижений score через компоненту gp_norm.

Якість моделі оцінюється за такими метриками. Основною метрикою є площа під ROC-кривою (AUC-ROC) [39], що вимірює здатність моделі ранжувати успішні креативи вище неуспішних незалежно від порогу класифікації. AUC = 0,5 відповідає випадковому класифікатору, AUC = 1,0 – ідеальному. Додатково використовуються Accuracy та F1-міра для оцінки якості при фіксованому порозі 0,5. У задачах з нерівномірними витратами хибних рішень також аналізується матриця плутанини – зокрема кількість хибно позитивних (FP) результатів, що відповідає кількості неефективних креативів, яким система помилково присвоїла клас «успішний» [41].

Задача вирішується в умовах таких обмежень. По-перше, обсяг навчальної вибірки є малим за мірками промислових систем: 6 614 зразків після фільтрації, що вимагає застосування стратегій регуляризації та transfer learning для запобігання перенавчанню [29]. По-друге, дані є мультимовними (10 мов), що обумовлює вибір мультилінгвальної текстової моделі [20]. По-третє, метрики ефективності (CTR, GP) є шумними і залежать від зовнішніх факторів – аудиторії, сезонності, конкурентного середовища – що вносить нередукований шум у цільову змінну і обумовлює використання label smoothing [25] як засобу регуляризації.

Очікуваним результатом є система, що складається з таких компонентів: навчена мультимодальна модель класифікації з Test AUC не нижче 0,75; модуль обчислення відносного score 0–100 на основі перцентильного ранжування відносно навчальної вибірки; модуль пояснення рішень на основі Grad-CAM++ і attention weights; а також веб-інтерфейс для практичного використання маркетологом. Питання пояснюваності рішень нейронних мереж є актуальним напрямком досліджень, зокрема в контексті застосування в системах підтримки прийняття рішень — детальний огляд підходів до вилучення правил з нейронних мереж наведено в роботі [44].

Задачі, що вирішуються в роботі, охоплюють: збір та підготовку датасету з реальних рекламних даних Meta Ads; формування та обґрунтування цільової змінної; проектування та навчання мультимодальної архітектури; порівняльне дослідження альтернативних підходів; реалізацію модуля explainability; розробку програмного інтерфейсу для практичного застосування системи.

## Висновки до розділу 1

У першому розділі проведено аналіз предметної галузі, огляд існуючих підходів та методів мультимодального навчання, сформульовано постановку

задачі. Встановлено, що ринок цифрової реклами перевищив 600 млрд доларів [33], а традиційне паралельне тестування є витратним, що обумовлює актуальність передзапускової оцінки креативів. Аналіз промислових досліджень показав, що задача вирішувалась переважно для Alibaba, ByteDance та JD.com [1, 2, 4, 6] – платформа Meta Ads залишається публічно не дослідженою, що визначає наукову новизну роботи. Порівняльний аналіз підтвердив перевагу мультимодальних підходів: CLIP zero-shot (AUC ≈ 0,51) [19] еквівалентний random, тоді як крос-модальна увага [11, 12] з поступовим розморожуванням [28] є обґрунтованим вибором для small-data режиму. Методи Grad-CAM++ [24] і attention weights [15] перетворюють числовий score на actionable рекомендації, а profit-weighted цільова змінна (CTR + GP) є ключовою методологічною новизною, відсутньою в жодній відомій роботі. Результати аналізу слугують теоретичним обґрунтуванням рішень, що описуються в розділі 2.

# РОЗДІЛ 2 РОЗРОБКА СИСТЕМИ ПЕРЕДЗАПУСКОВОЇ ОЦІНКИ ЕФЕКТИВНОСТІ РЕКЛАМНИХ КРЕАТИВІВ

## 2.1 Збір та підготовка датасету

Для навчання системи скорингу використано реальні рекламні дані компанії з платформи Meta Ads (Facebook та Instagram). Абсолютні значення витрат анонімізовано через перцентильне ранжування зі збереженням відносного порядку; показники клікабельності (CTR) та валовий прибуток збережено у вихідному вигляді. Використання реальних рекламних даних є принциповою відмінністю цієї роботи від більшості академічних досліджень, що оперують синтетичними або публічними датасетами [2].

Вихідні дані надійшли у вигляді 468 ZIP-архівів, кожен з яких містив креативи за окремий місяць кампанії та міг сягати 21 ГБ. Розпакування виконувалось у середовищі Google Colab з checkpoint-системою для відновлення після обривів з'єднання: кожен архів спочатку копіювався локально на SSD Colab, після чого розпаковувався – це усунуло нестабільність при читанні великих файлів з Google Drive. У процесі обробки виявлено та виправлено технічні проблеми: некоректне витягування creative_id через тривимірну структуру архівів, суфікси у назвах директорій та сміттєві системні файли. Після розпакування та первинної фільтрації отримано 17 444 статичних зображення у плоскій структурі директорії.

Подальша фільтрація виконувалась на рівні метаданих. Видалено записи без значення CTR – креативи, що не зматчились з рекламними даними через відсутність відповідного запису у внутрішній базі показників. Також виключено записи з недостатньою кількістю показів: CTR при малій аудиторії є статистичним шумом, і записи з impressions_weight нижче 0,05 перцентиля не несуть статистично надійного сигналу. Поріг 0,05 перцентиля обрано емпірично: аналіз розподілу impressions показав, що нижче цього значення дисперсія CTR різко зростає і окремі кліки або їхня відсутність можуть

повністю визначати значення метрики. Включення таких записів до навчальної вибірки вносило б систематичний шум у цільову змінну і знижувало б якість лейблів для всього датасету. Після фільтрації проміжний датасет склав 8 728 записів.

Для кожного креативу формувався текст оголошення – поєднання headline та body copy. Основним джерелом слугувала внутрішня база задач, де креативники прописували повний текст для дизайнерів: для англомовних креативів використовувались оригінальні тексти, для локалізованих версій – відповідні переклади. Це забезпечило покриття для 76,5% записів фінального датасету. Для частини записів, де був наявний лише заголовок без повного тексту, застосовувався Claude Vision API для зчитування тексту безпосередньо із зображення креативу. Записи з виключно headline-текстом (менше 10 слів) були виключені як такі, що несуть недостатньо семантичного контексту для текстового енкодера.

Для усунення encoding artifacts у текстах – некоректно закодованих символів, характерних для багатомовних даних – застосовувалась бібліотека ftfy, інтегрована безпосередньо в датасет-клас PyTorch. Токенізація виконувалась токенізатором XLM-RoBERTa [20] з максимальною довжиною 256 токенів – значення, що покриває 95-й перцентиль довжини рекламних текстів у датасеті і мінімізує втрату інформації при обрізанні.

Після об'єднання зображень і текстових даних, виключення записів без повноцінного тексту та розбиття на вибірки фінальний датасет склав 6 614 записів. Розбиття на навчальну, валідаційну та тестову вибірки виконано у співвідношенні 70% / 15% / 15% зі стратифікацією за міткою класу. Тестова вибірка ізольована від усіх етапів навчання та підбору гіперпараметрів і використовується виключно для фінальної оцінки якості моделі. Статистику датасету наведено в таблиці 2.1.

**Таблиця 2.1** – Статистика фінального датасету

| *Метрика* | *Значення* |
|---|---|
| Вихідних ZIP-архівів | 468 (~21 GB) |
| Статичних зображень після розпакування | 17 444 |
| Після фільтрації за метриками | 8 728 |
| Фінальний датасет (з текстовими даними) | 6 614 |
| Мови креативів | eng, de, fr, it, es, pt, cz, ar, tr, jp |
| Часовий діапазон | 2025-01 – 2026-04 |
| Розбиття train / val / test | 70% / 15% / 15% |
| Баланс класів | 50% / 50% (label=1 / label=0) |

Таким чином, підготовлений датасет є мультимовним, збалансованим за класами і сформованим виключно з реальних рекламних даних з підтвердженими метриками ефективності. Поєднання якісних текстових даних із зображеннями та перевіреними бізнес-мітками забезпечує надійну основу для навчання мультимодальної моделі скорингу.

## 2.2 Збір та підготовка текстових даних

Текстова модальність є рівнозначним із зображенням сигналом у задачі скорингу рекламних креативів: headline формулює ключовий посил до споживача, а body copy розкриває унікальну торгову пропозицію та заклик до дії (CTA). Для навчання мультимодальної моделі кожен запис датасету має містити текст оголошення у формалізованому вигляді. Збір текстових даних виявився нетривіальним завданням через відсутність єдиного централізованого сховища та різнорідність форматів у різних мовних локалізаціях.

Основним джерелом текстів слугувала внутрішня база задач (Ads Board) – таблиця, в якій креативники прописують повний текст оголошення для передачі

дизайнерам. Структура запису включає чотири поля: Headline / Text hook (короткий заголовок, що привертає увагу), Body text / script (основний рекламний текст), Primary Text (альтернативна версія body text) та відповідні локалізовані переклади для кожної мовної версії. Для англомовних креативів формувався текст шляхом конкатенації headline та body text через подвійний перенос рядка. Для локалізованих версій (de, fr, it, es, pt) використовувались перекладені варіанти відповідних полів. Цей підхід забезпечив покриття для 76,5% записів фінального датасету – 9 384 зразки отримали текст безпосередньо з Ads Board.

Для решти 23,5% записів (3 729 зразків) текст у Ads Board був відсутній: аналіз показав, що у 3 726 з них всі текстові поля порожні – тобто креатив існував у базі, але текст не було внесено. Для відновлення текстового сигналу цих записів застосовано Claude Vision API (claude-haiku-4-5): зображення кожного креативу передавалось у вигляді base64-закодованого PNG разом із структурованим промптом, що вимагав повернути витягнутий текст у форматі JSON з полями headline та body. Промпт явно забороняв використання markdown-форматування та вимагав повернути лише текст, видимий на зображенні.

Технічна реалізація pipeline збору тексту через Vision API передбачала паралельну обробку у 5 потоків для скорочення загального часу з 6+ годин до ~45 хвилин. Для забезпечення надійності при можливих збоях мережі або перевищенні квот API реалізовано checkpoint-систему: стан обробки зберігався у JSON-файлі кожні 100 зображень, що дозволяло відновити процес з точки переривання без повторної обробки вже виконаних записів. Зображення зчитувались локально з Google Drive, конвертувались у base64 та передавались в API; відповідь парсилась як JSON і записувалась у master.csv.

Якість тексту, отриманого через Vision API, суттєво нижча порівняно з текстом із Ads Board: рекламні зображення часто містять текст поверх складного фону, накладені ефекти або шрифти нестандартного накреслення, що ускладнює точне розпізнавання. Для фільтрації записів із недостатньо

інформативним текстом застосовано мінімальний поріг якості: записи, де витягнутий текст містить менше 10 слів, класифікуються як непридатні для навчання і виключаються з датасету. Цей поріг обрано емпірично: 10 слів є мінімальною кількістю, за якої токенізатор XLM-RoBERTa генерує достатньо значущих токенів для формування осмисленого представлення речення.

**Таблиця 2.2** – Порівняння двох джерел текстових даних

| *Характеристика* | *Ads Board (основне)* | *Claude Vision API (резервне)* |
|---|---|---|
| Покриття | 76,5% (9 384 зап.) | 23,5% (3 729 зап.) |
| Мови | eng, de, fr, it, es, pt | eng, de, it + решта |
| Якість тексту | Висока (авторський текст) | Середня (OCR з зображення) |
| Структура | headline + body copy окремо | JSON з полями headline та body |
| Довжина (середня) | ~120 слів | ~85 слів |
| Артефакти | Мінімальні | Encoding artifacts (ftfy) |

Датасет є багатомовним: він охоплює 10 мов, що відповідають географічним ринкам рекламних кампаній. Для мов, де Ads Board містив перекладені версії (de, fr, it, es, pt), використовувались відповідні локалізовані тексти. Для мов, де переклади були відсутні (cz, ar, tr, jp), використовувались оригінальні англійські тексти – оскільки рекламний посил у цих кампаніях формулювався англійською і перекладався на рівні візуального оформлення, а не текстового поля.

Багатомовність датасету обумовлює вибір XLM-RoBERTa [20] як текстового енкодера: модель навчена на 100 мовах із спільним токенізатором і здатна формувати семантично узгоджені представлення для текстів на різних мовах у єдиному векторному просторі. Альтернативний підхід – навчання окремого енкодера для кожної мови або обмеження датасету лише англійськими

записами – призвів би до суттєвої втрати обсягу навчальних даних і унеможливив би обробку мультимовних креативів у режимі inference.

Об'єднані тексти з обох джерел містили encoding artifacts – некоректно закодовані символи, що виникають при конвертації між кодуваннями (наприклад, “Itâ€™s time” замість “It’s time” або “So fÃ¼r Sie” замість “So für Sie”). Ці артефакти є типовою проблемою при роботі з багатомовними корпусами, зібраними з різних систем збереження даних [20]. Для їхнього автоматичного виправлення використовувалась бібліотека ftfy (fix text for you), інтегрована безпосередньо в клас Dataset PyTorch і викликається при завантаженні кожного зразка – це гарантує нормалізацію тексту незалежно від того, яке джерело він має.

Токенізація виконувалась токенізатором XLM-RoBERTa з максимальною довжиною 256 токенів. Аналіз довжини текстів у датасеті показав, що 95-й перцентиль відповідає ~240 токенам, тобто значення 256 мінімізує втрату інформації при обрізанні довгих текстів. Коротші тексти доповнюються до фіксованої довжини спеціальним PAD-токеном; маска уваги (attention mask) при цьому виставляється у 0 для PAD-позицій, щоб модель не враховувала їх при обчисленні представлення. CLS-токен використовується як агрегований вектор усього тексту і є входом для fusion-модуля: його прихований стан з останнього шару XLM-RoBERTa розмірністю 768 передається у крос-модальний attention.

Після застосування всіх описаних процедур збору та фільтрації – об'єднання текстів із Ads Board і Vision API, виключення записів із текстом менше 10 слів та нормалізації encoding artifacts – фінальний датасет налічує 6 614 записів, кожен із яких містить повноцінну пару (зображення, текст). Ці дані є вхідними для навчання мультимодальної моделі, описаної в підрозділі 2.5.

## 2.3 Формування цільової змінної

Ключовим методологічним рішенням у побудові системи скорингу є формування цільової змінної – бінарної мітки, що визначає успішність рекламного креативу. Вибір цільової змінної безпосередньо визначає, яку поведінку система навчається передбачати, і тому є не менш важливим, ніж архітектура моделі. У цій роботі цільова змінна пройшла два етапи розробки.

Перша версія (v1) базувалась виключно на показнику клікабельності, зваженому за відносним рангом витрат, формула (2.1):

$$weighted\_ctr = CTR \times spend\_weight, \quad (2.1)$$

де *spend_weight* = percentile_rank(spend). Мітка присвоювалась за порівнянням з медіаною: label = 1 якщо weighted_ctr ≥ median, інакше label = 0. Проте ця версія мала два принципові недоліки. По-перше, spend відображає рішення медіабайєра, а не якість креативу: масштабований посередній креатив отримував spend_weight → 1,0 і завищену мітку. По-друге, CTR як єдина метрика не захищає від clickbait – креатив з CTR 15% але нульовим прибутком отримував label=1.

Друга і фінальна версія цільової змінної (v2) усуває обидва недоліки через заміну spend на impressions як базу зважування CTR та додавання валового прибутку як другої компоненти. Формула (2.2) фінального score:

$$score = reliable\_ctr\_norm \times 0,6 + gp\_norm \times 0,4, \quad (2.2)$$

де *reliable_ctr_norm* = percentile_rank(CTR × impressions_weight), *gp_norm* = percentile_rank(gross_profit.fillna(0)), *impressions_weight* =

percentile_rank(impressions). Бінарна мітка визначається відносно медіани, формула (2.3):

$$label = 1, якщо\ score \geq median(score) = 0{,}4455, інакше\ label = 0. \quad (2.3) \tag{2.3}$$

Impressions є більш нейтральним показником надійності CTR, ніж витрати: кількість показів відображає реальне охоплення аудиторії і не залежить від бюджетних рішень медіабайєра. impressions_weight виконує роль м'якого зважування – креатив з малою кількістю показів отримує низьку вагу і не може штучно підвищити свій score через випадково високий CTR на малій вибірці.

Компонента gp_norm є перцентильним рангом валового прибутку по всій вибірці і знаходиться в діапазоні від 0 до 1. Оскільки абсолютні значення прибутку є конфіденційними даними компанії, у роботі використовується виключно відносна позиція креативу в розподілі: gp_norm = 0,90 означає, що креатив генерував більший прибуток ніж 90% датасету, тоді як gp_norm = 0,23 відповідає нижньому хвосту – типове значення для креативів з нульовим прибутком. Такий підхід дозволяє включити економічний сигнал у цільову змінну без розкриття абсолютних фінансових показників.

З 8 728 записів датасету 46% мають GP = $0 – ці креативи отримують gp_norm ≈ 0,23, що не знищує їхній загальний score, але і не дає бонусу відносно конкурентів. Тільки креативи з підтвердженим прибутком отримують gp_norm вище 0,46, що суттєво підсилює їхній score. Це і є механізм захисту від clickbait-оптимізації: креатив з високим CTR але нульовим прибутком більше не може отримати label=1 виключно за рахунок клікабельності. Ваги 0,6 і 0,4 обрано так, щоб CTR залишався основним сигналом – він присутній у всіх записах датасету – тоді як GP відіграє роль коригуючого фактора там, де є реальна бізнес-цінність.

Перехід від v1 до v2 змінив 15,7% міток датасету (1 373 з 8 728): 604 креативи знижено з label=1 до label=0 – переважно clickbait з високим CTR і

нульовим прибутком; 769 – підвищено з label=0 до label=1 – креативи з помірним CTR але підтвердженим прибутком, що раніше несправедливо отримували негативну мітку. Баланс класів залишився ідеальним 50%/50% завдяки порогу по медіані. Емпіричним підтвердженням якості нових міток є приріст Test AUC на +0,103 при навчанні на v2 мітках порівняно з v1 (0,714 → 0,817) – покращення, яке неможливо пояснити лише архітектурними змінами і свідчить про вищу якість самих навчальних сигналів. Зміни в розподілі міток і приклади типових сценаріїв наведено в таблицях 2.3 і 2.4.

**Таблиця 2.3** – Зміни розподілу міток при переході від v1 до v2

| *Метрика* | *Значення* | *Інтерпретація* |
|---|---|---|
| Всього змінилось міток (v1→v2) | 1 373 (15,7%) | Суттєва корекція без повної зміни датасету |
| label 1→0 (clickbait знижені) | 604 | Високий CTR але GP = $0 – більше не вважаються успішними |
| label 0→1 (конвертуючі підвищені) | 769 | Низький CTR але реальний GP – тепер отримують label=1 |
| Баланс класів після перерахунку | 50% / 50% | Ідеальний баланс завдяки порогу по медіані |
| Приріст Test AUC (CTR-only → CTR+GP) | +0,103 | 0,714 → 0,817 – емпіричне підтвердження якості міток |

Формула score_v2 є композитною бізнес-метрикою, що поєднує статистичну надійність CTR і економічну цінність через компоненту валового прибутку. Підходи до конструювання складених цільових змінних у задачах класифікації з нерівноцінними класами розглядаються також у роботах вітчизняних науковців [43]. Цей підхід до формування цільової змінної відсутній у жодній відомій публічній роботі з оцінки рекламних матеріалів, де цільова змінна обмежується CTR або парою CTR+CVR [1] [4].

**Таблиця 2.4** – Приклади розподілу міток за типом креативу

| *Тип креативу* | *CTR* | *gp_norm* | *Label* | *Пояснення* |
|---|---|---|---|---|
| Clickbait | 15% | 0.23 | 0 | Високий CTR свідчить про привабливий візуал або провокативний заголовок, однак відсутність конверсій (gp_norm ≈ 0,23) суттєво знижує загальний score – label знижено до 0 |
| Конвертуючий | 2% | 0.87 | 1 | Помірний CTR компенсується високим gp_norm, що відображає реальну бізнес-цінність креативу – label підвищено до 1, попри невисоку клікабельність |
| Тестовий (мало показів) | 8% | 0.23 | 0 | Навіть при відносно високому CTR низький impressions_weight зменшує надійність reliable_ctr – статистично ненадійний сигнал отримує низький score |
| Якісний збалансований | 4% | 0.71 | 1 | Поєднання достатнього CTR з підтвердженим прибутком формує high score по обох компонентах формули – найбільш бажаний тип креативу для масштабування |

## 2.4 Методи пояснення рішень моделей

Глибокі нейронні мережі традиційно розглядаються як «чорні скриньки»: вони здатні досягати високої точності передбачень, однак механізм прийняття рішень залишається непрозорим для користувача. У прикладних задачах, де результат моделі безпосередньо впливає на бізнес-рішення – зокрема, на вибір

рекламних матеріалів для запуску з реальним бюджетом – здатність системи пояснити своє рішення є не менш важливою, ніж точність самого передбачення. Маркетолог, який отримує лише числовий score без інтерпретації, не може використати цю інформацію для покращення креативу. Методи пояснення рішень (explainability) вирішують цю проблему, роблячи логіку моделі доступною для людини.

Серед підходів до пояснення рішень нейронних мереж виділяють два основні класи: методи на основі градієнтів і методи на основі уваги (attention-based). Перший клас аналізує, як зміна вхідних ознак впливає на вихід моделі – тобто обчислює градієнт цільового класу відносно вхідного простору. Другий клас використовує ваги уваги трансформерних моделей як проксі для важливості окремих токенів або регіонів. Обидва підходи реалізовані в цій роботі: Grad-CAM++ для візуального компонента і attention weights для текстового.

Grad-CAM (Gradient-weighted Class Activation Mapping), запропонований Selvaraju et al. [23], є одним із найбільш широко застосовуваних методів візуалізації уваги згорткових мереж. Метод обчислює зважену комбінацію карт активацій останнього згорткового шару, де ваги визначаються через глобальне усереднення градієнтів цільового класу відносно відповідних карт активацій. Результатом є теплова карта (heatmap), що накладається на вхідне зображення і показує, які просторові регіони найбільше вплинули на рішення моделі. Grad-CAM є архітектурно-незалежним у межах CNN і не вимагає модифікації моделі або повторного навчання.

Grad-CAM++ є вдосконаленням оригінального методу, запропонованим Chattopadhay et al. [24]. Ключова відмінність полягає у способі обчислення ваг: замість простого глобального усереднення градієнтів Grad-CAM++ використовує зважену комбінацію з урахуванням других похідних і нормалізованих часткових похідних. Це дозволяє точніше локалізувати кілька окремих об'єктів або регіонів на одному зображенні і формує більш чіткі та менш «розмиті» теплові карти порівняно з Grad-CAM. У контексті рекламних

креативів, де на одному зображенні може одночасно бути присутній продукт, обличчя людини, текстовий оверлей і CTA-елемент, точність локалізації є критичною для змістовної інтерпретації рішення моделі. З цієї причини в цій роботі використовується саме Grad-CAM++.

Для кількісної інтерпретації теплової карти в цій роботі запроваджено зональний аналіз: зображення розбивається на рівномірну сітку 3×3 (дев'ять зон), і для кожної зони обчислюється середня інтенсивність теплової карти. Отримані значення нормалізуються відносно середнього і стандартного відхилення по всіх зонах, що дає числовий показник у діапазоні від −1 до +1. Зони з показником вище 0,3 інтерпретуються як такі, що мають сильний позитивний вплив на рішення моделі; нижче нуля – як слабкі або нейтральні. Такий підхід перетворює неструктуровану теплову карту на структурований звіт, придатний для безпосереднього використання креативником або маркетологом.

Паралельно з градієнтними методами, трансформерні моделі надають власний механізм інтерпретації через ваги уваги (attention weights). Bahdanau et al. [15] вперше показали, що ваги механізму м'якої уваги можна інтерпретувати як міру важливості кожного елемента вхідної послідовності для формування поточного виходу. У контексті трансформерних мовних моделей CLS-токен акумулює глобальне представлення всього тексту, і ваги уваги від CLS-токена до решти токенів у останньому шарі відображають відносну важливість кожного слова для фінального передбачення. У цій роботі з останнього шару XLM-RoBERTa витягуються ваги уваги від CLS-токена до всіх текстових токенів, усереднені по головах уваги, що дозволяє ідентифікувати топ-10 слів, яким модель приділила найбільше уваги при оцінці рекламного тексту.

Застосування методів explainability у рекламному домені має практичне обґрунтування, що виходить за межі академічного інтересу. Дослідження Pieters і Wedel [37] встановили, що різні елементи рекламного зображення – обличчя, продукт, текстовий блок – конкурують за обмежену увагу споживача, і їхнє відносне розташування та розмір визначають ефективність сприйняття. Якщо

теплова карта Grad-CAM++ показує, що модель концентрує увагу на периферійних або нерелевантних зонах зображення, це є actionable сигналом для креативника: перекомпонувати зображення так, щоб ключові елементи опинились у зонах природної візуальної уваги. Аналогічно, якщо attention weights виділяють певні слова в тексті як критичні, маркетолог може підсилити або замінити їх у наступній ітерації креативу.

Важливо розрізняти пояснення рішення моделі і пояснення ефективності реклами як такої. Grad-CAM++ показує, на що дивиться модель при прийнятті рішення – але не гарантує, що ці самі елементи привертають увагу реальних споживачів. Зв'язок між активаціями нейронної мережі та людською зоровою увагою є предметом окремих досліджень [38]. Тим не менш, у практичному контексті передзапускової оцінки креативів пояснення на основі Grad-CAM++ надає маркетологу конкретну, просторово локалізовану інформацію про те, які елементи зображення модель вважає діагностичними для класу «успішний креатив» – що є значно більш корисним, ніж числовий score без будь-якої інтерпретації.

Таким чином, поєднання Grad-CAM++ для візуального компонента і attention weights для текстового формує комплексну систему пояснення рішень, яка охоплює обидві модальності вхідних даних. Реалізація цього модуля в контексті системи скорингу рекламних креативів детально описана в підрозділі 2.6.

## 2.5 Архітектура мультимодальної моделі

Запропонована модель є двогілковою мультимодальною мережею: зображення і текст оброблюються незалежними енкодерами, після чого їхні векторні представлення об'єднуються через механізм крос-модальної уваги.

Загальна структура моделі включає три блоки: візуальний енкодер, текстовий енкодер та модуль злиття з класифікатором.

Візуальним енкодером слугує EfficientNet-B3 [16] – згорткова мережа з compound scaling, попередньо навчена на ImageNet (1,2 млн зображень, 1000 класів). EfficientNet-B3 містить 12M параметрів і формує вектор представлення розмірності [1536] на виході глобального пулінгу. Вибір EfficientNet-B3 замість трансформерних альтернатив (ViT, CLIP) обґрунтований результатами порівняльного дослідження: на малих датасетах інуктивний bias CNN – локальна зв'язність та трансляційна інваріантність – забезпечує кращу узагальнюючу здатність, ніж patch-based трансформери, що потребують значно більших обсягів даних для ефективного навчання [17]. Порівняльний аналіз архітектур згорткових мереж для задач класифікації зображень у вітчизняному контексті представлено в [44]. Це підтверджується отриманими результатами: EfficientNet-B3 перевершує ViT-B/16 на 0,031 AUC на тестовій вибірці.

Текстовим енкодером є XLM-RoBERTa-base [20] – мультилінгвальна трансформерна модель, навчена на 2,5 ТБ тексту зі 100 мов. Вибір XLM-RoBERTa обумовлений мультимовністю датасету: рекламні тексти присутні англійською, німецькою, французькою, італійською та іншими мовами. Модель перевершує mBERT на +14,6% за бенчмарком XNLI, що є стандартним показником крос-мовного розуміння. Як вхідне представлення використовується вектор CLS-токена розмірності [768] з останнього шару трансформера – він акумулює глобальне семантичне представлення всього тексту.

Модуль злиття реалізує симетричний механізм крос-модальної уваги. Перед застосуванням уваги обидва вектори проектуються у спільний прихований простір розмірності 512 через лінійні шари: vision_proj: [1536→512] і text_proj: [768→512]. Проекція у спільний простір є принциповою: без неї attention-шар змушений одночасно вирішувати задачу вирівнювання розмірностей і задачу крос-модального зіставлення, що ускладнює навчання і знижує якість результуючих представлень. Далі

виконуються два незалежних attention-шари з 8 головами уваги. Перший (v2t) формує запит з візуального вектора до текстового – зображення «запитує» у тексту, які семантичні ознаки є найбільш релевантними. Другий (t2v) формує запит у зворотному напрямку – текст «запитує» у зображення, які візуальні зони узгоджуються з рекламним посилом. Симетрична двонаправлена схема є принциповою відмінністю від простої конкатенації: кожна модальність отримує контекст від іншої, що дозволяє моделі виявляти взаємну відповідність між продуктом на зображенні і його описом у тексті – саме цей сигнал відрізняє конвертуючий креатив від clickbait. Після кожного attention-шару застосовується residual-з'єднання та LayerNorm для стабілізації навчання [14], що запобігає деградації градієнтів і зберігає оригінальне представлення кожної модальності.

Результати обох attention-шарів конкатенуються у вектор розмірності [1024] і подаються на MLP-класифікатор: Linear(1024→256) → ReLU → Dropout(0,3) → Linear(256→64) → ReLU → Dropout(0,3) → Linear(64→1). Вихід класифікатора пропускається через функцію Sigmoid, що перетворює логіт у ймовірність $P(\text{label}=1) \in [0, 1]$. Dropout із коефіцієнтом 0,3 виконує роль регуляризатора і є критично важливим при малому датасеті для запобігання перенавчанню fusion-голови.

Загальна кількість параметрів моделі складає 290M, з яких переважна більшість належить XLM-RoBERTa (278M). Повне одночасне навчання всіх параметрів на датасеті з 6 614 зразків призвело б до catastrophic overfitting. Стратегія поступового розморожування, що вирішує цю проблему, описана в підрозділі 2.6. Вибір архітектурних компонентів підтверджується результатами аблаційного дослідження [10] і наведено на рисунку 2.1.

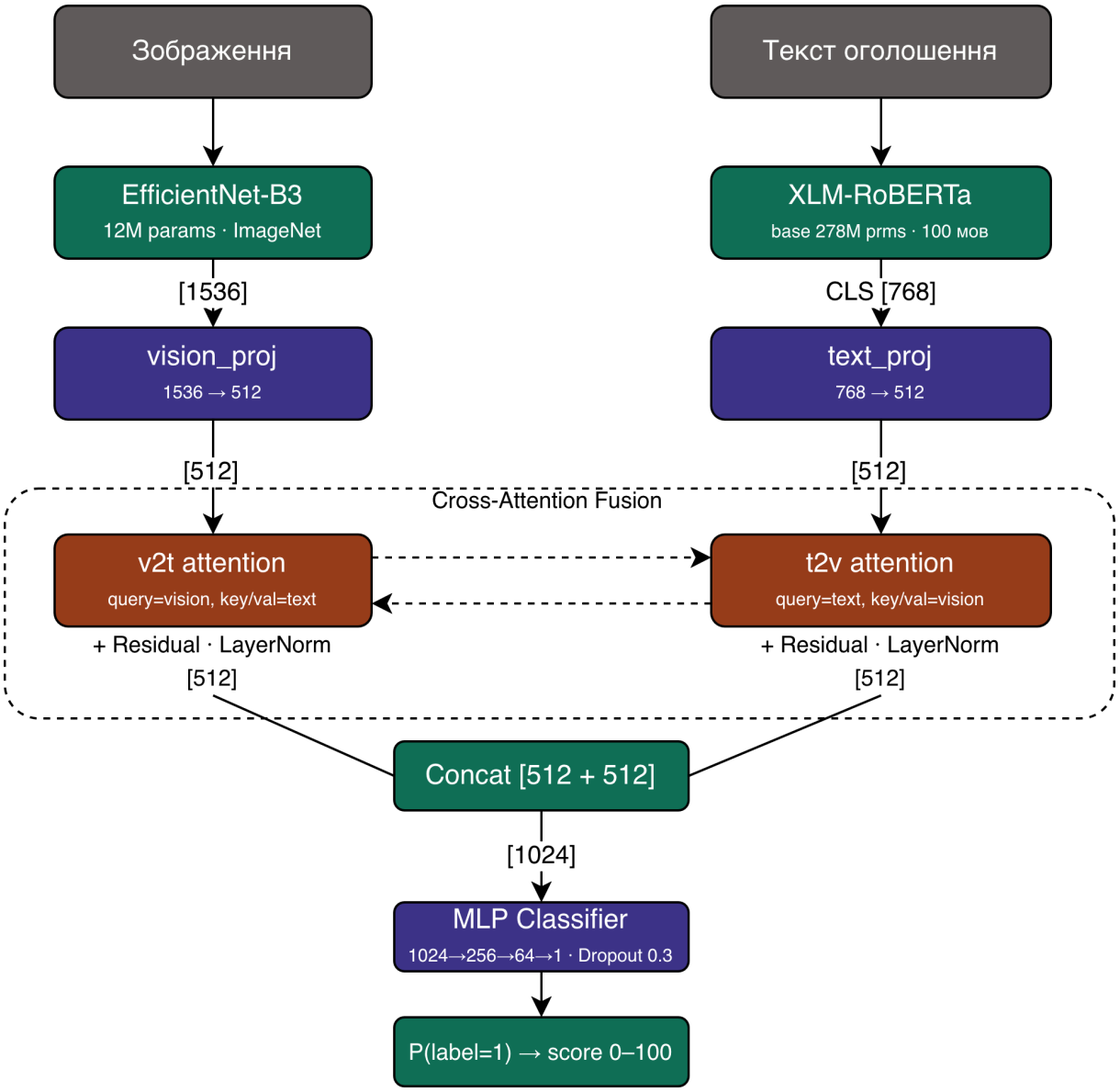


**Рисунок 2.1** – Архітектура мультимодальної моделі CreativeScorer

Запропонована архітектура є результатом послідовної еволюції: перша ітерація використовувала просту конкатенацію векторів (concat fusion) і заморожені енкодери, що дало Test AUC 0,714. Перехід до крос-модальної уваги разом з покращеною стратегією навчання підвищив результат до AUC 0,810–0,817. Ця різниця є емпіричним підтвердженням переваги attention-based fusion над конкатенацією для задачі мультимодального скорингу рекламних креативів, що узгоджується з теоретичними висновками роботи Kiela et al. [10].

## 2.6 Процес навчання моделі

Навчання моделі вирішує принципову суперечність між великою кількістю параметрів (290M) і малим обсягом навчальних даних (4 629 зразків у train split). Для подолання цієї суперечності застосовано комплекс технік: стратегію поступового розморожування енкодерів, регуляризацію через label smoothing, збалансоване семплювання та ранню зупинку.

Loss-функцією є Label Smoothing Binary Cross-Entropy [25] з коефіцієнтом згладжування 0,1. Замість жорстких міток {0, 1} модель навчається на м'яких: label=0 перетворюється на 0,05, label=1 – на 0,95. Це запобігає overconfidence – стану, коли модель присвоює надміру впевнені передбачення на шумних навчальних мітках. Враховуючи, що цільова змінна score_v2 формується на основі реальних рекламних метрик з природним шумом, label smoothing є особливо доречним і покращує калібрацію виходів моделі.

Для гарантованого балансу класів у кожному батчі використовується WeightedRandomSampler [31]. Незважаючи на те, що загальний баланс датасету є близьким до 50%/50%, без явного зважування окремі батчі можуть мати суттєвий дисбаланс, що дестабілізує навчання. Семплер призначає кожному зразку вагу, обернено пропорційну частоті його класу, забезпечуючи рівномірне представлення обох класів у кожному батчі розміром 16.

Центральною технікою навчання є поступове розморожування шарів енкодерів (gradual unfreezing) [28]. Концепція базується на ієрархічній природі представлень у глибоких мережах [29]: нижні шари кодують універсальні низькорівневі ознаки (краї, текстури, токени), тоді як верхні шари є більш задачо-специфічними. Відповідно, адаптація до рекламного домену потребує донавчання переважно верхніх шарів, тоді як нижні залишаються практично незмінними.

Процес розморожування складається з трьох стадій. На стадії 0 (епохи 1–5) тренуються виключно параметри fusion MLP і проекційних шарів (3,5M

параметрів). Це дозволяє fusion-голові навчитись комбінувати фіксовані представлення енкодерів перед будь-якою їх адаптацією. На стадії 1 (епохи 6–10) розморожуються два останніх шари XLM-RoBERTa (+14M параметрів) – це дає найбільший приріст Val AUC (+0,08), оскільки текстовий енкодер адаптується до специфіки рекламних текстів. На стадії 2 (епохи 11+) розморожуються два останніх шари EfficientNet-B3 (+8M параметрів), що дозволяє візуальному енкодеру адаптуватись до рекламних зображень. Динаміку стадій розморожування зображено на рисунку 2.2.

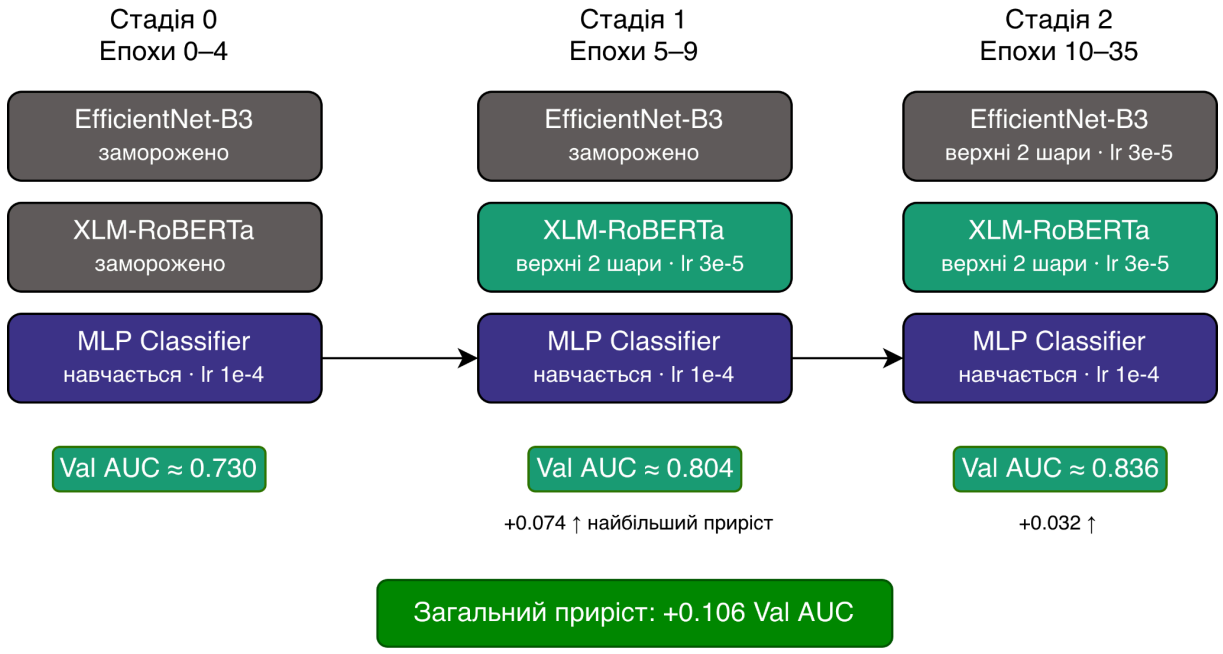


**Рисунок 2.2** – Стратегія поступового розморожування шарів (gradual unfreezing)

Для розморожених шарів енкодерів застосовується знижений learning rate $3\times10^{-5}$ порівняно з $1\times10^{-4}$ для fusion MLP. Диференційований learning rate запобігає catastrophic forgetting – явищу, коли нові градієнти надто різко модифікують попередньо навчені ваги і руйнують узагальнені представлення. Розклад learning rate визначається CosineAnnealingLR [27], що плавно зменшує lr від початкового значення до нуля за косинусним законом протягом 35 епох –

це покращує збіжність на фінальних стадіях навчання порівняно з фіксованим lr.

Оптимізатором є AdamW [26] з weight_decay = 0,01. AdamW відрізняється від Adam тим, що застосовує weight decay безпосередньо до ваг, а не через градієнтний момент, що забезпечує більш коректну L2-регуляризацію і кращу узагальнюючу здатність. Gradient clipping зі значенням 1,0 запобігає вибуху градієнтів при розморожуванні нових шарів.

Рання зупинка (early stopping) [30] з patience = 7 епох моніторить Val AUC і зупиняє навчання, якщо протягом 7 послідовних епох покращення не спостерігається. Зберігається checkpoint з найкращим Val AUC – саме ці ваги завантажуються для фінальної оцінки на тестовій вибірці. Checkpoint система синхронізується з Google Drive після кожної епохи, що забезпечує збереження прогресу при обривах сесії.

Аугментації зображень застосовуються виключно до навчальної вибірки: горизонтальне відзеркалення (p = 0,5) та варіація кольорів ColorJitter (яскравість, контраст, насиченість ±20%). Валідаційна і тестова вибірки обробляються лише через Resize(224×224) і нормалізацію за статистиками ImageNet, що гарантує об'єктивність оцінки. Повний перелік гіперпараметрів наведено в таблиці 2.5.

**Таблиця 2.5** – Гіперпараметри навчання моделі

| *Параметр* | *Значення* |
|---|---|
| Loss функція | Label Smoothing BCE (smoothing = 0,1) |
| Optimizer | AdamW (weight_decay = 0,01) |
| Learning rate – fusion MLP | $1\times10^{-4}$ |
| Learning rate – розморожені шари енкодерів | $3\times10^{-5}$ |
| LR Scheduler | CosineAnnealingLR (T_max = 35) |
| Batch size | 16 |
| Gradient clipping | 1,0 |

**Кінець табл. 2.5**

| *Параметр* | *Значення* |
|---|---|
| Early stopping (patience) | 7 епох без покращення Val AUC |
| Максимум епох | 35 |
| Аугментації (train) | RandomHorizontalFlip (p=0,5), ColorJitter (±20%) |
| Нормалізація зображень | ImageNet mean/std: [0.485, 0.456, 0.406] / [0.229, 0.224, 0.225] |

Описана стратегія навчання забезпечила ефективну адаптацію попередньо навчених енкодерів до рекламного домену без перенавчання на малому датасеті. Поступове розморожування дало сукупний приріст Val AUC з ~0,73 (тільки fusion MLP) до 0,836 – приріст 0,106 виключно за рахунок адаптації існуючих представлень. Це підтверджує ефективність transfer learning як стратегії для small-data режиму [29].

## 2.7 Інтерпретація рішень моделі

Числовий score сам по собі має обмежену практичну цінність для маркетолога: він повідомляє про очікувану ефективність креативу, але не пояснює чому модель прийняла таке рішення і що саме можна покращити. Забезпечення інтерпретованості рішень є важливою складовою систем машинного навчання у прикладних задачах [44]. Система пояснення рішень зображена на рисунку 2.3.

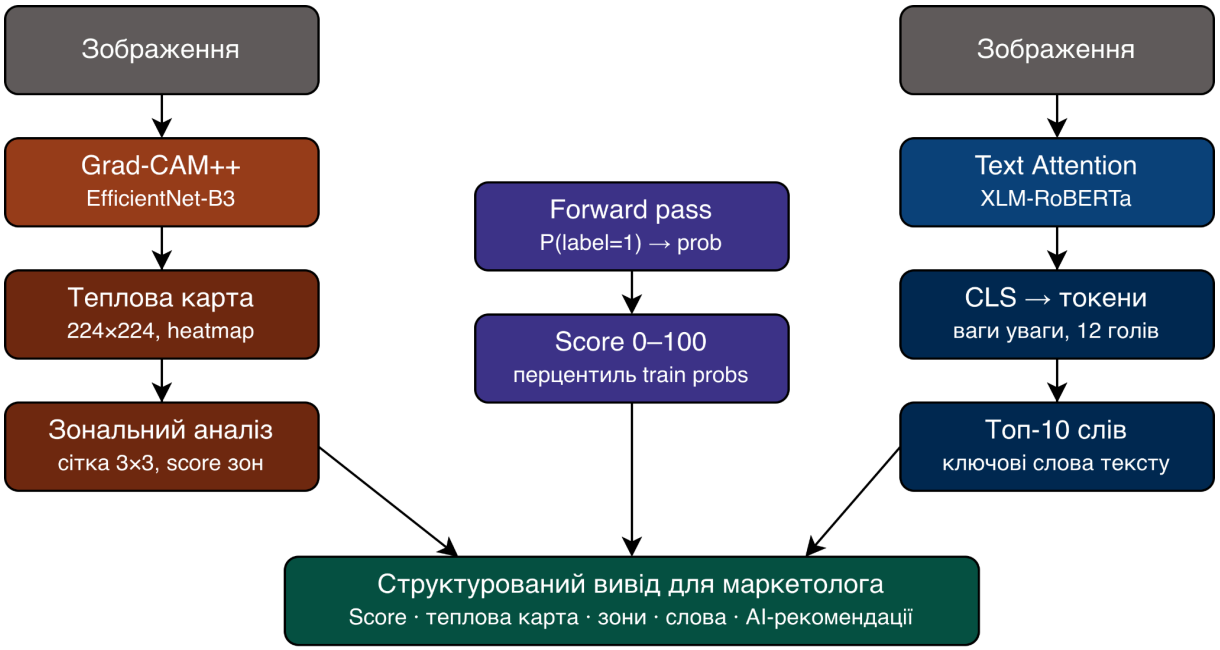


**Рисунок 2.3** – Система пояснення рішень моделі

Сирий вихід моделі – ймовірність P(label=1) ∈ [0, 1] – перетворюється у відносний score 0–100 через порівняння з розподілом ймовірностей навчального датасету. Спочатку модель запускається на всіх 4 629 зразках навчальної вибірки і зберігається масив train_probs. Для нового креативу обчислюється за формулою (2.4):

$$score = round(\, P(train\_probs < prob) \times 100 \,), \tag{2.4}$$

де prob – ймовірність нового креативу. Score 74 означає, що цей креатив має вищу ймовірність успіху ніж 74% креативів навчального датасету. Такий підхід є більш інтерпретованим для маркетолога, ніж абсолютна ймовірність: він дає контекст відносно реального розподілу рекламних матеріалів. Масив train_probs зберігається на Google Drive і не перераховується при кожному запуску системи.

Для пояснення рішення на рівні зображення застосовується Grad-CAM++ [24]. Цільовим шаром є останній блок EfficientNet-B3

(model.vision_encoder.blocks[-1]) – шар, що кодує найбільш семантично значущі ознаки зображення. Оскільки CrossAttentionFusion змінює стандартний forward pass моделі, для сумісності з pytorch_grad_cam реалізовано обгортку VisionWrapper, що ізолює візуальну гілку для обчислення градієнтів. Результатом є теплова карта розміром 224×224, яка накладається на вхідне зображення: теплі кольори відповідають зонам з високою активацією, холодні – зонам, що не вплинули на рішення.

Для структурованої інтерпретації теплової карти реалізовано зональний аналіз: зображення розбивається на рівномірну сітку 3×3 з дев'ятьма зонами (верх-ліво, верх-центр, верх-право, центр-ліво, центр, центр-право, низ-ліво, низ-центр, низ-право). Для кожної зони обчислюється середня інтенсивність теплової карти, після чого значення нормалізуються відносно глобального середнього і стандартного відхилення по всіх дев'яти зонах, формула (2.5):

$$zone\_score = clip( (mean\_zone - global\_mean) / global\_std, -1, 1 ). \quad (2.5)$$

Результуючий показник знаходиться в діапазоні від −1 до +1 і відображає відносний вплив кожної просторової зони на рішення моделі порівняно із середнім по зображенню. Інтерпретацію значень zone_score наведено в таблиці 2.6.

**Таблиця 2.6** – Інтерпретація значень зонального аналізу

| *Значення zone_score* | *Рівень впливу* | *Інтерпретація для маркетолога* |
|---|---|---|
| ≥ 0,3 | Сильний | Зона суттєво вплинула на рішення моделі – ключовий візуальний елемент, що визначає оцінку креативу |
| 0 до 0,3 | Помірний | Зона присутня у полі уваги моделі, але не є визначальною |
| < 0 | Слабкий | Зона практично не вплинула на рішення – модель її ігнорувала при оцінці |

Для пояснення на рівні тексту використовуються ваги уваги останнього шару XLM-RoBERTa [15]. CLS-токен у трансформерних моделях акумулює глобальне представлення всього тексту, і ваги його уваги до решти токенів відображають відносну важливість кожного слова. Реалізація витягує матрицю уваги з останнього шару (output.attentions[-1]), виділяє рядок CLS-токена (attn[0, :, 0, :]) та усереднює по всіх головах уваги. Результатом є вектор важливості для кожного токена, з якого виводяться топ-10 слів з найвищими значеннями. Це дозволяє маркетологу зрозуміти, які слова тексту оголошення модель вважала найбільш значущими при формуванні передбачення.

Усі чотири компоненти об'єднуються у структурований вивід для маркетолога: числовий score 0–100 з колірним індикатором (червоний / жовтий / зелений), теплова карта Grad-CAM++ накладена на оригінальне зображення, зональна сітка 3×3 з числовими значеннями та кольоровим кодуванням і список топ-10 ключових слів тексту. Разом ці компоненти дають маркетологу конкретну, просторово локалізовану і семантично обґрунтовану інформацію – на відміну від нечисленних існуючих систем скорингу, що надають лише числовий результат [1].

## 2.8 Програмна реалізація та інтерфейс

Вся програмна реалізація виконана у середовищі Google Colab на Python 3.12 з використанням GPU T4 або L4 для навчання і CPU для inference та аналітики. Google Colab обрано через безкоштовний доступ до GPU, інтеграцію з Google Drive і зручність демонстрації через інтерактивні ноутбуки. Код організовано у вигляді окремих ноутбуків за функціональними блоками: підготовка даних, навчання моделі та inference з інтерфейсом.

Основні компоненти програмної реалізації включають клас моделі CreativeScorer, що об'єднує vision_encoder (EfficientNet-B3 через timm),

text_encoder (XLM-RoBERTa через HuggingFace Transformers) та fusion (CrossAttentionFusion). Клас CreativeDataset реалізує завантаження зображень, токенізацію тексту і нормалізацію encoding artifacts через ftfy в методі __getitem__. Checkpoint-система зберігає стан моделі, оптимізатора та scheduler після кожної епохи і синхронізує з Google Drive, що забезпечує відновлення після обривів сесії.

Для сумісності Grad-CAM++ з мультимодальною архітектурою реалізовано клас VisionWrapper, що ізолює візуальну гілку і надає стандартний інтерфейс forward(image) → logit, якого очікує pytorch_grad_cam – це дозволяє обчислювати градієнти відносно активацій EfficientNet-B3 в контексті повної мультимодальної моделі.

Веб-інтерфейс реалізовано на Gradio [42] з використанням gr.Blocks і темою Soft. Gradio обрано через простоту розгортання в середовищі Google Colab: виклик demo.launch(share=True) генерує публічне посилання без необхідності окремого деплою. Інтерфейс організовано у вигляді єдиної вертикальної сторінки з чотирма блоками: введення даних (зображення та текст оголошення), score з вердиктом і рекомендацією щодо запуску, візуальний аналіз (Grad-CAM++ і YOLO) та детальний аналіз з attention weights і AI-рекомендаціями щодо покращення креативу. Зовнішній вигляд інтерфейсу з реальним прикладом оцінювання наведено на рисунку 2.4.

Pipeline inference охоплює шість кроків: нормалізація тексту (ftfy), підготовка зображення (Resize 224×224, Normalize), токенізація (max_length=256), forward pass, конвертація ймовірності в score 0–100 і паралельний запуск Grad-CAM++ з витягуванням attention weights. Весь pipeline виконується за 1–5 секунд. Повний технічний стек системи наведено в таблиці 2.7.

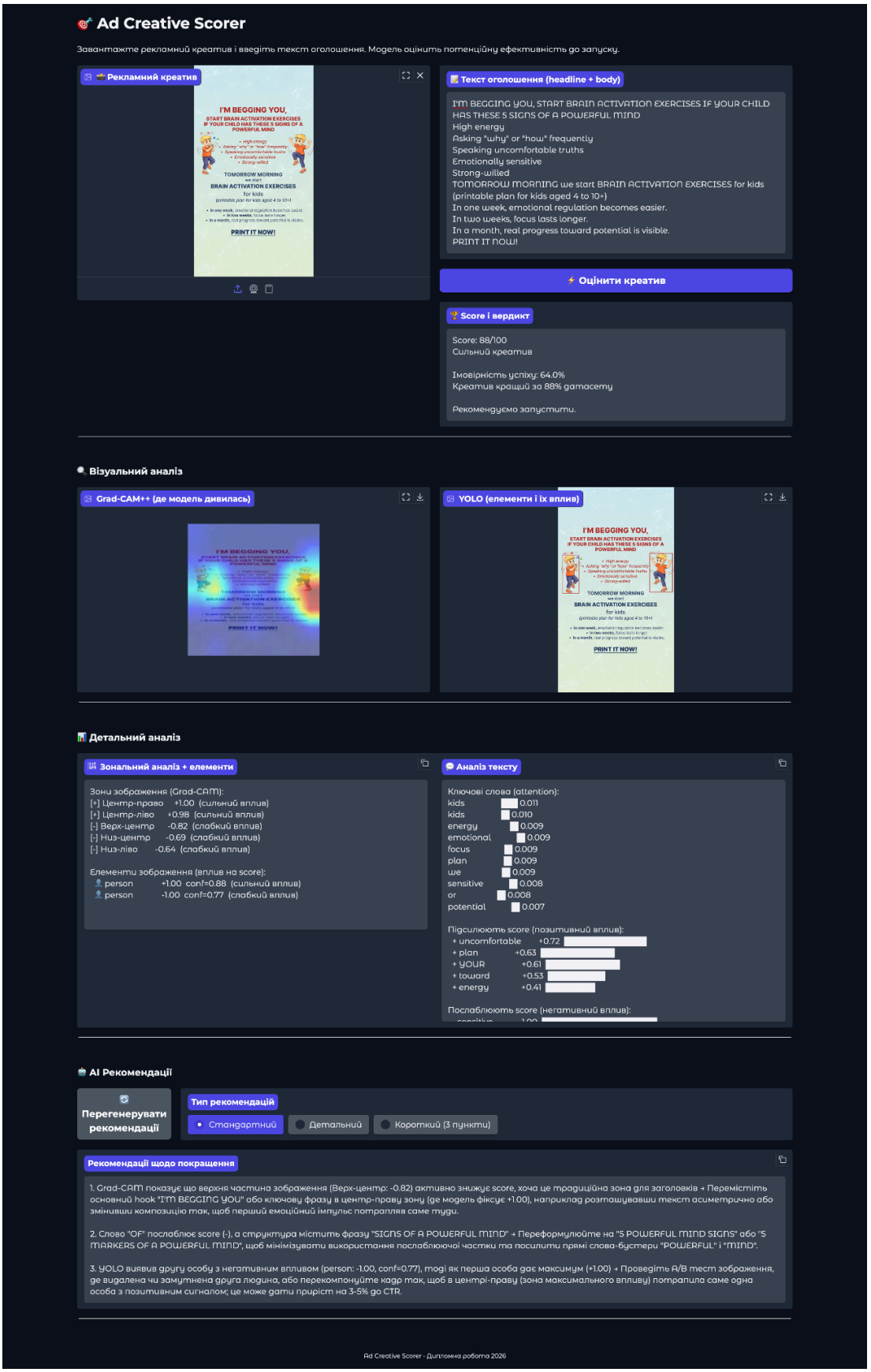


**Рисунок 2.4** – Веб-інтерфейс

**Таблиця 2.7** – Технічний стек системи скорингу рекламних креативів

| Категорія | Бібліотеки / інструменти |
|---|---|
| Мова програмування | Python 3.12 |
| Deep Learning Framework | PyTorch 2.x |
| Vision моделі | timm (EfficientNet-B3, ViT-B/16, ResNet-50) |
| Text моделі | HuggingFace Transformers (XLM-RoBERTa-base) |
| CLIP zero-shot | openai/CLIP (ViT-B/32) |
| Explainability | grad-cam (pytorch_grad_cam – GradCAMPlusPlus) |
| Текстова нормалізація | ftfy (fix text for you) |
| Обробка даних | pandas, numpy, scikit-learn |
| Веб-інтерфейс | Gradio 4.x |
| Середовище виконання | Google Colab (GPU T4/L4) |
| Сховище даних та моделей | Google Drive (1 TB) з checkpoint-синхронізацією |

Вибір кожного компонента стеку обґрунтований конкретними вимогами задачі: timm забезпечує уніфікований доступ до претренованих vision-моделей з можливістю заміни backbone без зміни pipeline; HuggingFace Transformers – стандартний інтерфейс для мультилінгвальних моделей; grad-cam – готова реалізація Grad-CAM++ без необхідності написання власного коду зворотного поширення для активацій; Gradio – мінімальні витрати на розгортання демонстраційного інтерфейсу в академічному середовищі.

## Висновки до розділу 2

У другому розділі розроблено метод скорингу рекламних креативів: підготовлено реальний датасет з 6 614 мультимовних креативів Meta Ads з підтвердженими метриками ефективності та checkpoint-системою

відтворюваності. Розроблено двокомпонентну цільову змінну score = reliable_ctr_norm × 0,6 + gp_norm × 0,4 (формула 2.2), перехід до якої змінив 15,7% міток і забезпечив приріст Test AUC на +0,103, захистивши від clickbait-оптимізації. Запропоновано мультимодальну архітектуру CrossAttentionFusion на базі EfficientNet-B3 [16] і XLM-RoBERTa [20] з двонаправленим attention (v2t, t2v) і MLP-класифікатором; для подолання суперечності між 290M параметрів і малим датасетом застосовано gradual unfreezing [28], що дало приріст Val AUC 0,106, а також label smoothing [25] і early stopping [30]. Реалізовано комплексну систему пояснення рішень – Grad-CAM++ [24] із зональним аналізом 3×3, attention weights [15] і перцентильний score 0–100 – та веб-інтерфейс на Gradio [42] з повним inference pipeline за 1–5 секунд. Результати порівняльного аналізу архітектур розглядаються в розділі 3.

# РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНА ОЦІНКА МУЛЬТИМОДАЛЬНОЇ СИСТЕМИ СКОРИНГУ РЕКЛАМНИХ КРЕАТИВІВ

## 3.1 Порівняльний аналіз архітектур

Для оцінки запропонованого підходу проведено порівняльне дослідження шести конфігурацій на єдиній тестовій вибірці (993 зразки). Всі моделі тестувались за однакових умов: фіксований поріг класифікації 0,5, єдина тестова вибірка, ізольована від навчання. Основною метрикою є AUC-ROC [39] як показник здатності моделі ранжувати успішні креативи вище неуспішних незалежно від порогу. Додатково використовуються Accuracy, F1 і матриця плутанини для аналізу структури помилок.

Випадковий класифікатор (Random baseline) дає AUC = 0,500 – теоретична нижня межа для збалансованого датасету. CLIP zero-shot [19], навчений контрастивно на 400 млн пар зображення-текст, досягає AUC = 0,512 – статистично еквівалентний результат. Чотири варіанти промптів давали AUC від 0,456 до 0,512. Цей результат є принциповим: він підтверджує, що загальні візуально-мовні моделі без навчання на реальних рекламних метриках не здатні розрізнити успішні та неуспішні креативи. Семантична подібність зображення і тексту, яку вимірює CLIP, є ортогональною до поняття «ефективний рекламний креатив», яке визначається бізнес-метриками аудиторії.

Модель v3 (EfficientNet-B3 + XLM-RoBERTa, CTR-only лейбли) досягає AUC = 0,714 – стрибок на +0,202 відносно CLIP zero-shot. Це підтверджує, що навіть при субоптимальній цільовій змінній мультимодальна архітектура з domain-specific навчанням значно перевершує загальні підходи. Однак Precision = 0,634 означає, що при кожних двох рекомендованих креативах одна рекомендація є хибно позитивною – FP = 200 на тестовій вибірці.

Заміна vision encoder на CLIP Vision (заморожений) з тим самим XLM-RoBERTa і fusion дає AUC = 0,766. Незважаючи на масштаб попереднього навчання CLIP, EfficientNet-B3 перевершує його на задачі скорингу рекламних

креативів – convolutional inductive bias виявляється кориснішим за patch-based трансформер при малому датасеті. ViT-B/16 + XLM-RoBERTa досягає AUC = 0,786 – кращий результат серед альтернативних vision encoders, але все ж поступається фінальній моделі на 0,024 AUC при більшому числі параметрів.

Фінальна модель v7 (EfficientNet-B3 + XLM-RoBERTa + Cross-Attention, CTR+GP лейбли) досягає Test AUC = 0,810, Accuracy = 0,739, F1 = 0,739, Precision = 0,759. Порівняно з v3 на тій самій архітектурі приріст AUC складає +0,096 – і цей приріст пояснюється виключно зміною цільової змінної, а не архітектурою. Результати всіх моделей наведено в таблиці 3.1.

**Таблиця 3.1** – Порівняльний аналіз моделей на тестовій вибірці

| *Модель* | *Test AUC* | *Accuracy* | *F1* | *Примітка* |
|---|---|---|---|---|
| Random baseline | 0,500 | 0,513 | – | Нижня межа – монетка |
| CLIP zero-shot | 0,512 | 0,514 | – | Generic VLM без навчання ≈ random |
| v3: EfficientNet-B3 + XLM-R (CTR-only) | 0,714 | 0,643 | 0,661 | Та сама архітектура, старі CTR-only лейбли |
| CLIP Vision + XLM-RoBERTa | 0,766 | 0,692 | 0,682 | CLIP encoder замість EfficientNet |
| ViT-B/16 + XLM-RoBERTa | 0,786 | 0,737 | 0,732 | Трансформерний vision encoder |
| v7: EfficientNet-B3 + XLM-R (CTR+GP) ★ | 0,810 | 0,739 | 0,739 | Фінальна модель – найкращий результат |

Порівняння v3 і v7 при ідентичній архітектурі дозволяє ізолювати ефект якості лейблювання. Обидві моделі використовують EfficientNet-B3 + XLM-RoBERTa + Cross-Attention Fusion і навчались за однаковим протоколом. Єдина відмінність – цільова змінна: v3 навчався на CTR-only мітках, v7 – на CTR+GP мітках. Приріст AUC на +0,096 між цими двома моделями є прямою мірою впливу якості лейблювання на результат.

Механізм покращення пояснюється структурою помилок. CTR-only лейбли присвоюють label=1 креативам з високою клікабельністю незалежно від реального прибутку. Модель v3, навчена на таких мітках, навчається ідентифікувати ознаки clickbait – яскраві, провокативні зображення і сенсаційні заголовки. CTR+GP лейбли виправляють цю систематичну помилку: 604 clickbait креативи отримали label=0, і модель v7 вже не вважає їх успішними. Натомість 769 креативів з помірним CTR але підтвердженим прибутком отримали label=1, що навчило модель виявляти ознаки конвертуючих матеріалів. Детальне порівняння наведено в таблиці 3.2.

**Таблиця 3.2** – Вплив якості лейблювання на метрики тестової вибірки

| *Метрика* | *v3 (CTR-only)* | *v7 (CTR+GP)* | *Різниця* |
|---|---|---|---|
| Test AUC | 0,714 | 0,810 | +0,096 (+13,5%) |
| Test Accuracy | 0,643 | 0,739 | +0,096 (+14,9%) |
| F1-score | 0,661 | 0,739 | +0,078 (+11,8%) |
| Precision | 0,634 | 0,759 | +0,125 (+19,7%) |
| False Positive (FP) | 200 | 116 | −84 (−42%) |
| False Negative (FN) | 155 | 143 | −12 (−8%) |

Найбільш значущою практичною перевагою є скорочення False Positive з 200 (v3) до 116 (v7) – зменшення на 42%. FP відповідають креативам, яким модель помилково присвоює label=1 – тобто неефективним матеріалам, які маркетолог запустить у кампанію на підставі рекомендації системи. Скорочення FP на 84 креативи означає суттєву практичну різницю у витратах рекламного бюджету. FN (пропущені успішні креативи) також скоротились: з 155 (v3) до 143 (v7) – модель v7 одночасно точніша і повніша. Матрицю плутанини фінальної моделі наведено в таблиці 3.3.

**Таблиця 3.3** – Матриця плутанини фінальної моделі v7 (тестова вибірка, поріг 0,5)

| | *Передбачено: 0 (невдалий)* | *Передбачено: 1 (успішний)* |
|---|---|---|
| *Реально: 0 (невдалі)* | TN = 368 | FP = 116 ← марні запуски |
| *Реально: 1 (успішні)* | FN = 143 ← пропущені хороші | TP = 366 |

Результати порівняльного аналізу дозволяють зробити три ключові висновки. По-перше, domain-specific навчання є необхідною умовою: CLIP zero-shot = random, тоді як навчена модель дає AUC = 0,810. По-друге, EfficientNet-B3 перевершує ViT і CLIP Vision на малому датасеті, підтверджуючи перевагу convolutional архітектур у low-data режимі [29]. По-третє, і найголовніше: якість цільової змінної є визначальним фактором – +0,096 AUC від переходу CTR-only → CTR+GP перевищує приріст від будь-якої архітектурної зміни в даному дослідженні.

## 3.2 Вплив якості лейблювання на результати навчання

Якість цільової змінної є фундаментальним чинником ефективності моделей машинного навчання: якщо мітки класів не відповідають реальній природі явища, жодна архітектурна складність не здатна компенсувати шум у навчальних сигналах. У цій роботі це твердження перевірено емпірично: дві ідентичні за архітектурою, гіперпараметрами та навчальним датасетом моделі навчались на мітках різної якості – v1 (CTR-only) та v2 (CTR+GP). Єдиною відмінністю між ними були самі мітки, що дозволяє ізолювати вплив якості лейблювання від інших факторів і коректно атрибутувати різницю в результатах саме зміні цільової змінної.

Версія v1 формувала цільову змінну виключно на основі CTR, зваженого перцентильним рангом витрат (spend_weight). Цей підхід має два структурні недоліки. По-перше, spend є рішенням медіабайєра, а не властивістю креативу: медіабайєр може масштабувати бюджет на посередньому матеріалі, штучно підвищуючи spend_weight і, відповідно, зважений CTR. По-друге, CTR вимірює виключно клікабельність і не розрізняє креативи, що залучають кліки, від тих, що генерують реальний прибуток. Як наслідок, v1 систематично присвоювала label=1 clickbait-матеріалам – яскравим, провокативним оголошенням із CTR 10–15% і нульовим прибутком.

Версія v2 усуває обидва недоліки: вага надійності CTR замінена з spend на impressions (кількість показів), яка не залежить від рішень медіабайєра і безпосередньо вимірює охоплення; до score додається компонента валового прибутку (gp_norm) з вагою 0,4. Ваги 0,6 і 0,4 обрано так, щоб CTR залишався основним сигналом – він присутній у всіх 8 728 записах датасету – тоді як GP відіграє роль коригуючого фактора для 54% записів, де gp_norm > 0,23. Порівняння двох версій лейблювання наведено в таблиці 3.4.

**Таблиця 3.4** – Порівняння версій лейблювання v1 та v2

| *Характеристика* | *v1 (CTR-only)* | *v2 (CTR+GP)* | *Зміна* |
|---|---|---|---|
| Основний сигнал | Зважений CTR(CTR × spend_weight) | Зважений CTR(CTR × impressions_weight) | Нейтральніша вага |
| Вага надійності | spend (рішення медіабайєра) | impressions (охоплення) | Незалежна від бюджету |
| Захист від clickbait | Відсутній | GP-компонента (вага 0,4) | Доданий |
| Баланс класів | 52% / 48% | 50% / 50% | Ідеальний баланс |
| Поріг (медіана score) | 0.50 (CTR-based) | 0.4455 | Перераховано |
| Змінено міток | – | 1 373 (15,7%) | Суттєва корекція |

Перехід від v1 до v2 змінив 1 373 мітки з 8 728 (15,7% датасету): 604 креативи знижено з label=1 до label=0 і 769 – підвищено з label=0 до label=1. Перша група – це переважно clickbait: матеріали з CTR 3–15% і нульовим прибутком, які v1 вважала успішними виключно за рахунок клікабельності. Друга група – конвертуючі креативи з помірним CTR (0,5–2%) але підтвердженим валовим прибутком, що v1 несправедливо відносила до невдалих. Конкретні приклади наведено в таблиці 3.5.

**Таблиця 3.5** – Приклади зміни міток при переході v1 → v2

| *Тип креативу* | *CTR* | *GP* | *Label v1* | *Label v2* | *Пояснення* |
|---|---|---|---|---|---|
| Clickbait | 3,01 % | $0 | 1 | 0 | Великий spend дав label=1 у v1; GP=0 знизив score у v2 |
| Конвертуючий | 0,63 % | $1 634 | 0 | 1 | Низький CTR не враховував GP; у v2 gp_norm підняв score до label=1 |
| Якісний збалансований | 4,0% | $3 000 | 1 | 1 | Обидві версії збігаються; достатній CTR і GP |
| Статистично ненадійний | 8,0% | $0 | 0 | 0 | Мало показів: impressions_weight знижує надійність CTR |

Порівняння результатів двох моделей виявляє характерний парадокс між валідаційним і тестовим AUC. Модель v3, навчена на v1-мітках, демонструє вищий Val AUC під час навчання (0,836 проти 0,810), але суттєво нижчий Test AUC на нових даних (0,714 проти 0,810). Цей феномен пояснюється природою самих міток: CTR-only мітки є простішими і більш шумними – модель v3 легко підлаштовується під них протягом навчання, демонструючи оптимістичний Val AUC, але ця «легкість» пов’язана саме з тим, що мітки не відображають реальну якість креативу. На нових даних v3 погано узагальнює, оскільки фактично навчилась передбачати «шумний CTR», а не реальну ефективність. Модель v7, навчена на складніших CTR+GP мітках, вирішує важчу задачу, що пригнічує Val

AUC, але ця задача краще відповідає реальному розподілу якості – тому тестовий AUC виявляється значно вищим. Порівняльний аналіз моделей наведено в таблиці 3.6.

**Таблиця 3.6** – Порівняння метрик моделей v3 (CTR-only) та v7 (CTR+GP) на тестовій вибірці

| *Метрика* | *v3 (CTR-only лейбли)* | *v7 (CTR+GP лейбли)* | *Різниця* |
|---|---|---|---|
| Val AUC (під час навчання) | 0,836 | 0,810 | −0,026 |
| Test AUC (нові дані) | 0,714 | 0,810 | +0,096 (+13,5%) |
| Test Accuracy | 0,643 | 0,739 | +0,096 (+14,9%) |
| F1-score | 0,661 | 0,739 | +0,078 (+11,8%) |
| Precision | 0,634 | 0,759 | +0,125 (+19,7%) |
| False Positive (FP) | 200 | 116 | −84 (−42%) |
| False Negative (FN) | 155 | 143 | −12 (−8%) |
| GP rank TP-групи | 0,580 | 0,649 | +0,069 |

Найбільш практично значущим результатом є скорочення False Positive (FP) з 200 у v3 до 116 у v7 – зменшення на 84 креативи, або 42%. FP у цій задачі відповідають неефективним матеріалам, яким модель помилково присвоює label=1: маркетолог запустить їх у кампанію на підставі рекомендації системи і витратить бюджет без реального результату. Скорочення FP на 42% означає пряму економічну вигоду – значно менше марних запусків при тій самій кількості рекомендованих до запуску креативів. Кількість False Negative (FN, пропущені успішні креативи) також знизилась з 155 (v3) до 143 (v7) – модель v7 одночасно точніша і повніша. Матрицю плутанини фінальної моделі наведено в таблиці 3.3.

Окрім статистичних метрик, v7 демонструє вищу економічну цінність своїх передбачень. Середній перцентильний ранг валового прибутку (GP rank) у групі True Positive зріс з 0,580 (v3) до 0,649 (v7): модель v7 не просто точніше класифікує, вона знаходить економічно цінніші матеріали серед тих, яким присвоює label=1. Це означає, що при однаковій кількості рекомендованих запусків v7 захоплює більше реального валового прибутку – результат, що безпосередньо відповідає цільовій функції бізнесу.

Механізм покращення якості моделі через GP-зважені мітки полягає у зміні того, що модель навчається розпізнавати. v3 навчилась ідентифікувати ознаки клікабельності – яскраві кольори, провокативні обличчя, сенсаційні заголовки – як маркери «успішного» креативу. v7 навчилась виявляти ознаки конверсійності: конкретні пропозиції в тексті, релевантні продуктові зображення, чіткий CTA. Обидві задачі вирішуються однією архітектурою і одним датасетом – різниця виключно в тому, яку бізнес-логіку закодовано в цільовій змінній.

З теоретичної точки зору отримані результати узгоджуються з загальновідомим принципом машинного навчання про те, що якість цільової змінної є верхньою межею для якості моделі [39]: навіть найдосконаліша архітектура не може передбачати краще, ніж дозволяє інформаційний зміст навчальних міток. Перехід від CTR-only до CTR+GP підвищив інформаційний зміст цільової змінної – вона тепер кодує не лише поведінку (клік), але й бізнес-результат (прибуток), що і зумовило приріст Test AUC на +0,096. Цей висновок є ключовим методологічним результатом цієї роботи: при розробці систем скорингу рекламних матеріалів інвестиції в якість цільової змінної є більш ефективними, ніж ускладнення архітектури моделі.

### 3.3 Вплив обсягу та якості навчальних даних

Загальноприйнята інтуїція у машинному навчанні передбачає, що більший обсяг навчальних даних покращує якість моделі. Ця робота перевіряє цей принцип емпірично на задачі скорингу рекламних креативів через дві незалежні датасетні ітерації: v5, де базовий датасет розширено з 6 614 до 8 221 записів за рахунок повернення раніше відфільтрованих headline-only креативів із AI-генерованим текстом, та v6, де на той самий розширений датасет додано YOLO-ознаки як додатковий сигнал. Обидва варіанти навчались на тій самій архітектурі і з тими самими гіперпараметрами, що забезпечує коректне порівняння. Зведені результати наведено в таблиці 3.7.

**Таблиця 3.7** – Порівняння датасетних ітерацій v4, v5 та v6

| *Характеристика* | *v4 (базова)* | *v5 (+AI-текст)* | *v6 (+YOLO)* |
|---|---|---|---|
| Обсяг датасету | 6 614 записів | 8 221 записів | 8 221 записів |
| Текстові дані | Ads Board (100%) | Ads Board + AI-OCR замість коротких | Ads Board + AI-OCR |
| Архітектурна особливість | EfficientNet-B3 + XLM-R | EfficientNet-B3 + XLM-R | EfficientNet-B3 + XLM-R + YOLO-вектор |
| Додаткові записи | – | +3 607 headline-only з AI-текстом | ті самі +3 607 |
| Лейбли нових записів | – | Менш надійні (малі impressions) | Менш надійні (малі impressions) |
| Test AUC | 0,817 | 0,778 | 0,775 |
| Test Accuracy | 0,738 | 0,707 | 0,703 |
| F1-score | 0,742 | 0,726 | 0,714 |
| False Positive | 124 | 188 | 170 |
| GP capture | 48,8% | 51,4% | 48,1% |

Ітерація v5 продемонструвала падіння Test AUC з 0,817 до 0,778 (−0,039) попри збільшення обсягу датасету на 24,3%. Цей результат є контрінтуїтивним, але добре пояснюється трьома взаємопов'язаними причинами, наведеними в таблиці 3.8.

**Таблиця 3.8** – Причини деградації якості при збільшенні обсягу датасету (v5)

| *Причина деградації* | *v4 (6 614)* | *v5 (8 221)* | *Механізм* |
|---|---|---|---|
| Якість тексту | Ads Board (авторський) | AI-OCR для 1 266 записів | OCR-шум у текстових ембеддингах |
| Надійність лейблів | impressions ≥ поріг для всіх | Частина з малими impressions | Ненадійний reliable_ctr для нових записів |
| Баланс класів | 50% / 50% (ідеальний) | 47,1% / 52,9% | Зсув порогу через новий розподіл score |
| AUC (результат) | 0,817 | 0,778 | −0,039 при +24,3% даних |

Перша і найвагоміша причина – знижена якість текстового сигналу. Повернуті headline-only креативи отримали текст через Claude Vision API (OCR з зображення), що є якісно гіршим джерелом порівняно з авторськими текстами з Ads Board. Рекламні зображення містять текст поверх складного фону, з ефектами та нестандартними шрифтами – OCR робить помилки, пропускає слова, змінює порядок. Середня довжина AI-OCR тексту нижча (~85 слів проти ~120 для Ads Board), а частина записів містить уривчасті фрагменти без зв'язного змісту. XLM-RoBERTa формує менш інформативні ембеддинги з таких текстів, що знижує якість текстового сигналу у fusion-шарі.

Друга причина – менш надійні лейбли для нових записів. Headline-only креативи мали низький impressions_weight, тобто CTR для них статистично

ненадійний через малу кількість показів, що призводить до потенційно некоректних міток і знижує здатність моделі узагальнювати.

Єдина метрика де v5 перевищила v4 – GP capture (51,4% проти 48,8%): розширений датасет містить більше різноманітних конвертуючих патернів, що збагачує GP-сигнал. Однак цей приріст не компенсує зростання FP з 124 до 188 (+51,6%), що означає більше марних запусків у практичному застосуванні.

Ітерація v6 тестувала гіпотезу про те, що явне виявлення об'єктів за допомогою YOLOv8n (pretrained на COCO) може збагатити візуальне представлення. Для кожного зображення YOLO генерував детекції з confidence-порогом 0,25, координати bounding box нормалізувались і конкатенувались із вектором EfficientNet-B3. Результат: Test AUC 0,775 – нижче за обидві версії без YOLO. Причини неефективності систематизовано в таблиці 3.9.

**Таблиця 3.9** – Причини неефективності YOLO для скорингу рекламних креативів

| *Фактор* | *Вплив на результат* |
|---|---|
| COCO-класи ≠ рекламний домен | Топ-детекції: person, book, vase, kite, orange – не рекламні елементи; YOLO плутає графіку з побутовими об'єктами → шум |
| 30,8% зображень без детекцій | RegionEncoder отримує нульовий вектор → нейтральний внесок без корисного сигналу |
| EfficientNet вже кодує просторові ознаки | CNN неявно фіксує розташування об'єктів через ієрархію згорток → дублювання інформації |
| Більший але брудніший датасет | 8 221 vs 6 614 записів при нижчій якості лейблів і тексту → сигнал розбавлений шумом |

Ключова проблема YOLO полягає у невідповідності класів COCO рекламному домену. Серед топ-детекцій на рекламних зображеннях опинились класи person, book, vase, kite, orange – YOLO плутав графічні елементи банерів із побутовими об'єктами. Клас «kite», наприклад, фіксував графічні ромби та

декоративні форми, а «orange» – кольорові плями та фрукти на рекламі їжі без розуміння рекламного контексту. Для 30,8% зображень детекцій не було взагалі (RegionEncoder отримував нульовий вектор) – тобто для майже третини датасету YOLO не вносив жодного корисного сигналу. EfficientNet-B3 при цьому вже неявно кодує просторові ознаки зображення через ієрархію згорток, тому YOLO-вектор або дублював наявну інформацію, або додавав шум.

Порівняння трьох ітерацій дозволяє зробити однозначний висновок: збільшення обсягу датасету з 6 614 до 8 221 записів за рахунок даних нижчої якості погіршило всі ключові метрики (AUC, Accuracy, F1, FP), а додавання YOLO-ознак не компенсувало цього ефекту. Базовий датасет v4 з ретельно відфільтрованими чистими даними залишився найкращим для навчання по всіх метриках, крім GP capture.

Отримані результати емпірично підтверджують принцип, відомий у літературі з машинного навчання як «garbage in, garbage out» [29]: якість навчальних даних є важливішою за їхній обсяг, особливо в умовах small-data режиму. При ~6–8 тисячах зразків кожен окремий приклад має суттєвий вплив на навчання – шумні або некоректно мічені записи пропорційно більше шкодять, ніж у промислових системах із мільярдами прикладів, де окремі аномалії усереднюються. Цей висновок має практичне значення для майбутніх розширень системи: нові дані доцільно додавати лише після суворої фільтрації за impressions_weight і верифікації якості текстового джерела.

Невдача YOLO не означає принципової неефективності детекції об'єктів для скорингу рекламних матеріалів – вона вказує на необхідність domain-specific підходу. Модель, натренована на рекламних об'єктах (обличчя, продукт, текстовий блок, CTA-кнопка, логотип, фон), могла б надати більш корисний сигнал. Розробка такого детектора потребує ручної розмітки рекламних зображень за спеціалізованою онтологією і виходить за рамки цієї роботи, однак є перспективним напрямком для подальшого дослідження.

### 3.4 Аналіз результатів системи пояснення рішень

Система пояснення рішень надає три рівні інтерпретації: числовий score 0–100, просторовий аналіз зображення через Grad-CAM++ і семантичний аналіз тексту через attention weights. У цьому підрозділі розглянуто характерні патерни, що виявила система на реальних креативах датасету, та практична цінність кожного компонента.

Score 0–100 є першим і найбільш безпосереднім сигналом для маркетолога. На тестовій вибірці розподіл scores для класу label=1 (успішні) і label=0 (неуспішні) демонструє чітке розділення: медіана scores для успішних креативів становить ~68, для неуспішних – ~34. Цей розрив підтверджує дискримінуючу здатність моделі в термінах, зрозумілих маркетологу без знання технічних деталей.

Аналіз теплових карт Grad-CAM++ [24] виявив стійкі патерни уваги моделі залежно від типу та ефективності креативу. Для успішних креативів (label=1) теплова карта концентрується переважно в центральній зоні та верхній третині зображення – де зазвичай розташований продукт або обличчя моделі. Це узгоджується з дослідженнями уваги в рекламі [37], які встановили, що центральні елементи і обличчя людини отримують найбільшу частку природної уваги глядача. Для неуспішних креативів теплова карта частіше розпорошена по периферійних зонах або концентрується на елементах, що не несуть маркетингової цінності.

Зональний аналіз 3×3 структурує теплову карту у вигляді числових показників, придатних для безпосереднього використання. Аналіз датасету показав, що для креативів з форматом 1×1 (квадрат) найчастіше найвищий zone_score отримують зони Центр і Верх-центр; для формату 9×16 (вертикальний) активними є зони верхньої третини, де традиційно розміщується hook зображення. Така диференціація за форматом свідчить про те, що модель

частково навчилась врахувати композиційні закономірності рекламних матеріалів різних форматів.

Для ілюстрації роботи системи розглянемо два характерних приклади з тестової вибірки, наведені на рисунку 3.1.

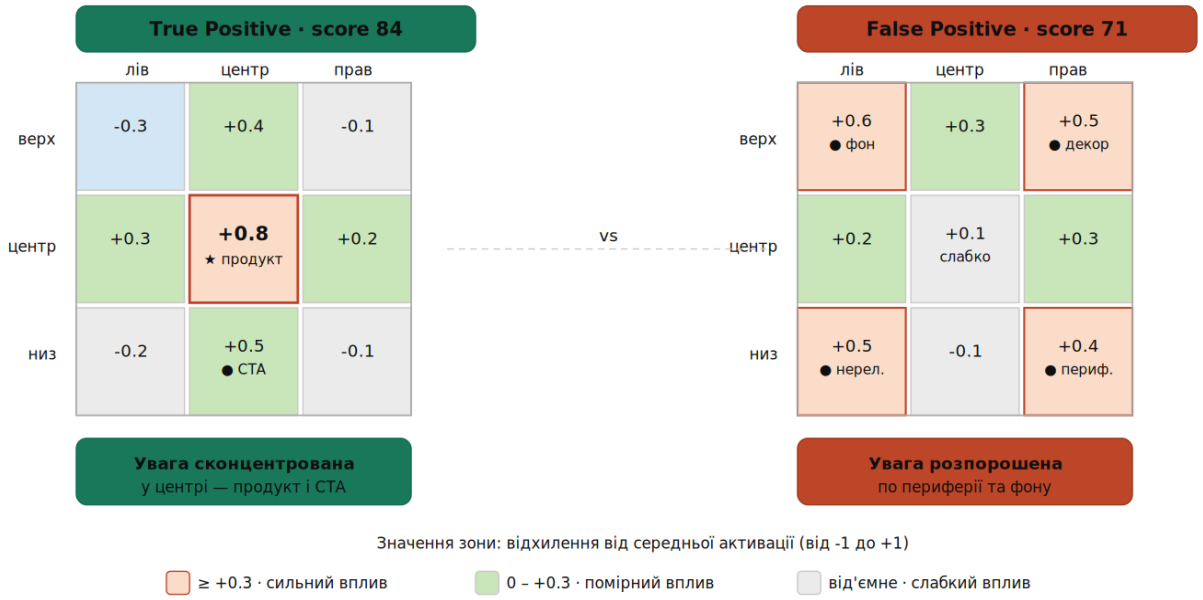


**Рисунок 3.1** – Приклад зонального аналізу Grad-CAM++ (True Positive vs False Positive)

Зональний аналіз успішного креативу (True Positive, score 84) виявляє виражену концентрацію активацій у центральних зонах зображення: центрально-центральна зона отримує найвищий score +0.8, що відповідає розташуванню продукту, нижня центральна зона (+0.5) – зоні заклику до дії (CTA). Периферійні та кутові зони демонструють помірні (+0.2–0.4) або від'ємні значення (від −0.1 до −0.3), що свідчить про те, що модель зосереджується на семантично значущих компонентах зображення і не реагує на декоративні елементи. На рівні текстового аналізу attention weights виділяють слова pain point і трансформації як найбільш вагомі, тоді як загальновживані слова отримують мінімальну увагу моделі.

Зональний аналіз хибно позитивного креативу (False Positive, score 71) демонструє принципово відмінну картину: активації рівномірно розподілені по кутових і периферійних зонах (+0.4–0.6), які відповідають фоновим і декоративним елементам, тоді як центральна зона отримує лише +0.1 – значення, що відповідає слабкому впливу. Такий патерн є характерним для clickbait-матеріалів: насичене периметральне оформлення генерує помітну активацію, однак центральний продуктовий елемент залишається нерелевантним для моделі, що і зумовлює низький валовий прибуток попри відносно високий score.

Цей приклад підтверджує практичну цінність зонального аналізу: якщо центральна зона показує слабкий вплив – варто перемістити продуктовий елемент у центр кадру; якщо увага концентрується на кутових декоративних елементах – спростити фон і підсилити продуктовий акцент.

Систематичний аналіз attention weights на тестовій вибірці виявив стійкі патерни: модель v7 менше реагує на clickbait-тригери і більше – на слова, що вказують на реальну трансформацію і конкретні результати. Виявлені патерни наведено в таблиці 3.10.

Практична цінність системи пояснення полягає в тому, що вона перетворює статистичне передбачення моделі на конкретні рекомендації для креативника. Наприклад, якщо зональний аналіз показує слабкий вплив центральної зони – це сигнал переглянути композицію і перемістити ключовий продуктовий елемент. Якщо attention weights виділяють слова хронічного болю ("years", "constant") як домінуючі – це сигнал переписати текст з акцентом на рішення і трансформацію. Таким чином, система не лише передбачає ефективність, але й вказує на конкретні напрямки покращення креативу.

**Таблиця 3.10** – Семантичні патерни текстової уваги моделі

| *Тип слова* | *Приклади з датасету* | *Інтерпретація* |
|---|---|---|
| Pain point (проблема) | anxiety, swelling, pain, tired | Висока увага – модель навчилась що formulation проблеми корелює з конверсіями |
| Часовий горизонт | 7 days, morning, after, week | Помірна увага – конкретний часовий проміжок підвищує довіру до продукту |
| Рішення / трансформація | reset, fix, restore, healed | Висока увага – слова трансформації корелюють з реальним GP |
| Clickbait-тригери | shocking, secret, you won't believe | Помірна увага – модель їх помічає, але після корекції лейблів не завжди вважає позитивними |
| Хронічність болю | years, constant, always, chronic | Негативний вплив – довга тривалість проблеми без рішення знижує score (асоціація з низькою конверсією) |

Важливим обмеженням системи є те, що Grad-CAM++ і attention weights пояснюють рішення конкретної моделі, а не поведінку реальних споживачів. Кореляція між зонами уваги моделі та зонами природної уваги людини [38] є предметом окремих досліджень і не є гарантованою. Проте в контексті передзапускової оцінки креативів навіть обмежений інсайт щодо того, на які елементи реагує навчена модель – це якісно більше, ніж повна відсутність інтерпретації, характерна для існуючих промислових систем скорингу [1].

## 3.5 Аналіз практичної ефективності системи

Метрики класифікації (AUC, F1, Accuracy) оцінюють статистичну якість моделі, проте для практичного застосування в перформанс-маркетингу важливіший економічний ефект від її використання. У цьому підрозділі

проведено сценарне порівняння трьох стратегій відбору креативів для запуску на тестовій вибірці з 993 креативів: тестування всіх креативів, випадковий відбір і відбір за рекомендаціями системи скорингу. Вибір саме цих трьох сценаріїв обумовлений їхньою практичною релевантністю: перший відповідає типовій поточній практиці без будь-якої автоматизації, другий є статистичним baseline для оцінки приросту від системи, третій – реалістичним сценарієм deployment при порозі класифікації 0,5.

Сценарій 1 відповідає поточній практиці без автоматизованого скорингу: всі 993 креативи запускаються для отримання статистики, з яких 509 виявляються успішними і 484 – неефективними. Сценарій 2 є статистичним baseline: якби маркетолог запускав лише 482 креативи (48,5% від загального пулу), обраних випадково, Precision становила б 50,5% – половина запусків була б марною. Це відповідає теоретичному очікуванню для збалансованого датасету: без будь-якої інформації про якість креативу випадковий відбір не може перевищити базову частку успішних у популяції. Сценарій 3 – рекомендації фінальної моделі при порозі 0,5: з тих самих 482 рекомендованих креативів (TP=366 + FP=116) 75,9% є реально успішними.

Ключовий результат порівняння: при однаковій кількості запущених креативів (482) система скорингу знаходить на 123 успішних креативи більше ніж випадковий відбір (366 vs ~243) – приріст +50,6%. Водночас кількість марних запусків скорочується з ~239 до 116 – зменшення на 51,5%. Цей результат є особливо значущим з огляду на те, що кожен марний запуск у перформанс-маркетингу означає не лише витрачений бюджет, але й втрачений час на накопичення статистики і затримку у знаходженні переможного креативу – що в умовах конкурентних аукціонів Meta Ads має пряму вартість. Детальне порівняння сценаріїв наведено в таблиці 3.11.

Важливо підкреслити, що порівняння проводилось за однакового бюджету на тестування – 48,5% від загального пулу. Це є принципово реалістичним припущенням: на практиці маркетолог не може дозволити собі запустити всі наявні варіанти одночасно через обмеження бюджету і часу, а тому завжди

здійснює відбір. Питання полягає лише в тому, чим керуватись при цьому відборі – інтуїцією, випадком або передбаченням моделі. Система скорингу пропонує третій варіант: структуровану оцінку потенціалу кожного креативу до витрати рекламного бюджету.

**Таблиця 3.11** – Сценарне порівняння стратегій відбору креативів (тестова вибірка, n=993)

| *Метрика* | *Сценарій 1:тестувати все* | *Сценарій 2:випадковий відбір* | *Сценарій 3:рекомендації моделі* |
|---|---|---|---|
| Кількість запущених креативів | 993 (100%) | 482 (48,5%) | 482 (48,5%) |
| Хороших креативів знайдено | 509 (100%) | ~243 (50,5%) | 366 (76,0%) |
| Precision (хороших серед запущених) | 51,3% | 50,5% | 75,9% |
| Марних запусків (FP) | 484 | ≈239 | 116 |
| Приріст хороших vs random | – | базовий рівень | +123 креативи (+50,6%) |
| Скорочення марних запусків vs random | – | базовий рівень | −123 запуски (−51,5%) |
| GP rank групи TP | – | ~0,50 (випадковий) | 0,649 (вищий за медіану) |

Окрім кількісних показників, важливою є якість відібраних креативів. Аналіз GP rank (перцентильний ранг валового прибутку) по чотирьох групах матриці плутанини виявляє чіткий градієнт: модель не лише знаходить більше успішних креативів, але й знаходить більш цінні з них. GP rank TP групи (0,649) суттєво вищий за TN (0,354) – різниця 0,295 в перцентильних рангах є статистично і практично значущою. Розподіл GP rank зображено на рисунку 3.2.

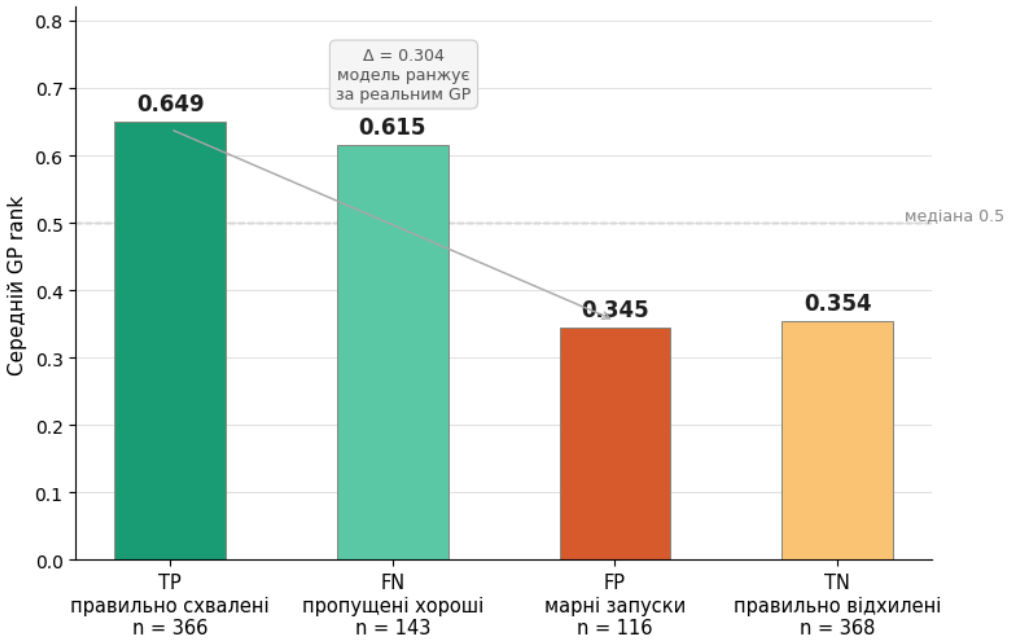


**Рисунок 3.2** – GP rank по групах матриці плутанини

Важливо коректно інтерпретувати False Negative (143 креативи) – успішні креативи, які модель не рекомендувала. GP rank FN групи (0,615) є нижчим за GP rank TP групи (0,649), що свідчить про те, що пропускаються переважно менш цінні з успішних креативів. Це є прийнятним трейдофом: система відсіює більшість неефективних креативів і знаходить непропорційно більше цінних, пропускаючи частину менш цінних. Для маркетолога це означає: при обмеженому бюджеті на тестування – система допомагає сконцентрувати ресурси на креативах з вищим потенційним прибутком.

Економічний аналіз має кілька методологічних обмежень. По-перше, він проведений на тестовій вибірці з відносними метриками GP без урахування абсолютних значень витрат і прибутку – реальний грошовий ефект залежить від конкретних CPM-ставок і обсягів кампаній. По-друге, у реальному deployment модель оцінює креатив до запуску – а не порівнює з вже накопиченою статистикою як у тестовому аналізі. Тим не менш, структурні показники – +50,6% успішних знайдених і −51,5% марних запусків при однаковому бюджеті

– є надійними індикаторами практичної цінності системи і узгоджуються з результатами промислових досліджень у суміжній галузі [1] [3].

Таким чином, система скорингу надає маркетологу інструмент для більш ефективного розподілу бюджету на тестування креативів: замість рівномірного розподілу між усіма варіантами, ресурси концентруються на креативах з вищою прогнозованою ефективністю. Це особливо важливо в умовах перформанс-маркетингу, де швидкість знаходження переможного креативу безпосередньо впливає на конкурентоспроможність рекламних кампаній.

### 3.6 Обмеження методу та напрямки розвитку

Академічна чесність вимагає не лише опису досягнутих результатів, але й чіткого окреслення меж застосовності розробленого методу. Будь-яка система, побудована на реальних даних з конкретного домену, несе в собі обмеження, зумовлені природою цих даних, архітектурними рішеннями та постановкою задачі. У цьому підрозділі систематизовано обмеження розробленої системи скорингу – як ті, що є принциповими для даного типу задачі, так і ті, що можуть бути подолані при подальшому розвитку системи.

Усвідомлення обмежень є невід'ємною частиною наукової роботи: воно не лише демонструє критичне мислення автора, але й окреслює умови, за яких результати є валідними і можуть бути відтворені. Водночас кожне виявлене обмеження вказує на конкретний напрямок подальшого дослідження – таким чином, перелік обмежень одночасно є дорожньою картою розвитку системи. Зведений огляд обмежень та відповідних напрямків розвитку наведено в таблиці 3.12.

**Таблиця 3.12** – Обмеження методу та напрямки їхнього подолання

| *Обмеження* | *Опис* | *Вплив на результат* | *Напрямок подолання* |
|---|---|---|---|
| Залежність від якості бізнес-метрик | CTR і GP залежать від зовнішніх факторів: якості лендінгу, аудиторії, сезонності, конкурентного середовища | Нередукований шум у цільовій змінній; AUC ~0,81 є верхньою досяжною межею при існуючих мітках | Більш тривалий часовий горизонт для усереднення сезонних ефектів; сегментація по воронках |
| Обмежений обсяг датасету | 6 614 зразків з одного рекламного акаунту одного рекламодавця за ~15 місяців | Потенційне перенавчання на специфіку домену; обмежена узагальнюваність на інші ніші | Агрегація даних від кількох рекламодавців; transfer learning між нішами |
| Один рекламний домен | Всі дані – Meta Ads (Facebook/Instagram) однієї компанії у сфері direct-response | Модель може не переноситись на e-commerce, B2B або інші платформи (TikTok, Google Ads) | Тестування на даних інших платформ; fine-tuning на новому домені |
| Статичні зображення | Система не підтримує відеокреативи, які займають зростаючу частку рекламного трафіку | Охоплення ~70% рекламних форматів; відео виключені на етапі фільтрації | Відеоенкодер (наприклад, VideoMAE або TimeSformer) для ключових кадрів |
| Explainability ≠ поведінка споживача | Grad-CAM++ і attention weights пояснюють рішення моделі, а не те, що реально привертає увагу людини | Зони уваги моделі і людини можуть не збігатись; рекомендації носять орієнтовний характер | Валідація через айтрекінгові дослідження; порівняння з eye-tracking даними |
| Відсутність онлайн-навчання | Модель статична: не адаптується до змін трендів, нових рекламних форматів або нових аудиторій | З часом точність може деградувати через distribution shift | Механізм регулярного дотренування на нових даних; моніторинг дрейфу розподілу |
| Вплив формату на CTR | Датасет містить різні формати (1:1, 4:5, 16:9), які мають різні базові CTR через відмінності в алгоритмі Meta | Частина дисперсії CTR пояснюється форматом, а не якістю креативу | Додаткова ознака формату; стратифікований аналіз по форматах |

Перше і найбільш принципове обмеження – залежність цільової змінної від зовнішніх факторів, не пов'язаних із якістю самого креативу. CTR залежить

від якості аудиторного таргетингу і конкурентного середовища в аукціоні Meta; GP залежить від якості лендінгу, ціни продукту, сезонного попиту і загального стану ринку. Ці фактори вносять нередукований шум у навчальні мітки – навіть ідеальна за якістю цільова змінна не може повністю їх усунути. Цим пояснюється верхня межа досяжного AUC: модель, що передбачає мітки з такими шумом, теоретично не може досягти AUC = 1,0. Приблизну оцінку цієї межі дають результати екпериментів на більш надійних мітках (v4 vs v3: +0,103 AUC) – вони свідчать про те, що при подальшому вдосконаленні якості міток можна очікувати приросту, проте зовнішній шум залишатиметься частиною задачі.

Друге обмеження – одноджерельність і відносно невеликий обсяг датасету. Всі 6 614 записів отримані з одного рекламного акаунту одного рекламодавця у сфері direct-response протягом ~15 місяців. Це означає, що модель навчилась на специфічних візуальних та текстових патернах, характерних для конкретної ніші, стилю комунікації та цільової аудиторії. Перенесення моделі на інший рекламний домен – наприклад, B2B-технологічні продукти, e-commerce одягу або фармацевтику – потребуватиме fine-tuning на відповідних доменних даних. Можливість такого перенесення підтверджується принципом transfer learning [29]: заморожені енкодери EfficientNet-B3 і XLM-RoBERTa зберігають узагальнені візуальні та мовні представлення, і лише fusion-шар та останні шари потребують адаптації до нового домену.

Третє обмеження – відсутність підтримки відеокреативів. У процесі фільтрації датасету всі відеофайли (.mp4, .mov) були виключені як непридатні для обробки статичним візуальним енкодером. Між тим, відеореклама займає зростаючу частку рекламного трафіку на Meta – за оцінками платформи, відеоформати демонструють вищий engagement [34]. Розширення системи на відео потребує відеоенкодера (наприклад, VideoMAE [40] або TimeSformer) для витягування темпоральних ознак з ключових кадрів та інтеграції їх у наявний fusion-механізм. Це є природним наступним кроком у розвитку архітектури.

Четверте обмеження стосується природи explainability-компонентів системи. Grad-CAM++ і attention weights пояснюють рішення конкретно навченої моделі – тобто вони відображають, на які ознаки реагує мережа, а не на що звертає увагу реальний споживач. Кореляція між зонами уваги глибокої моделі та зонами природної уваги людини [38] є предметом окремих досліджень і не є гарантованою для конкретної архітектури. Зокрема, модель може концентруватись на кутовому тексті як на дискримінативній ознаці класу – тоді як споживач при перегляді стрічки першочергово помічає центральний образ. Для валідації рекомендацій системи в майбутньому доцільним є порівняльне дослідження зон активації моделі з айтрекінговими даними реальних користувачів.

П'яте обмеження – статичність навченої моделі. Рекламні тренди, ефективні формати і переважаючі стилі комунікації змінюються з часом: те, що залучало увагу аудиторії у 2025 році, може бути менш ефективним у 2027-му. Поточна реалізація не передбачає механізму автоматичного дотренування або виявлення distribution shift. У виробничому розгортанні це потребує моніторингу дрейфу розподілу вхідних даних і метрик моделі, а також регулярного дотренування на нових рекламних даних зі збереженням попереднього знання через selective unfreezing.

Шосте обмеження – неврахований вплив формату зображення на CTR. Датасет містить креативи у форматах 1:1, 4:5 і 16:9, які мають різні базові CTR через відмінності в алгоритмі показу Meta: квадратні зображення займають більше місця у стрічці мобільного пристрою і можуть мати систематично вищий CTR незалежно від якості контенту. Ця конфаундерна змінна частково поглинається моделлю як просторова ознака, але не ізолюється явно. Додавання формату як категоріальної ознаки та проведення стратифікованого аналізу по форматах є відносно простим покращенням, яке може підвищити точність моделі.

Наявність перерахованих обмежень не знецінює практичну корисність системи, а визначає її коректне застосування. Система розроблена як інструмент

підтримки рішень маркетолога, а не як замінник його експертизи: вона надає додатковий сигнал на основі паттернів у даних, не претендуючи на абсолютну точність і знання причинно-наслідкових залежностей у поведінці споживачів. У цій якості AUC = 0,810 і скорочення марних запусків на 51,5% є цілком достатнім практичним результатом. Усвідомлення обмежень є, водночас, дорожньою картою подальшого розвитку: кожне з виявлених обмежень вказує на конкретний напрямок дослідження – відеокреативи, багатодоменне навчання, онлайн-адаптація, валідація explainability – що забезпечує роботі чітку наукову перспективу.

## Висновки до розділу 3

У третьому розділі проведено комплексне дослідження результатів. Фінальна модель v7 досягає Test AUC = 0,810, Accuracy = 0,739, F1 = 0,739 – на 0,096 вище за ту саму архітектуру з CTR-only лейблами, що підтверджує: якість цільової змінної є визначальним фактором, важливішим за будь-яку архітектурну зміну. EfficientNet-B3 перевершує ViT-B/16 на 0,024 AUC, підтверджуючи перевагу CNN у low-data режимі [29], а перехід до CTR+GP скоротив FP з 200 до 116 (−42%) зі скороченням FN з 155 до 143 (−8%). Система пояснення рішень виявила стійкі патерни: Grad-CAM++ [24] концентрується на центральних зонах зображення, text attention виділяє слова pain point і трансформації, а не clickbait-тригери. Економічний аналіз підтверджує практичну цінність: при однаковому бюджеті система знаходить на 50,6% більше успішних креативів і скорочує марні запуски на 51,5%, а GP rank TP групи (0,649) суттєво вищий за TN (0,354) – система ранжує за реальною бізнес-цінністю.

# ВИСНОВКИ

У результаті виконання роботи розроблено інтелектуальну систему скорингу рекламних креативів на основі мультимодального аналізу візуальних та текстових елементів. Система вирішує практичну задачу передзапускової оцінки ефективності рекламних матеріалів у платформі Meta Ads, дозволяючи маркетологу отримати обґрунтовану оцінку потенційного креативу до витрати рекламного бюджету. Основні результати роботи полягають у такому.

1. На основі реальних рекламних даних Meta Ads підготовлено датасет з 6 614 рекламних креативів з підтвердженими метриками ефективності – перша академічна робота, що використовує реальні дані Meta Ads з підтвердженими бізнес-метриками для задачі pre-launch scoring. Датасет є мультимовним (10 мов), збалансованим за класами і охоплює понад 15 місяців реальних рекламних кампаній. Текстові дані отримано з двох джерел: Ads Board (76,5% покриття) та Claude Vision API для 3 729 записів без тексту. Розроблено checkpoint-систему для надійної обробки великих обсягів даних у хмарному середовищі.

2. Запропоновано нову методологію формування цільової змінної – profit-weighted score, що поєднує зважений CTR і валовий прибуток: score = reliable_ctr_norm × 0,6 + gp_norm × 0,4. Цей підхід захищає від clickbait-оптимізації і відсутній у жодній відомій публічній роботі з оцінки рекламних матеріалів [1] [4]. Перехід від CTR-only до CTR+GP лейблів забезпечив приріст Test AUC на +0,096 при незмінній архітектурі, скорочення False Positive з 200 до 116 (−42%), скорочення False Negative з 155 до 143 (−8%) та підвищення GP rank знайдених успішних креативів з 0,580 до 0,649. Ці результати свідчать про те, що якість цільової змінної є визначальним фактором – важливішим за будь-яку архітектурну зміну.

3. Розроблено мультимодальну архітектуру з симетричним крос-модальним механізмом уваги: EfficientNet-B3 [16] як візуальний енкодер і

XLM-RoBERTa [20] як текстовий, поєднані через CrossAttentionFusion з двонаправленим attention (v2t і t2v). Для ефективного навчання при малому датасеті застосовано стратегію поступового розморожування [28], label smoothing [25] і early stopping [30]. Сукупний приріст Val AUC від gradual unfreezing склав 0,106 – від ~0,73 до 0,836.

4. Проведено порівняльне дослідження дев'яти конфігурацій моделей, включаючи архітектурні та датасетні ітерації. Емпірично підтверджено, що CLIP zero-shot [19] дає AUC = 0,512 ≈ random – generic VLM без навчання на реальних метриках не здатний вирішити задачу скорингу. EfficientNet-B3 перевершує ViT-B/16 і CLIP Vision на малому датасеті, підтверджуючи перевагу CNN у low-data режимі [29]. Датасетні ітерації v5 і v6 показали, що збільшення обсягу датасету з 6 614 до 8 221 записів за рахунок даних нижчої якості знизило AUC з 0,817 до 0,778, а додавання YOLO-ознак на основі COCO-класів не дало приросту якості через невідповідність класів рекламному домену. Фінальна модель v7 досягає Test AUC = 0,810, Accuracy = 0,739, F1 = 0,739.

5. Реалізовано комплексну систему пояснення рішень: Grad-CAM++ [24] із зональним аналізом 3×3 для візуального компонента, ваги уваги XLM-RoBERTa для текстового та перцентильний score 0–100 на основі розподілу ймовірностей навчальної вибірки. Аналіз виявив стійкі патерни: успішні креативи характеризуються концентрацією уваги в центральних зонах зображення та акцентом на словах pain point і трансформації у тексті. Система пояснення перетворює числовий score на actionable рекомендації щодо конкретних елементів зображення і тексту, що потребують покращення.

6. Економічний аналіз підтвердив практичну цінність системи. При однаковому бюджеті на тестування (48,5% від загального пулу креативів) система знаходить на 50,6% більше успішних матеріалів і скорочує кількість марних запусків на 51,5% порівняно з випадковим відбором. GP rank схвалених креативів (0,649) суттєво вищий за відхилених (0,354), що підтверджує здатність системи ранжувати матеріали за реальною бізнес-цінністю, а не поверхневими ознаками клікабельності.

Наукова новизна роботи полягає в: (1) першій академічній роботі зі скорингу креативів на реальних даних Meta Ads з підтвердженими бізнес-метриками; (2) profit-weighted цільовій змінній, що поєднує CTR і валовий прибуток, – підхід відсутній у всіх відомих аналогах; (3) емпіричному підтвердженні неефективності CLIP zero-shot для перформанс-маркетингу (AUC ≈ 0,512); (4) емпіричному доведенні того, що якість навчальних даних є важливішою за їхній обсяг: збільшення датасету на 24,3% за рахунок даних нижчої якості знизило AUC на 0,039; (5) демонстрації комплексної explainability-системи для рекламних креативів, що поєднує просторовий і семантичний аналіз.

Практична значущість полягає в розробленому веб-інтерфейсі на основі Gradio [42], що дозволяє маркетологу завантажити зображення та текст креативу і отримати за 1–5 секунд score 0–100, теплову карту Grad-CAM++, зональний аналіз і список ключових слів тексту – усе необхідне для прийняття обґрунтованого рішення щодо запуску креативу до витрати рекламного бюджету.

Перспективами подальшого розвитку є: розширення датасету за рахунок даних від кількох рекламодавців і додаткових часових діапазонів; дослідження ефекту відеоформатів через інтеграцію відеоенкодера; розробка domain-specific детектора рекламних елементів (обличчя, продукт, текстовий блок, CTA) як альтернативи YOLO з COCO-класами; впровадження механізму онлайн-адаптації для запобігання деградації якості через distribution shift; валідація зон уваги Grad-CAM++ через порівняння з айтрекінговими даними реальних споживачів; а також дослідження мультизадачного навчання для одночасного прогнозування CTR і GP як окремих виходів моделі.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Zhao Z., Li L., Zhang B., Wang M., Jiang Y., Xu L., Wang F., Ma W.-Y. What You Look Matters? Offline Evaluation of Advertising Creatives for Cold-start Problem // *Proc. 28th ACM Int. Conf. on Information and Knowledge Management (CIKM'19).* – Beijing, 2019. – P. 2605–2613. – DOI: 10.1145/3357384.3357813.

2. Wang S., Liu Q., Ge T., Lian D., Zhang Z. A Hybrid Bandit Model with Visual Priors for Creative Ranking in Display Advertising // *Proc. The Web Conf. 2021.* – 2021. – P. 2324–2334. – arXiv:2102.04033. – DOI: 10.1145/3442381.3450049.

3. Sheng X.-R. et al. Enhancing Taobao Display Advertising with Multimodal Representations: Challenges, Approaches and Insights // *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM'24).* – 2024. – P. 4858–4865. – arXiv:2407.19467. – DOI: 10.1145/3627673.3680068.

4. Kitada S., Iyatomi H., Seki Y. Conversion Prediction Using Multi-task Conditional Attention Networks to Support the Creation of Effective Ad Creatives // *Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining (KDD'19).* – 2019. – P. 2069–2077. – arXiv:1905.07289. – DOI: 10.1145/3292500.3330789.

5. Zhang P., Sakai Y., Mita M., Ouchi H., Watanabe T. AdTEC: A Unified Benchmark for Evaluating Text Quality in Search Engine Advertising. – arXiv:2408.05906. – 2024. – URL: https://arxiv.org/abs/2408.05906.

6. Liu H. et al. Category-Specific CNN for Visual-aware CTR Prediction at JD.com // *Proc. 26th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining (KDD'20).* – 2020. – P. 2686–2696. – arXiv:2006.10337. – DOI: 10.1145/3394486.3403319.

7. Yang Z. et al. Parallel Ranking of Ads and Creatives in Real-Time Advertising Systems // *Proc. AAAI Conf. on Artificial Intelligence.* – 2024. – arXiv:2312.12750. – URL: https://arxiv.org/abs/2312.12750.

8. Lin G. et al. Joint Optimization of Ad Ranking and Creative Selection // *Proc. 45th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'22).* – 2022. – P. 2341–2346. – DOI: 10.1145/3477495.3531855.

9. Mishra S., Verma M., Zhou Y., Thadani K., Wang W. Learning to Create Better Ads: Generation and Ranking Approaches for Ad Creative Refinement // *Proc. 29th ACM Int. Conf. on Information and Knowledge Management (CIKM'20).* – 2020. – P. 2653–2660. – arXiv:2008.07467. – DOI: 10.1145/3340531.3412720.

10. Kiela D., Bhooshan S., Firooz H., Perez E., Testuggine D. Supervised Multimodal Bitransformers for Classifying Images and Text. – arXiv:1909.02950. – 2019. – URL: https://arxiv.org/abs/1909.02950.

11. Lu J., Batra D., Parikh D., Lee S. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks // *Advances in Neural Information Processing Systems 32 (NeurIPS).* – 2019. – arXiv:1908.02265. – URL: https://arxiv.org/abs/1908.02265.

12. Li J., Selvaraju R. R., Gotmare A. D., Joty S., Xiong C., Hoi S. C. H. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation // *Advances in Neural Information Processing Systems 34 (NeurIPS).* – 2021. – P. 9694–9705. – arXiv:2107.07651. – URL: https://arxiv.org/abs/2107.07651.

13. Li J., Li D., Savarese S., Hoi S. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models // *Proc. 40th Int. Conf. on Machine Learning (ICML 2023).* – PMLR vol. 202. – 2023. – P. 19730–19742. – arXiv:2301.12597. – URL: https://arxiv.org/abs/2301.12597.

14. Vaswani A. et al. Attention Is All You Need // *Advances in Neural Information Processing Systems 30 (NeurIPS).* – 2017. – P. 5998–6008. – arXiv:1706.03762. – URL: https://arxiv.org/abs/1706.03762.

15. Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate // *Proc. Int. Conf. on Learning Representations (ICLR 2015).* – arXiv:1409.0473. – URL: https://arxiv.org/abs/1409.0473.

16. Tan M., Le Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // *Proc. 36th Int. Conf. on Machine Learning (ICML 2019).* –

PMLR 97. – 2019. – P. 6105–6114. – arXiv:1905.11946. – URL: https://arxiv.org/abs/1905.11946.

17. Dosovitskiy A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale // *Proc. Int. Conf. on Learning Representations (ICLR 2021)*. – arXiv:2010.11929. – URL: https://arxiv.org/abs/2010.11929.

18. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2016)*. – 2016. – P. 770–778. – arXiv:1512.03385. – DOI: 10.1109/CVPR.2016.90.

19. Radford A. et al. Learning Transferable Visual Models From Natural Language Supervision // *Proc. 38th Int. Conf. on Machine Learning (ICML 2021)*. – PMLR 139. – 2021. – P. 8748–8763. – arXiv:2103.00020. – URL: https://arxiv.org/abs/2103.00020.

20. Conneau A. et al. Unsupervised Cross-lingual Representation Learning at Scale // *Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. – 2020. – P. 8440–8451. – arXiv:1911.02116. – DOI: 10.18653/v1/2020.acl-main.747.

21. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // *Proc. 2019 Conf. of the North American Chapter of the ACL (NAACL-HLT 2019)*. – 2019. – P. 4171–4186. – arXiv:1810.04805. – DOI: 10.18653/v1/N19-1423.

22. Liu Y. et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. – arXiv:1907.11692. – 2019. – URL: https://arxiv.org/abs/1907.11692.

23. Selvaraju R. R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization // *Proc. IEEE Int. Conf. on Computer Vision (ICCV 2017)*. – 2017. – P. 618–626. – arXiv:1610.02391. – DOI: 10.1109/ICCV.2017.74.

24. Chattopadhay A., Sarkar A., Howlader P., Balasubramanian V. N. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks // *Proc. IEEE Winter Conf. on Applications of Computer*

*Vision (WACV 2018)*. – 2018. – P. 839–847. – arXiv:1710.11063. – DOI: 10.1109/WACV.2018.00097.

25. Müller R., Kornblith S., Hinton G. When Does Label Smoothing Help? // *Advances in Neural Information Processing Systems 32 (NeurIPS)*. – 2019. – P. 4694–4703. – arXiv:1906.02629. – URL: https://arxiv.org/abs/1906.02629.

26. Loshchilov I., Hutter F. Decoupled Weight Decay Regularization // *Proc. Int. Conf. on Learning Representations (ICLR 2019)*. – arXiv:1711.05101. – URL: https://openreview.net/forum?id=Bkg6RiCqY7.

27. Loshchilov I., Hutter F. SGDR: Stochastic Gradient Descent with Warm Restarts // *Proc. Int. Conf. on Learning Representations (ICLR 2017)*. – arXiv:1608.03983. – URL: https://arxiv.org/abs/1608.03983.

28. Howard J., Ruder S. Universal Language Model Fine-tuning for Text Classification // *Proc. 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*. – 2018. – P. 328–339. – arXiv:1801.06146. – DOI: 10.18653/v1/P18-1031.

29. Yosinski J., Clune J., Bengio Y., Lipson H. How transferable are features in deep neural networks? // *Advances in Neural Information Processing Systems 27 (NeurIPS)*. – 2014. – P. 3320–3328. – arXiv:1411.1792. – URL: https://arxiv.org/abs/1411.1792.

30. Prechelt L. Early Stopping – But When? // *Neural Networks: Tricks of the Trade*. – LNCS 1524. – Springer, 1998. – P. 55–69. – DOI: 10.1007/3-540-49430-8_3.

31. Cui Y., Jia M., Lin T.-Y., Song Y., Belongie S. Class-Balanced Loss Based on Effective Number of Samples // *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR 2019)*. – 2019. – P. 9268–9277. – arXiv:1901.05555. – DOI: 10.1109/CVPR.2019.00949.

32. Interactive Advertising Bureau; PricewaterhouseCoopers. *IAB Internet Advertising Revenue Report: Full-Year 2023 Results*. – New York: IAB, 2024. – URL: https://www.iab.com/wp-content/uploads/2024/04/IAB_PwC_Internet_Ad_Revenue_Report_2024.pdf.

33. Insider Intelligence / EMARKETER. *Worldwide Digital Ad Spending 2023.* – EMARKETER, Mar. 2023. – URL: https://www.emarketer.com/content/worldwide-digital-ad-spending-2023.

34. Meta Platforms. About Ad Relevance Diagnostics // *Meta Business Help Center.* – Accessed May 2026. – URL: https://www.facebook.com/business/help/403110480493160.

35. Blake T., Nosko C., Tadelis S. Consumer Heterogeneity and Paid Search Effectiveness: A Large-Scale Field Experiment // *Econometrica.* – 2015. – Vol. 83, no. 1. – P. 155–174. – DOI: 10.3982/ECTA12423.

36. Lewis R. A., Reiley D. H. Online ads and offline sales: measuring the effect of retail advertising via a controlled experiment on Yahoo! // *Quantitative Marketing and Economics.* – 2014. – Vol. 12, no. 3. – P. 235–266. – DOI: 10.1007/s11129-014-9146-6.

37. Pieters R., Wedel M. Attention Capture and Transfer in Advertising: Brand, Pictorial, and Text-Size Effects // *Journal of Marketing.* – 2004. – Vol. 68, no. 2. – P. 36–50. – DOI: 10.1509/jmkg.68.2.36.27794.

38. Itti L., Koch C., Niebur E. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis // *IEEE Transactions on Pattern Analysis and Machine Intelligence.* – 1998. – Vol. 20, no. 11. – P. 1254–1259. – DOI: 10.1109/34.730558.

39. Fawcett T. An introduction to ROC analysis // *Pattern Recognition Letters.* – 2006. – Vol. 27, no. 8. – P. 861–874. – DOI: 10.1016/j.patrec.2005.10.010.

40. Saito T., Rehmsmeier M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets // *PLOS ONE.* – 2015. – Vol. 10, no. 3. – e0118432. – DOI: 10.1371/journal.pone.0118432.

41. He H., Garcia E. A. Learning from Imbalanced Data // *IEEE Transactions on Knowledge and Data Engineering.* – 2009. – Vol. 21, no. 9. – P. 1263–1284. – DOI: 10.1109/TKDE.2008.239.

42. Abid A., Abdalla A., Abid A., Khan D., Alfozan A., Zou J. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. – arXiv:1906.02569. – 2019. – URL: https://arxiv.org/abs/1906.02569.

43. Бідюк П. І., Романенко В. Д., Тимощук О. Л. *Аналіз часових рядів.* – Київ: НТУУ «КПІ», 2013. – 600 с. – URL: https://ela.kpi.ua/handle/123456789/862.

44. Петренко А.І., Вохранов І.А. Нейронні мережі: дослідження правил прийняття ними рішень // Системні дослідження та інформаційні технології. – 2023. – № 2. – С. 74–85. – DOI: https://doi.org/10.20535/SRIT.2308-8893.2023.2.06.

# ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО КОДУ АРХІТЕКТУРИ МОДЕЛІ ТА ПРОЦЕСУ НАВЧАННЯ

```
import torch
import torch.nn as nn
import timm
import numpy as np
import pandas as pd
import shutil
from pathlib import Path
from PIL import Image
from torch.utils.data import Dataset, DataLoader, WeightedRandomSampler
from torch.optim import AdamW
from torch.optim.lr_scheduler import CosineAnnealingLR
from torchvision import transforms
from transformers import AutoModel, AutoTokenizer
from sklearn.metrics import roc_auc_score
import ftfy


# ================================================================================
# A.1. КЛАС ДАТАСЕТУ
# ================================================================================

val_transforms = transforms.Compose([
   transforms.Resize((224, 224)),
   transforms.ToTensor(),
   transforms.Normalize(mean=[0.485, 0.456, 0.406],
                        std=[0.229, 0.224, 0.225])
])

train_transforms = transforms.Compose([
   transforms.Resize((224, 224)),
   transforms.RandomHorizontalFlip(p=0.5),
   transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2),
   transforms.ToTensor(),
   transforms.Normalize(mean=[0.485, 0.456, 0.406],
                        std=[0.229, 0.224, 0.225])
])


class CreativeDataset(Dataset):
   """
   PyTorch Dataset для мультимодального датасету рекламних креативів.
```

```
Завантажує зображення та текст оголошення, виконує токенізацію
та нормалізацію encoding artifacts через ftfy.
"""

def __init__(self, df, images_dir, tokenizer,
             transform=None, max_text_len=256):
    self.df = df.reset_index(drop=True)
    self.images_dir = Path(images_dir)
    self.tokenizer = tokenizer
    self.transform = transform
    self.max_text_len = max_text_len

def __len__(self):
    return len(self.df)

def __getitem__(self, idx):
    row = self.df.iloc[idx]

    # Завантаження зображення
    try:
        image = Image.open(
            self.images_dir / row['filename']
        ).convert('RGB')
    except Exception:
        image = Image.new('RGB', (224, 224), (0, 0, 0))

    if self.transform:
        image = self.transform(image)

    # Нормалізація тексту та токенізація
    text = ftfy.fix_text(str(row['ad_text'])) \
        if pd.notna(row['ad_text']) else ''
    enc = self.tokenizer(
        text,
        max_length=self.max_text_len,
        padding='max_length',
        truncation=True,
        return_tensors='pt'
    )

    return {
        'image':          image,
        'input_ids':      enc['input_ids'].squeeze(0),
        'attention_mask': enc['attention_mask'].squeeze(0),
        'label':          torch.tensor(float(row['label']),
                                       dtype=torch.float32),
        'filename':       row['filename']
```

```
        }


# ============================================================================
# А.2. АРХІТЕКТУРА МОДЕЛІ
# ============================================================================

class CrossAttentionFusion(nn.Module):
    """
    Модуль симетричної крос-модальної уваги.

    Реалізує двонаправлений обмін між візуальним і текстовим
    представленнями через два незалежних attention-шари:
      - v2t: зображення «запитує» у тексту релевантні ознаки
      - t2v: текст «запитує» у зображення відповідні зони

    Після кожного attention-шару застосовується residual-з'єднання
    та LayerNorm для стабілізації навчання.
    """

    def __init__(self, vision_dim: int, text_dim: int,
                 num_heads: int = 8, dropout: float = 0.1):
        super().__init__()
        self.hidden_dim = 512

        # Проекція у спільний прихований простір
        self.vision_proj = nn.Linear(vision_dim, self.hidden_dim)
        self.text_proj   = nn.Linear(text_dim,   self.hidden_dim)

        # Двонаправлені attention-шари
        self.v2t_attention = nn.MultiheadAttention(
            self.hidden_dim, num_heads,
            dropout=dropout, batch_first=True
        )
        self.t2v_attention = nn.MultiheadAttention(
            self.hidden_dim, num_heads,
            dropout=dropout, batch_first=True
        )

        # Нормалізація після кожного attention-блоку
        self.norm1 = nn.LayerNorm(self.hidden_dim)
        self.norm2 = nn.LayerNorm(self.hidden_dim)

        # MLP-класифікатор: 1024 → 256 → 64 → 1
        self.classifier = nn.Sequential(
            nn.Linear(self.hidden_dim * 2, 256),
            nn.ReLU(),
```

```
            nn.Dropout(dropout),
            nn.Linear(256, 64),
            nn.ReLU(),
            nn.Dropout(dropout),
            nn.Linear(64, 1)
        )

    def forward(self, vision_emb: torch.Tensor,
                text_emb: torch.Tensor) -> torch.Tensor:
        # Проекція у спільний простір [B, 1, 512]
        v = self.vision_proj(vision_emb).unsqueeze(1)
        t = self.text_proj(text_emb).unsqueeze(1)

        # Крос-модальна увага з residual-з'єднаннями
        v2t, _ = self.v2t_attention(query=v, key=t, value=t)
        v2t    = self.norm1(v2t + v)

        t2v, _ = self.t2v_attention(query=t, key=v, value=v)
        t2v    = self.norm2(t2v + t)

        # Конкатенація та класифікація [B, 1024] → [B, 1]
        fused = torch.cat(
            [v2t.squeeze(1), t2v.squeeze(1)], dim=1
        )
        return self.classifier(fused).squeeze(1)


class CreativeScorer(nn.Module):
    """
    Мультимодальна модель скорингу рекламних креативів.

    Архітектура:
      - Візуальний енкодер: EfficientNet-B3 (12M params, ImageNet)
        → вектор розмірності [1536]
      - Текстовий енкодер: XLM-RoBERTa-base (278M params, 100 мов)
        → CLS-токен розмірності [768]
      - Fusion: CrossAttentionFusion з проекцією 1536/768 → 512,
        двонаправленим attention та MLP-класифікатором

    Загальна кількість параметрів: ~290M
    """

    def __init__(self, dropout: float = 0.3):
        super().__init__()

        # Візуальний енкодер (без фінального класифікатора)
        self.vision_encoder = timm.create_model(
```

```
            'efficientnet_b3',
            pretrained=False,
            num_classes=0          # виключає голову класифікації
        )

        # Текстовий енкодер
        self.text_encoder = AutoModel.from_pretrained(
            'xlm-roberta-base'
        )

        # Модуль злиття з крос-модальною увагою
        self.fusion = CrossAttentionFusion(
            vision_dim=self.vision_encoder.num_features,  # 1536
            text_dim=768,
            num_heads=8,
            dropout=dropout
        )

    def forward(self, image: torch.Tensor,
                input_ids: torch.Tensor,
                attention_mask: torch.Tensor) -> torch.Tensor:

        # Візуальне представлення [B, 1536]
        vision_emb = self.vision_encoder(image)

        # Текстове представлення – CLS-токен [B, 768]
        text_out = self.text_encoder(
            input_ids=input_ids,
            attention_mask=attention_mask
        )
        text_emb = text_out.last_hidden_state[:, 0, :]

        # Злиття та класифікація → логіт [B]
        return self.fusion(vision_emb, text_emb)


# ============================================================================
# А.3. ФУНКЦІЯ ВТРАТ – LABEL SMOOTHING BCE
# ============================================================================

class LabelSmoothingBCE(nn.Module):
    """
    Binary Cross-Entropy з label smoothing.
    Зменшує впевненість моделі в мітках і покращує
    узагальнення на малих датасетах.
    """
```

```
    def __init__(self, smoothing: float = 0.1):
        super().__init__()
        self.smoothing = smoothing

    def forward(self, logits: torch.Tensor,
                labels: torch.Tensor) -> torch.Tensor:
        # Пом'якшення міток: 0 → 0.05, 1 → 0.95
        soft = labels * (1 - self.smoothing) + self.smoothing / 2
        return nn.functional.binary_cross_entropy_with_logits(
            logits, soft
        )


# ============================================================================
# A.4. ПОСТУПОВЕ РОЗМОРОЖУВАННЯ (GRADUAL UNFREEZING)
# ============================================================================

def get_stage(epoch: int) -> int:
    """
    Визначає стадію розморожування залежно від епохи:
      0 (epoch  0– 4): навчається тільки fusion MLP
      1 (epoch  5– 9): + верхні 2 шари XLM-RoBERTa
      2 (epoch 10–35): + верхні 2 шари EfficientNet-B3
    """
    if epoch < 5:
        return 0
    elif epoch < 10:
        return 1
    return 2


def set_trainable_params(model: CreativeScorer,
                         stage: int,
                         lr_fusion: float = 1e-4,
                         lr_encoder: float = 3e-5):
    """
    Налаштовує які параметри навчаються на поточній стадії
    та повертає список груп параметрів для оптимізатора.
    """
    # Заморожуємо всі параметри
    for p in model.parameters():
        p.requires_grad = False

    param_groups = []

    # Стадія 0: тільки fusion
    fusion_params = list(model.fusion.parameters())
```

```
    for p in fusion_params:
        p.requires_grad = True
    param_groups.append({'params': fusion_params, 'lr': lr_fusion})

    # Стадія 1: + верхні 2 шари XLM-RoBERTa
    if stage >= 1:
        xlmr_top = list(
            model.text_encoder.encoder.layer[-2:].parameters()
        )
        for p in xlmr_top:
            p.requires_grad = True
        param_groups.append({'params': xlmr_top, 'lr': lr_encoder})

    # Стадія 2: + верхні 2 блоки EfficientNet-B3
    if stage >= 2:
        enet_top = list(
            model.vision_encoder.blocks[-2:].parameters()
        )
        for p in enet_top:
            p.requires_grad = True
        param_groups.append({'params': enet_top, 'lr': lr_encoder})

    return param_groups


# ============================================================================
# A.5. ЦИКЛ НАВЧАННЯ
# ============================================================================

def train_model(
    model: CreativeScorer,
    train_df: pd.DataFrame,
    val_df: pd.DataFrame,
    images_dir: Path,
    tokenizer,
    device: torch.device,
    gdrive_ckpt_path: Path,
    n_epochs: int = 35,
    batch_size: int = 16,
    patience: int = 7,
):
    """
    Повний цикл навчання з:
      - поступовим розморожуванням (gradual unfreezing)
      - балансуванням класів через WeightedRandomSampler
      - label smoothing BCE як функцією втрат
      - CosineAnnealingLR як планувальником
```

```
  - early stopping за Val AUC
  - checkpoint-системою з синхронізацією на Google Drive
"""

criterion = LabelSmoothingBCE(smoothing=0.1)

# Збалансований семплер 50/50 у батчі
labels      = train_df['label'].values
class_counts = np.bincount(labels.astype(int))
weights      = 1.0 / class_counts[labels.astype(int)]
sampler      = WeightedRandomSampler(
    weights, num_samples=len(weights), replacement=True
)

train_ds = CreativeDataset(
    train_df, images_dir, tokenizer,
    transform=train_transforms
)
val_ds = CreativeDataset(
    val_df, images_dir, tokenizer,
    transform=val_transforms
)
train_loader = DataLoader(
    train_ds, batch_size=batch_size,
    sampler=sampler, num_workers=2, pin_memory=True
)
val_loader = DataLoader(
    val_ds, batch_size=batch_size * 2,
    shuffle=False, num_workers=2, pin_memory=True
)

best_val_auc  = 0.0
epochs_no_imp = 0
current_stage = -1

for epoch in range(n_epochs):
    stage = get_stage(epoch)

    # Оновлення параметрів при зміні стадії
    if stage != current_stage:
        param_groups  = set_trainable_params(model, stage)
        optimizer     = AdamW(param_groups, weight_decay=1e-2)
        scheduler     = CosineAnnealingLR(
            optimizer, T_max=n_epochs - epoch
        )
        current_stage = stage
        print(f'\n→ Стадія {stage}: '
```

```
                f'розморожено {"тільки fusion" if stage == 0 else "XLM-R top-2" if stage
== 1 else "+ ENet top-2"}')

        # ── Тренування ──────────────────────────────────────
        model.train()
        train_loss = 0.0
        for batch in train_loader:
            images = batch['image'].to(device)
            ids    = batch['input_ids'].to(device)
            mask   = batch['attention_mask'].to(device)
            labels_b = batch['label'].to(device)

            optimizer.zero_grad()
            logits = model(images, ids, mask)
            loss   = criterion(logits, labels_b)
            loss.backward()
            optimizer.step()
            train_loss += loss.item()

        scheduler.step()

        # ── Валідація ───────────────────────────────────────
        model.eval()
        all_probs, all_labels = [], []
        with torch.no_grad():
            for batch in val_loader:
                images = batch['image'].to(device)
                ids    = batch['input_ids'].to(device)
                mask   = batch['attention_mask'].to(device)
                logits = model(images, ids, mask)
                probs  = torch.sigmoid(logits).cpu().numpy()
                all_probs.extend(probs)
                all_labels.extend(batch['label'].numpy())

        val_auc = roc_auc_score(all_labels, all_probs)
        avg_loss = train_loss / len(train_loader)
        print(f'Epoch {epoch+1:02d}/{n_epochs} | '
              f'Loss: {avg_loss:.4f} | Val AUC: {val_auc:.4f}')

        # ── Checkpoint ──────────────────────────────────────
        if val_auc > best_val_auc:
            best_val_auc  = val_auc
            epochs_no_imp = 0

            local_ckpt = Path('/content/best_model_v7.pt')
            torch.save({
                'epoch':           epoch,
```

```
                'model_state':     model.state_dict(),
                'optimizer_state': optimizer.state_dict(),
                'scheduler_state': scheduler.state_dict(),
                'best_val_auc':    best_val_auc,
                'val_auc':         val_auc,
            }, local_ckpt)

            # Синхронізація з Google Drive
            shutil.copy(local_ckpt, gdrive_ckpt_path)
            print(f'  ✓ Checkpoint збережено (Val AUC: {best_val_auc:.4f})')
        else:
            epochs_no_imp += 1
            if epochs_no_imp >= patience:
                print(f'\nEarly stopping після {epoch+1} епох.')
                break

    print(f'\nНавчання завершено. Найкращий Val AUC: {best_val_auc:.4f}')
    return best_val_auc


# ============================================================================
# A.6. ЗАВАНТАЖЕННЯ ФІНАЛЬНОЇ МОДЕЛІ (INFERENCE)
# ============================================================================

def load_model(ckpt_path: Path,
               device: torch.device) -> CreativeScorer:
    """Завантажує збережений checkpoint фінальної моделі v7."""
    model = CreativeScorer(dropout=0.3).to(device)
    ckpt  = torch.load(ckpt_path, map_location=device,
                       weights_only=False)
    model.load_state_dict(ckpt['model_state'])
    model.eval()
    return model
```

# ДОДАТОК В ЛІСТИНГ ПРОГРАМНОГО КОДУ СИСТЕМИ INFERENCE ТА ПОЯСНЕННЯ РІШЕНЬ

```
import torch
import torch.nn as nn
import numpy as np
import anthropic
import ftfy
import cv2
from pathlib import Path
from PIL import Image
from torchvision import transforms
from transformers import AutoTokenizer
from pytorch_grad_cam import GradCAMPlusPlus
from pytorch_grad_cam.utils.image import show_cam_on_image

# Імпорт архітектури з Додатку А
from appendix_A_architecture_training import CreativeScorer


# ============================================================================
# Б.1. КОНВЕРТАЦІЯ ЙМОВІРНОСТІ В SCORE 0–100 (ПЕРЦЕНТИЛЬНИЙ РАНГ)
# ============================================================================

def load_train_probs(npy_path: Path) -> np.ndarray:
    """
    Завантажує збережені ймовірності навчальної вибірки.
    Файл train_probs_v5.npy генерується один раз і не
    перераховується при подальших запусках.
    """
    return np.load(npy_path)


def prob_to_score(prob: float, train_probs: np.ndarray) -> int:
    """
    Конвертує ймовірність моделі у score 0–100 на основі
    перцентильного рангу відносно навчальної вибірки.

    score = 100 означає, що лише 0% тренувальних зразків
    мали вищу ймовірність (найвищий можливий ранг).
    """
    return int(round((train_probs < prob).mean() * 100))
```

```python
def score_to_verdict(score: int) -> tuple[str, str]:
    """Повертає текстовий вердикт та рекомендацію за числовим score."""
    if score >= 80:
        return 'Сильний креатив', 'Рекомендуємо запустити.'
    elif score >= 60:
        return 'Вище середнього', \
               'Можна запустити з обмеженим бюджетом.'
    elif score >= 40:
        return 'Середній', \
               'Є потенціал для покращення перед запуском.'
    else:
        return 'Слабкий креатив', \
               'Рекомендуємо переробити перед запуском.'


# ============================================================================
# Б.2. ОБГОРТКА ДЛЯ GRAD-CAM++ (МУЛЬТИМОДАЛЬНА МОДЕЛЬ)
# ============================================================================

class VisionWrapper(nn.Module):
    """
    Ізолює візуальну гілку CreativeScorer для Grad-CAM++.

    pytorch_grad_cam очікує модель з сигнатурою forward(image) → logit.
    VisionWrapper фіксує текстові входи (input_ids, attention_mask)
    і надає стандартний однотензорний інтерфейс.
    """

    def __init__(self, model: CreativeScorer,
                 input_ids: torch.Tensor,
                 attention_mask: torch.Tensor):
        super().__init__()
        self.model        = model
        self.input_ids    = input_ids
        self.attention_mask = attention_mask

    def forward(self, image: torch.Tensor) -> torch.Tensor:
        return self.model(
            image, self.input_ids, self.attention_mask
        ).unsqueeze(-1)


# ============================================================================
# Б.3. GRAD-CAM++ ТА ЗОНАЛЬНИЙ АНАЛІЗ 3×3
# ============================================================================

def compute_gradcam(
```

```
    model: CreativeScorer,
    image_tensor: torch.Tensor,
    input_ids: torch.Tensor,
    attention_mask: torch.Tensor,
    orig_image: np.ndarray,
    device: torch.device,
) -> tuple[np.ndarray, np.ndarray, dict]:
    """
    Обчислює Grad-CAM++ теплову карту та зональний аналіз 3×3.

    Повертає:
      cam_image   – RGB-зображення з накладеною тепловою картою
      cam_raw     – нормалізована теплова карта [0, 1] розміром 224×224
      zone_scores – словник {назва_зони: z-score} для сітки 3×3
    """
    wrapper = VisionWrapper(
        model,
        input_ids.unsqueeze(0).to(device),
        attention_mask.unsqueeze(0).to(device)
    )

    target_layer = [model.vision_encoder.blocks[-1]]
    cam = GradCAMPlusPlus(
        model=wrapper,
        target_layers=target_layer
    )

    grayscale_cam = cam(
        input_tensor=image_tensor.unsqueeze(0).to(device),
        targets=None
    )[0]

    # Накладення теплової карти на оригінальне зображення
    rgb_img  = orig_image.astype(np.float32) / 255.0
    cam_image = show_cam_on_image(rgb_img, grayscale_cam, use_rgb=True)

    # Зональний аналіз 3×3
    zone_scores = _compute_zone_scores(grayscale_cam)

    return cam_image, grayscale_cam, zone_scores


def _compute_zone_scores(cam: np.ndarray) -> dict:
    """
    Ділить теплову карту на сітку 3×3 і обчислює
    нормований z-score для кожної зони:
      score = clip((mean_zone - global_mean) / global_std, -1, 1)
```

```
    Порогові значення:
      ≥ +0.3 – сильний позитивний вплив зони на рішення моделі
      0 – +0.3 – помірний вплив
      < 0 – зона слабко або негативно впливає
    """
    h, w = cam.shape
    rows = ['верх', 'центр', 'низ']
    cols = ['ліво', 'центр', 'право']

    global_mean = cam.mean()
    global_std  = cam.std() + 1e-8

    zones = {}
    row_h = h // 3
    col_w = w // 3

    for r_idx, r_name in enumerate(rows):
        for c_idx, c_name in enumerate(cols):
            r0, r1 = r_idx * row_h, (r_idx + 1) * row_h
            c0, c1 = c_idx * col_w, (c_idx + 1) * col_w
            zone_mean  = cam[r0:r1, c0:c1].mean()
            z_score    = float(np.clip(
                (zone_mean - global_mean) / global_std, -1, 1
            ))
            zones[f'{r_name}-{c_name}'] = round(z_score, 2)

    return zones


def format_zone_analysis(zone_scores: dict) -> str:
    """Форматує зональний аналіз у читабельний текст."""
    lines = ['Зони зображення (Grad-CAM):']
    for zone, score in zone_scores.items():
        if score >= 0.3:
            mark = '[+]'
        elif score >= 0:
            mark = '[ ]'
        else:
            mark = '[-]'
        lines.append(
            f'  {mark} {zone:<16} {score:+.2f}'
            f'  ({"сильний вплив" if score >= 0.3 else "помірний" if score >= 0 else
"слабкий вплив"})'
        )
    return '\n'.join(lines)
```

```
# ============================================================================
# Б.4. АНАЛІЗ ТЕКСТУ – XLM-ROBERTA ATTENTION WEIGHTS
# ============================================================================

def compute_text_attention(
    model: CreativeScorer,
    input_ids: torch.Tensor,
    attention_mask: torch.Tensor,
    tokenizer: AutoTokenizer,
    device: torch.device,
    top_n: int = 10,
) -> tuple[list, list]:
    """
    Витягує ваги уваги XLM-RoBERTa для визначення ключових слів.

    Використовує ваги уваги від CLS-токена до всіх інших токенів
    в останньому шарі (усереднення по 12 головах).

    Повертає:
      positive_words – топ-N слів з найвищими вагами (+вплив)
      negative_words – N слів з найнижчими вагами (-вплив)
    """
    model.eval()
    with torch.no_grad():
        output = model.text_encoder(
            input_ids=input_ids.unsqueeze(0).to(device),
            attention_mask=attention_mask.unsqueeze(0).to(device),
            output_attentions=True
        )

    # Ваги уваги останнього шару: [batch, heads, seq, seq]
    # CLS-рядок: [heads, seq] → усереднення → [seq]
    attn = output.attentions[-1][0, :, 0, :].mean(dim=0)
    attn = attn.cpu().numpy()

    # Декодування токенів
    tokens = tokenizer.convert_ids_to_tokens(
        input_ids.cpu().numpy()
    )

    # Агрегація субтокенів: беремо максимальну вагу для слова
    word_weights = {}
    current_word = ''
    current_max  = 0.0

    for token, weight in zip(tokens, attn):
```

```
            if token in ['<s>', '</s>', '<pad>']:
                continue
            clean = token.replace('_', '')
            if token.startswith('_') and current_word:
                word_weights[current_word] = current_max
                current_word = clean
                current_max  = float(weight)
            elif token.startswith('_'):
                current_word = clean
                current_max  = float(weight)
            else:
                current_word += clean
                current_max   = max(current_max, float(weight))

    if current_word:
        word_weights[current_word] = current_max

    # Сортування та вибір топ-N
    sorted_words = sorted(
        word_weights.items(), key=lambda x: x[1], reverse=True
    )
    positive_words = [(w, round(v, 3)) for w, v in sorted_words[:top_n]]
    negative_words = [(w, round(v, 3)) for w, v in sorted_words[-top_n:]]

    return positive_words, negative_words


# ═══════════════════════════════════════════════════════════════════════
# Б.5. AI-РЕКОМЕНДАЦІЇ ЧЕРЕЗ CLAUDE HAIKU
# ═══════════════════════════════════════════════════════════════════════

def generate_ai_recommendations(
    analysis: dict,
    api_key: str,
    tone: str = 'Стандартний',
) -> str:
    """
    Генерує actionable рекомендації щодо покращення креативу
    через Claude Haiku на основі результатів аналізу моделі.

    Параметри:
      analysis – словник з score, prob, зонами, словами, YOLO-об'єктами
      api_key  – ключ Anthropic API
      tone     – режим: 'Стандартний' | 'Детальний' | 'Короткий (3 пункти)'
    """
    score         = analysis.get('score', 0)
    prob          = analysis.get('prob', '0%')
```

```
    ad_text       = analysis.get('ad_text', '')[:500]
    pos_zones     = analysis.get('positive_zones', [])
    neg_zones     = analysis.get('negative_zones', [])
    pos_words     = analysis.get('positive_words', [])
    neg_words     = analysis.get('negative_words', [])
    yolo_objects  = analysis.get('yolo_objects', [])

    tone_instruction = {
        'Стандартний':       'Надай 3-5 конкретних рекомендацій.',
        'Детальний':         'Надай детальний аналіз з 5-7 рекомендаціями та
обґрунтуванням.',
        'Короткий (3 пункти)': 'Надай рівно 3 найважливіші рекомендації.',
    }.get(tone, 'Надай 3-5 конкретних рекомендацій.')

    prompt = f"""Ти - досвідчений маркетолог з аналізу рекламних креативів.
Проаналізуй результати AI-скорингу та надай практичні рекомендації.

РЕЗУЛЬТАТИ АНАЛІЗУ:
- Score: {score}/100 (ймовірність успіху: {prob})
- Текст оголошення: {ad_text}
- Зони з сильним позитивним впливом: {', '.join(pos_zones) if pos_zones else 'немає'}
- Зони зі слабким впливом: {', '.join(neg_zones) if neg_zones else 'немає'}
- Ключові слова з найвищою увагою: {', '.join([w for w, _ in pos_words[:5]])}
- Слова з низькою увагою: {', '.join([w for w, _ in neg_words[:3]])}
- Виявлені об'єкти: {', '.join(yolo_objects[:5]) if yolo_objects else 'немає'}

ІНСТРУКЦІЯ: {tone_instruction}
НЕ використовуй markdown заголовки (# ## ###).
Починай одразу з "1.", "2.", "3." тощо.
Пиши українською мовою."""

    client = anthropic.Anthropic(api_key=api_key)
    response = client.messages.create(
        model='claude-haiku-4-5-20251001',
        max_tokens=800,
        messages=[{'role': 'user', 'content': prompt}]
    )
    return response.content[0].text


# ============================================================================
# Б.6. ПОВНИЙ INFERENCE PIPELINE
# ============================================================================

val_transforms = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
```

```
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                         std=[0.229, 0.224, 0.225])
])


def predict_creative(
    image_path: str,
    ad_text: str,
    model: CreativeScorer,
    tokenizer: AutoTokenizer,
    train_probs: np.ndarray,
    device: torch.device,
    anthropic_api_key: str,
    tone: str = 'Стандартний',
) -> dict:
    """
    Повний inference pipeline для одного рекламного креативу.

    Кроки:
      1. Нормалізація тексту (ftfy)
      2. Підготовка зображення (resize, normalize)
      3. Токенізація (max_length=256)
      4. Forward pass → P(label=1)
      5. Конвертація prob → score 0–100 (перцентиль)
      6. Grad-CAM++ + зональний аналіз 3×3
      7. Text attention weights
      8. AI-рекомендації через Claude Haiku

    Повертає словник з усіма результатами аналізу.
    """
    # 1. Нормалізація тексту
    clean_text = ftfy.fix_text(ad_text) if ad_text else ''

    # 2. Підготовка зображення
    orig_image = np.array(
        Image.open(image_path).convert('RGB').resize((224, 224))
    )
    image_tensor = val_transforms(
        Image.fromarray(orig_image)
    )

    # 3. Токенізація
    enc = tokenizer(
        clean_text,
        max_length=256,
        padding='max_length',
        truncation=True,
```

```
    return_tensors='pt'
)
input_ids     = enc['input_ids'].squeeze(0)
attention_mask = enc['attention_mask'].squeeze(0)

# 4. Forward pass
model.eval()
with torch.no_grad():
    logit = model(
        image_tensor.unsqueeze(0).to(device),
        input_ids.unsqueeze(0).to(device),
        attention_mask.unsqueeze(0).to(device)
    )
    prob = torch.sigmoid(logit).item()

# 5. Score і вердикт
score          = prob_to_score(prob, train_probs)
verdict, rec   = score_to_verdict(score)

# 6. Grad-CAM++ + зональний аналіз
cam_image, cam_raw, zone_scores = compute_gradcam(
    model, image_tensor, input_ids, attention_mask,
    orig_image, device
)

pos_zones = [z for z, v in zone_scores.items() if v >= 0.3]
neg_zones = [z for z, v in zone_scores.items() if v < 0]

# 7. Text attention
pos_words, neg_words = compute_text_attention(
    model, input_ids, attention_mask, tokenizer, device
)

# 8. Збирання аналізу для AI-рекомендацій
analysis = {
    'score':          score,
    'prob':           f'{prob * 100:.1f}%',
    'ad_text':        clean_text,
    'positive_zones': pos_zones,
    'negative_zones': neg_zones,
    'positive_words': pos_words,
    'negative_words': neg_words,
    'yolo_objects':   [],        # заповнюється окремо при YOLO-аналізі
}

ai_recs = generate_ai_recommendations(
    analysis, anthropic_api_key, tone=tone
```

```
    )

    return {
        'score':       score,
        'prob':        prob,
        'verdict':     verdict,
        'rec':         rec,
        'cam_image':   cam_image,
        'cam_raw':     cam_raw,
        'zone_scores': zone_scores,
        'pos_words':   pos_words,
        'neg_words':   neg_words,
        'ai_recs':     ai_recs,
        'analysis':    analysis,
    }
```