Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Комп'ютерного моделювання та інтелектуальних технологій
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

«Дослідження ймовірнісних моделей дифузії для інтелектуальних технологій генерації зображень за їх текстовим представленням»
(тема)

Виконав:
здобувач II року навчання, групи СПРм-24-1
Микита ЦЕПОЧКО
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
Освітня програма Системне проєктування
(повна назва освітньої програми)

Керівник проф. каф. КМІТ
Володимир БЕЗКОРОВАЙНИЙ
(посада, власне ім'я, прізвище)

Допускається до захисту
Завідувач кафедри КМІТ _____ Ігор ГРЕБЕННІК
(підпис) (власне ім'я, прізвище)

2026 р.

Я, як студент ХНУРЕ, розумію та підтримую політику закладу з академічної доброчесності. Я не надавав та не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

(підпис) *Микита ЦЕПОЧКО*

Кваліфікаційна робота оформлена у відповідності до вимог діючих стандартів та методичних вказівок.

Матеріали кваліфікаційної роботи не містять відомостей, що заборонені для опублікування у відкритих виданнях.

Попередній захист проведено.

Керівник кваліфікаційної роботи  *В. В. Безкоровайний*

16.05.2026

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*

Кафедра *Комп'ютерного моделювання та інтелектуальних технологій*

Рівень вищої освіти *другий (магістерський)*

Спеціальність *122 Комп'ютерні науки*

Освітня програма *Системне проєктування*

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2026 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві *Цепочку Микиті Геннадійовичу*
(прізвище, ім'я, по батькові)

Тема роботи *«Дослідження ймовірнісних моделей дифузії для інтелектуальних технологій генерації зображень за їх текстовим представленням»*
затверджена наказом університету від *«06» квітня 2026 р. №268Ст.*
2. Термін подання студентом роботи до екзаменаційної комісії *19.05.2026 р.*
3. Вихідні дані до роботи *Об'єкт дослідження – процес генерації зображень на основі їх текстового представлення. Предмет дослідження – ймовірнісні дифузійні моделі, механізми мультимодальної інтеграції текстових і зображувальних представлень. Мета дослідження: підвищення ефективності технологій генерації зображень на основі їх текстового представлення за рахунок використання ймовірнісних моделей дифузії. Предмет розробки: інформаційна система генерації зображень на базі латентної дифузійної моделі з модифікованим механізмом перехресної уваги на основі алгоритмів прискореного семплюванн. Базові моделі: генеративні змагальні мережі; варіаційні автоенкодери; дифузійні ймовірнісні моделі. Базові систем синтезу зображень: DALL·E 3, Midjourney v6, Stable Diffusion XL, Imagen 2; Adobe Firefly 2. Технічне забезпечення: IBM-сумісний персональний комп'ютер.*

4. Перелік питань, що потрібно опрацювати в роботі *Аналіз предметної області дослідження. Огляд особливостей задачі синтезу зображень за текстовим описом, підходів до генеративного моделювання зображень, сучасних систем синтезу зображень. Порівняльний аналіз систем синтезу зображень. Розроблення постановки задачі дослідження. Аналіз сучасних архітектур ймовірнісних моделей дифузії та алгоритмів їх реалізації. Проєктування та розробка програмного забезпечення. Розроблення діаграм та реалізація компонентів системи. Тренування розробленої моделі. Тестування функціональних можливостей системи. Аналіз отриманих результатів.*

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) *У вигляді презентації кресленики, схеми, плакати та/або комп'ютерні ілюстрації (слайди) на аркушах формату А4, що включаються до тексту пояснювальної записки або складу додатків (10–15 аркушів): концептуальна схема задачі синтезу зображень за текстовим описом; схеми прямого та зворотного дифузійних процесів; схема архітектури латентної дифузійної моделі; графіки характеристик алгоритмів вибірки і розкладів шуму; діаграма компонентів системи; діаграма діяльності системи; графіки результатів навчання розробленої моделі; екранні форми розробленого програмного засобу.*

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	*Отримання завдання на кваліфікаційну роботу*	*06.04.2026*	*Виконано*
2	*Аналіз предметної області*	*13.04.2026*	*Виконано*
3	*Постановка задачі на розробку системи*	*15.04.2026*	*Виконано*
4	*Проєктування та розробка компонентів*		
	системи	*20.04.2026*	*Виконано*
5	*Оформлення пояснювальної записки*	*27.05.2026*	*Виконано*
6	*Подання закінченої роботи науковому*		
7	*керівникові*	*28.04.2026*	*Виконано*
8	*Усунення зауважень наукового керівника*	*30.04.2026*	*Виконано*
9	*Підготовка презентації*	*05.04.2026*	*Виконано*
10	*Перевірка оригінальності тексту*	*06.05.2026*	*Виконано*
11	*Подання роботи на рецензування*	*10.05.2026*	*Виконано*
12	*Попередній захист*	*15.05.2026*	*Виконано*
13	*Подання роботи до екзаменаційної комісії*	*18.05.2026*	*Виконано*

Дата видачі завдан *«06» квітня 2026* р.

Здобувач _____ *Микита ЦЕПОЧКО*
(підпис)

Керівник роботи _____ *проф. каф. СТ Володимир БЕЗКОРОВАЙНИЙ*
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 79 с., 1 табл., 10 рис., 2 додатки, 33 джерела інформації.

ГЕНЕРАЦІЯ ЗОБРАЖЕНЬ, ДИФУЗІЙНА МОДЕЛЬ, КЛАСИФІКАТОРНО-ВІЛЬНЕ КЕРУВАННЯ, ЛАТЕНТНИЙ ПРОСТІР, МЕХАНІЗМ УВАГИ, ТЕКСТОВЕ КОДУВАННЯ, U-NET.

Об'єктом дослідження є процес генерації зображень на основі їх текстового представлення.

Предметом дослідження є ймовірнісні дифузійні моделі, механізми мультимодальної інтеграції текстових і зображувальних представлень.

Мета дослідження: підвищення ефективності технологій генерації зображень на основі їх текстового представлення за рахунок використання ймовірнісних моделей дифузії.

Методи дослідження – теорія ймовірностей та стохастичне числення, глибинне навчання та оптимізація нейронних мереж, архітектурне моделювання на основі трансформерів та згорткових мереж, об'єктно-орієнтоване проєктування програмних систем.

У роботі спроєктовано та реалізовано систему умовної генерації зображень на базі латентної дифузійної моделі з модифікованим механізмом перехресної уваги на основі алгоритмів прискореного семплювання DDIM та DPM-Solver.

Галузь застосування – цифровий дизайн та створення ілюстративного контенту, розширення навчальних датасетів у науково-дослідних роботах з комп'ютерного зору, платформи для тонкого налаштування під специфічні предметні галузі методами LoRA та DreamBooth, а також локальні системи генерації зображень без залежності від хмарних сервісів.

ABSTRACT

Master's Thesis: 79 pages, 1 table, 10 figures, 2 appendices, 33 references.

IMAGE GENERATION, DIFFUSION MODEL, CLASSIFIER-FREE CONTROL, LATENT SPACE, ATTENTION MECHANISM, TEXT CODING, U-NET.

The object of the study is the process of generating images based on their text representation.

The subject of the study is probabilistic diffusion models, mechanisms of multimodal integration of text and image representations.

The purpose of the study: to increase the efficiency of image generation technologies based on their text representation through the use of probabilistic diffusion models.

Research methods - probability theory and stochastic calculus, deep learning and optimization of neural networks, architectural modeling based on transformers and convolutional networks, object-oriented design of software systems.

In the work, a system of conditional image generation based on a latent diffusion model with a modified cross-attention mechanism based on the DDIM and DPM-Solver accelerated sampling algorithms is designed and implemented.

The field of application is digital design and creation of illustrative content, expansion of training datasets in computer vision research, platforms for fine-tuning for specific subject areas using LoRA and DreamBooth methods, as well as local image generation systems without dependence on cloud services.

ЗМІСТ

ВСТУП

Стрімкий розвиток методів глибокого навчання впродовж останнього десятиліття відкрив принципово нові можливості у галузі автоматизованої генерації візуального контенту. Серед усіх підходів до синтезу зображень особливе місце посідають ймовірнісні дифузійні моделі, які за відносно короткий проміжок часу від моменту свого теоретичного обґрунтування перетворились на домінуючу парадигму генеративного моделювання, витіснивши генеративно-змагальні мережі та варіаційні автокодери з позицій лідерів у переважній більшості бенчмаркових досліджень.

Актуальність обраної теми дослідження зумовлена кількома взаємопов'язаними чинниками.

По-перше, незважаючи на стрімкий практичний прогрес у цій галузі, теоретичні засади ймовірнісних дифузійних моделей залишаються предметом активних наукових дискусій, а їхній зв'язок з класичними результатами стохастичного числення та варіаційного виводу потребує систематичного дослідження.

По-друге, ефективна реалізація систем умовної генерації зображень за текстовим представленням вимагає глибокого розуміння механізмів інтеграції мовних та зображувальних модальностей у спільному латентному просторі, що є самостійною науковою задачею на перетині обробки природної мови та комп'ютерного зору.

По-третє, практичне застосування дифузійних моделей стримується їхньою обчислювальною складністю, зокрема необхідністю виконання сотень або тисяч кроків зворотного ланцюга під час інференсу, що зумовлює актуальність дослідження прискорених стратегій семплювання та їхнього впливу на якість генерованих результатів.

Об'єктом дослідження є процес генерації зображень на основі їх текстового

представлення.

Предметом дослідження є ймовірнісні дифузійні моделі, механізми мультимодальної інтеграції текстових і зображувальних представлень.

Метою кваліфікаційної роботи є підвищення ефективності технологій генерації зображень на основі їх текстового представлення за рахунок використання ймовірнісних моделей дифузії.

Для досягнення мети кваліфікаційної роботи необхідно виконати наступні завдання:

– дослідити теоретичні засади застосування генеративних моделей у задачах синтезу зображень та проаналізувати математичний апарат ймовірнісних підходів до генеративного моделювання;

– проаналізувати сучасні архітектури ймовірнісних моделей дифузії для генерації зображень на основі текстових даних;

– спроєктувати та розробити програмне забезпечення для генерації зображень на основі моделі дифузії;

– провести експериментальне тестування функціональних можливостей розробленої системи генерації зображень на основі ймовірнісної дифузійної моделі;

Наукова новизна дослідження полягає у розробці модифікованої архітектури умовної дифузійної моделі для синтезу зображень за текстовим описом, яка, на відміну від стандартної реалізації Stable Diffusion XL, використовує збільшену кількість голів механізму перехресної уваги (з 8 до 16) у денойзинговій U-Net мережі, що забезпечує більш точне атрибутивне зв'язування між токенами та просторовими елементами синтезованої сцени.

Практичне значення отриманих результатів полягає у створенні функціонального програмного застосунку для генерації зображень за довільним текстовим описом на основі відкритої дифузійної моделі, що допускає локальне розгортання без залежності від хмарних сервісів. Розроблений застосунок може бути використаний у галузі цифрового дизайну та створення ілюстративного

контенту, у науково-дослідних роботах з комп'ютерного зору для розширення навчальних датасетів, а також як базова платформа для подальшого тонкого налаштування під специфічні предметні галузі методами LoRA та DreamBooth.

За результатами кваліфікаційної роботи опубліковано тези на трьох науково-практичних конференціях, у яких представлено основні теоретичні положення дослідження.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Особливості задачі синтезу зображень за текстовим описом

Задача синтезу зображень за текстовим описом полягає у побудові системи, здатної автоматично генерувати зображення, зміст якого відповідає довільному природньомовному запиту [1].

На відміну від суміжних задач розпізнавання зображень або машинного перекладу, дана задача є принципово обернено-визначеною: одному текстовому опису може відповідати нескінченна множина коректних візуальних інтерпретацій. Наприклад, запит «кіт на підвіконні вдень» допускає різноманітні варіації породи, пози, освітлення, кольору меблів і заднього фону – і всі вони є семантично правомірними відповідями. Це суттєво відрізняє задачу від детермінованих перетворень і вимагає стохастичних, ймовірнісних підходів до її вирішення. Семантичний розрив між природньою мовою і зображенням є головним технічним викликом.

Текст описує реальність через абстрактні символьні структури: іменники позначають класи об'єктів, прикметники – їхні атрибути, прийменники – просторові відношення. Зображення, натомість, є числовим масивом значень яскравості пікселів, де семантика прихована у складних нелінійних залежностях між мільйонами числових значень. Система синтезу має навчитися будувати міст між цими двома принципово різними просторами представлення інформації. Практична цінність задачі визначається широким колом застосувань. У галузі цифрового дизайну та маркетингу такі системи дозволяють генерувати ілюстративний контент, рекламні матеріали і концепт-арт за словесним описом, скорочуючи час виробництва візуального матеріалу від годин до секунд [2].

У медичній візуалізації синтетичні зображення застосовуються для розширення обмежених навчальних датасетів діагностичних систем. В освіті та науці забезпечується автоматичне формування наочних матеріалів, адаптованих

до конкретної теми. В інженерному проєктуванні генерація концептуальних візуалізацій за технічним описом прискорює ітерацію між замовником і виконавцем. Успішне вирішення задачі вимагає одночасного застосування методів із кількох науково-технічних дисциплін: обробки природньої мови для побудови семантичних представлень тексту; теорії ймовірностей і статистичного моделювання для апроксимації розподілу реальних зображень; архітектур глибокого навчання для реалізації відображення між семантичними просторами; комп'ютерного зору для оцінки якості синтезованих результатів. Саме ця мультидисциплінарність обумовлює значну складність задачі і активний науковий інтерес до неї [3].

1.2 Огляд підходів до генеративного моделювання зображень

За останнє десятиліття у галузі генеративного моделювання зображень відбулося декілька хвиль технологічного розвитку, кожна з яких визначалась домінуванням певної парадигми. Кожен клас характеризується власним математичним апаратом, оптимізаційною ціллю та специфічними перевагами і обмеженнями з точки зору якості і керованості контенту.

1.2.1 Генеративні змагальні мережі

Генеративні змагальні мережі (GAN), запропоновані Гудфелоу та ін. у 2014 році, реалізують генерацію через навчання пари конкуруючих нейронних мереж. Мережа-генератор синтезує зображення із випадкового вектора шуму, намагаючись ввести в оману мережу-дискримінатор, яка, в свою чергу, намагається відрізнити синтезовані зображення від реальних. Навчання через протидію двох мереж призводить до поступового покращення якості синтезу до рівня, за якого дискримінатор не може впевнено відрізнити реальне зображення від синтезованого.

Основними перевагами GAN є висока швидкість вибірки – генерація зображення зводиться до одного прямого проходу через мережу-генератор – та здатність досягати гострої деталізації. Водночас цей клас моделей страждає від принципових недоліків. Першим є нестабільність навчання: рівновага між генератором і дискримінатором є нестійкою, і процес навчання може зазнавати коливань або повного зриву. Другим є вироджування генератора – явище, за якого мережа зосереджується на синтезі обмеженого набору зображень, ігноруючи різноманітність реального розподілу. Треба зазначити, що умовні модифікації GAN (AttnGAN, StackGAN) суттєво поступились дифузійним моделям за метрикою якості після 2021 року і нині відійшли на другий план [5, 6].

1.2.2 Варіаційні автоенкодери

Варіаційні автоенкодери (VAE), розроблені Велінгом у 2013 році, навчають стохастичне кодування вхідних зображень у компактний латентний простір із подальшим декодуванням. Навчання VAE є значно стабільнішим за GAN і забезпечує структурований неперервний латентний простір, придатний для семантичної інтерполяції. Проте пряме використання VAE для синтезу зображень обмежене надмірною розмитістю результатів, спричиненою усередненням у ймовірнісному просторі.

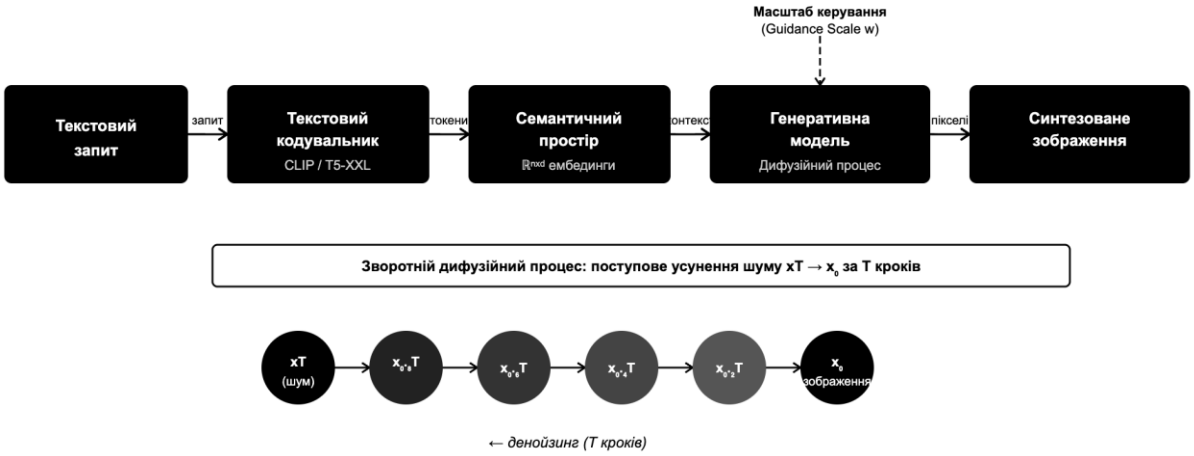
1.2.3 Дифузійні ймовірнісні моделі

Дифузійні ймовірнісні моделі (DDM), теоретичні засади яких сформовані у роботах Со–Дікстейна та ін. (2015) і практично реалізовані Хо та ін. (2020), відновлюють зображення з гаусівського шуму шляхом ітеративного застосування навченого денойзингового перетворення. Прямий процес поступово додає шум до зображення за T кроків, доводячи його до чистого гаусівського розподілу. Зворотній процес, що параметризується нейронною мережею, навчається

відтворювати оригінальне зображення, поступово видаляючи шум крок за кроком. Ключовою перевагою дифузійних моделей є відсутність проблеми вироджування генератора і стабільне навчання через мінімізацію функції втрат денойзингу. За метриками якості (FID, CLIP-Score) цей клас моделей перевершує GAN при достатній кількості кроків вибірки.

Підтримка умовної генерації реалізується через механізм безкласифікаторного керування, що дозволяє регулювати ступінь відповідності синтезованого зображення текстовій умові.

На рисунку 1.1 зображено концептуальну схему задачі синтезу зображення за текстовим описом із зазначенням ролі дифузійного процесу (додаток А).



Джерело: власна розробка автора.

Рисунок 1.1 – Концептуальна схема задачі синтезу зображень за текстовим описом

Відповідно до рисунку 1.2, процес синтезу включає чотири послідовні етапи. На першому етапі вхідний текстовий запит надходить до текстового кодувальника, що перетворює його у послідовність числових векторів (ембедингів) у семантичному просторі. На другому етапі отримані ембединги подаються як умова до генеративної моделі, яка реалізує зворотній дифузійний процес: починаючи з випадкового гаусівського шуму, модель за T кроків

поступово усуває шум, формуючи зображення, семантично узгоджене з текстом. На третьому етапі результат декодується у простір пікселів і формується кінцеве зображення. Масштаб керування (Guidance Scale, w) регулює компроміс між різноманіттям результатів і точністю відповідності текстовій умові: при більших значеннях w зображення точніше відповідає запиту, але різноманітність зменшується.

1.3 Огляд сучасних систем синтезу зображень за текстовим описом

Починаючи з 2021 року, коли дифузійні моделі продемонстрували перевагу над GAN за якістю синтезу, галузь зазнала стрімкого розвитку. Провідні технологічні компанії і відкрите наукове співтовариство запропонували ряд систем, що стали індустріальним стандартом. Нижче наведено характеристику найбільш значущих із них.

1.3.1 DALL·E

DALL·E 3 є третьою ітерацією системи від компанії OpenAI і являє собою принципово вдосконалену версію порівняно з попередницями. Ключова архітектурна новація полягає у попередній деталізації вхідного запиту за допомогою великої мовної моделі GPT-4: перед подачею до дифузійного модуля коротка фраза користувача автоматично розгортається у розширений детальний опис, що суттєво підвищує семантичну точність результату. Наслідком є здатність системи відтворювати у зображенні всі елементи складного запиту без їх вибіркового ігнорування – проблеми, притаманної попереднім поколінням моделей. DALL·E 3 першим серед доступних систем продемонстрував прийнятну якість відтворення текстових написів усередині зображень, що тривалий час лишалось принципово невирішеним завданням. Система доступна

через хмарний програмний інтерфейс OpenAI та інтегрована у застосунок ChatGPT.

Вагові параметри і детальна архітектура моделі не оприлюднені; локальне розгортання і тонке налаштування неможливі.

1.3.2 Midjourney

Midjourney v6 – закрита комерційна система, що здобула широке визнання у художній і дизайнерській спільноті. За результатами численних суб'єктивних оцінювань людьми, система демонструє найвищі естетичні показники серед доступних рішень, особливо у відтворенні художнього стилю, освітлення і образотворчої композиції. Взаємодія зі системою здійснюється виключно через Discord-бот або власний вебінтерфейс; публічний програмний інтерфейс відсутній. Система підтримує розширений набір параметрів управління результатом: співвідношення сторін зображення, ступінь стилізації, рівень варіативності та художньої нестандартності. Відсутність програмного інтерфейсу унеможливлює інтеграцію у конвеєри і налаштування під специфічні предметні галузі.

1.3.3 Stable Diffusion XL

Stable Diffusion XL (SDXL) є повністю відкритою дифузійною системою від компанії Stability AI, що реалізує принцип латентної дифузії. – виконання дифузійного процесу не у просторі пікселів, а у стисненому латентному просторі нейромережевого кодувальника. Це суттєво знижує обчислювальну складність генерації без значної втрати якості. SDXL використовує двокомпонентну систему текстового кодування, що паралельно опрацьовує запит двома різними енкодерами для більш повного охоплення семантики. Відкрита ліцензія Apache 2.0 дозволяє комерційне використання, локальне розгортання без залежності від

хмарних сервісів, а також тонке налаштування методами низькорангової адаптації, персоналізованого навчання і просторового обумовлення. Широка екосистема підтримуваних розширень (DreamBooth, Inpaint) зробила SDXL гнучкою платформою для дослідницьких і промислових застосувань.

1.3.4 Imagen

Imagen 2 є продовженням каскадної дифузійної серії від Google DeepMind, що розпочалась із публікації оригінальної Imagen у 2022 році (Saharia et al.). Система вирізняється принциповим архітектурним рішенням щодо вибору текстового кодувальника: на відміну від більшості конкурентів, що покладаються на контрастивно навчені енкодери типу CLIP, Imagen 2 використовує T5-XXL – трансформерну модель типу encoder-decoder із розмірністю прихованого простору 4096 вимірів. Це майже у чотири рази перевищує розмірність енкодерів на базі CLIP (768–1024 вимірів), що забезпечує значно більш детальне і контекстно чутливе представлення складних природньомовних запитів. Дослідження, проведені командою Saharia et al., емпірично підтвердили, що збільшення потужності текстового кодувальника справляє більший вплив на якість семантичного узгодження, ніж еквівалентне збільшення параметрів денойзингової мережі, що є нетривіальним і практично важливим висновком. Архітектура Imagen 2 реалізує каскадний дифузійний підхід, що принципово відрізняється від одностадійної латентної дифузії, застосованої у Stable Diffusion XL. Каскад складається з декількох послідовних спеціалізованих дифузійних моделей, кожна з яких відповідає за окремий діапазон просторових роздільних здатностей. Перша модель синтезує низькороздільне базове зображення розміром 64×64 пікселів, що відповідає загальній семантичній структурі сцени. Наступні каскадні моделі поступово підвищують роздільну здатність: 64×64 → 256×256 → 1024×1024 пікселів, при цьому кожна стадія доповнює зображення дрібнішими деталями, текстурами і локальними структурами. Такий підхід

дозволяє ефективно розподілити складність завдання між стадіями: ранні моделі відповідають за глобальну семантику і композицію, пізніші – за високочастотні деталі.За кількісними метриками якості Imagen 2 демонструє найвищі показники серед порівнюваних систем: значення метрики Фреше (FID) становить ~7–10, семантичної точності (CLIP-Score) ~0.35. Система також відзначається особливо точним відтворенням семантично складних запитів із множиною об'єктів, просторовими відношеннями і числовими атрибутами – категорії запитів, у яких системи на базі CLIP-енкодерів традиційно демонструють слабші результати через обмежену розмірність семантичного простору.

Крім того, Imagen 2 показує кращі результати у відтворенні текстових написів у зображеннях порівняно з попередніми версіями.

1.3.5 Adobe Firefly

Adobe Firefly є комерційною генеративною системою від компанії Adobe, глибоко інтегрованою у пакет Adobe Creative Cloud. Система доступна безпосередньо у застосунках Photoshop, Illustrator, Express та InDesign, що забезпечує безшовну інтеграцію генеративного синтезу у усталені дизайнерські робочі процеси без необхідності перемикання між інструментами. Це є значною практичною перевагою для профе сійної аудиторії, яка вже працює в екосистемі Adobe. Принциповою відмінністю Firefly 2 від усіх інших розглянутих систем є джерело навчальних даних. Тоді як DALL·E 3, Stable Diffusion XL, Imagen 2 і Midjourney навчались на масштабних корпусах, зібраних із відкритого інтернету – що породжує юридичні суперечки щодо авторського права, - Firefly 2 навчалась виключно на матеріалах Adobe Stock, що перебувають під комерційною ліцензією, і зображеннях із відкритою ліцензією Creative Commons. Це рішення безпосередньо усуває правові ризики при комерційному використанні синтезованого контенту: Adobe надає комерційну гарантію відповідності, що є критично важливим для корпоративного середовища, де питання інтелектуальної

власності регулюються юридичними відділами. Саме цей аспект виокремив Firefly 2 як унікальне рішення у контексті розгорнутих у 2023–2024 роках судових позовів проти постачальників генеративних систем. З технічного боку Firefly 2 базується на дифузійній моделі з власним текстовим кодувальником Adobe, навченим на мультимодальному корпусі Adobe Stock. Підтримується генерація зображень розміром до 2048×2048 пікселів з масштабуванням, широкий вибір співвідношень сторін і стилів, а також розширена типографічна генерація. Firefly 2 демонструє особливо сильні результати у синтезі маркетингових матеріалів, рекламних банерів, текстур і типографіки, що відображає специфіку навчальних даних Adobe Stock, де зазначені категорії добре представлені. Водночас система поступається DALL·E 3 та Imagen 2 у відтворенні фотореалістичних сцен із складною багатоелементною просторовою композицією і природніми середовищами – наслідок менш різноманітного навчального корпусу порівняно з моделями, навченими на відкритих інтернет-даних.

1.4 Порівняльний аналіз систем синтезу зображень за текстовим описом

З метою комплексного порівняння розглянутих систем за технічними, функціональними та ліцензійними характеристиками у таблиці 1.1 наведено їх зіставлення за дванадцятьма критеріями, що безпосередньо визначають придатність системи для науково-дослідних і промислових застосувань.

Аналіз даних таблиці 1.1 дозволяє зробити такі узагальнення щодо поточного стану галузі. За критерієм відкритості та відтворюваності результатів єдиною повністю відкритою системою є Stable Diffusion XL. Лише ця система надає можливість локального розгортання без залежності від стороннього сервісу, тонкого налаштування під специфічну предметну галузь і незалежної верифікації науково-дослідних результатів.

Таблиця 1.1 – Порівняльний аналіз систем синтезу зображень за текстовим описом

Критерій	DALL·E 3	Midjourney v6	Stable Diffusion XL	Imagen 2	Adobe Firefly 2
Розробник, рік	OpenAI, 2023	Midjourney, 2024	Stability AI, 2023	Google DeepMind, 2023	Adobe, 2023
Принцип генерації	Дифузійна модель (латент. простір)	Дифузійна (закрита реал.)	Латентна дифузійна (LDM)	Каскадна дифузійна	Дифузійна
Текстовий кодувальник	GPT-4 + CLIP	Не розкрито	OpenCLIP ViT-H + CLIP ViT-L	T5-XXL (4096-вим.)	CLIP + власний
Роздільна здатність	1024×1024	2048×2048 (+масштаб.)	1024×1024	1024×1024	2048×2048 (+масштаб.)
Метрика FID	~12–15	Не розкрито	~17–22	~7–10	~15–20
Семантична точність	Висока (CLIP-S ~0.33)	Дуже висока	Висока (CLIP-S ~0.31)	Дуже висока (~0.35)	Висока
Доступність / ліцензія	API (платно) закритий код	Підписка, закритий код	Відкриті ваги Apache 2.0	Обмежений API Google	Adobe CC підписка
Локальне розгортання	Ні	Ні	Так	Ні	Ні
Донавчання	Ні	Ні	Так (LoRA, DreamBooth)	Обмежено	Ні

Джерело: розроблено автором на основі [10, 11, 12].

Для академічних досліджень, де відтворюваність і прозорість є принциповими вимогами, ця властивість є вирішальною.

За кількісними показниками якості лідирують Imagen 2 (FID ~7–10, CLIP-Score ~0.35) і DALL·E 3 (FID ~12–15, CLIP-Score ~0.33). Перевага Imagen 2 обумовлена значно вищою розмірністю текстового кодувальника T5-XXL (4096 вимірів), що забезпечує більш повне кодування семантики складних запитів. DALL·E 3 досягає подібного ефекту через попередню деталізацію запиту засобами GPT-4, а не через розширення простору ембедингів. За критерієм гнучкості налаштування Stable Diffusion XL не має конкурентів: підтримка низькорангової адаптації (LoRA), персоналізованого навчання (DreamBooth), просторового обумовлення (ControlNet) та інших методів тонкого налаштування дозволяє адаптувати модель до будь-якої специфічної предметної галузі без повного перенавчання.

Саме ця властивість є ключовою для науково-дослідної розробки. Загальний висновок: жодна із закритих комерційних систем (DALL·E 3, Midjourney v6, Imagen 2) не дозволяє проводити систематичне наукове дослідження механізмів синтезу через відсутність доступу до вагових параметрів і вихідного коду.

1.5 Актуальні проблеми та обмеження сучасних підходів

Незважаючи на значний прогрес у якості синтезованих зображень, що спостерігається починаючи з 2020 року, сучасні дифузійні системи мають ряд невирішених проблем як фундаментального теоретичного, так і прикладного характеру. Нижче систематизовано ключові технічні обмеження, що визначають актуальні напрями досліджень у галузі.

Проблема атрибутивного зв'язування є одним із найбільш досліджуваних недоліків сучасних систем умовної генерації. Вона полягає у систематичній

нездатності денойзингових моделей надійно асоціювати атрибути – колір, форму, матеріал, просторову орієнтацію, кількісні характеристики – з конкретними об'єктами у сценах із кількома суб'єктами.

Типовий прояв: запит «червоний куб і синя сфера» стабільно призводить до взаємної заміни кольорів між об'єктами або їх неповного відтворення. Технічна причина полягає у механізмі перехресної уваги денойзингової мережі: токени текстового запиту і просторові позиції латентного представлення зображення взаємодіють через скалярний добуток у загальному просторі ключів і запитів, що не гарантує вибіркового прив'язування атрибутивного токена до просторово локалізованого об'єкта.

Додатковим чинником є статистичне змішування атрибутивних асоціацій у навчальних даних: корпуси типу LAION-5B містять пари зображення–опис, де атрибути рідко вживаються у строго прив'язаному до об'єктів синтаксичному контексті, що призводить до розмитих ймовірнісних асоціацій у навченій моделі. Запропоновані рішення, зокрема маніпуляції картами уваги [14], та структуроване синтаксичне декомпозування запиту дозволяють частково усунути проблему, однак не вирішують її системно.

Проблема семантичних галюцинацій проявляється у синтезі анатомічно і фізично неправдоподібних зображень: надлишкова або неправильно орієнтована кількість пальців на кістках руки, злиття облич при перекритті суб'єктів, порушення закону збереження маси при взаємодії об'єктів, генерація неіснуючих текстових символів замість конкретних написів.

Причина носить фундаментальний характер: дифузійна модель є апроксиматором розподілу $p_{data}(x|y)$ – вона моделює статистику зображень у навчальному корпусі, а не фізику або геометрію реального світу. Відсутність явного геометричного пріору, анатомічного шаблону або фізичного рушія призводить до того, що у зонах низької щільності навчальних даних (нестандартні пози, незвичні ракурси, рідкісні об'єкти) модель генерує статистично правдоподібний, але фізично некоректний результат.

Дослідження у напрямі інтеграції 3D-пріорів і нейросимволічних обмежень [15, 16] є перспективними, але поки не перейшли у реалізації.

Проблема захисту інтелектуальної власності набуває дедалі більшої юридичної і технічної гостроти. З технічного боку задокументовано явище меморизації: дифузійні моделі здатні з ненульовою імовірністю відтворювати зображення з навчальної вибірки або їх близькі варіації при певних текстових запитах [17], що безпосередньо порушує авторські права власників оригінальних зображень. Окремою проблемою є можливість цільової імітації художнього стилю конкретних авторів через підбір промпту або тонке налаштування – без їхньої згоди і будь-якої компенсації.

З технічної точки зору розробляються методи диференціально-приватного навчання і водяних знаків для контенту, проте жоден із них не досяг рівня промислового стандарту [18, 19, 20].

1.6 Постановка задачі дослідження

На підставі проведеного аналізу предметної галузі сформульовано задачу кваліфікаційної роботи. Встановлено, що попри значну практичну цінність і наявність конкурентних промислових рішень, механізми функціонування ймовірнісних моделей і їх адаптація до специфічних прикладних умов потребують систематичного наукового дослідження.

Метою кваліфікаційної роботи є підвищення ефективності технологій генерації зображень на основі їх текстового представлення за рахунок використання ймовірнісних моделей дифузії. Мета передбачає дослідження ймовірнісних моделей дифузії та розробку на їх основі інтелектуальної системи синтезу зображень за текстовим описом.

Для досягнення поставленої мети необхідно виконати такі завдання:

– дослідити теоретичні засади ймовірнісних дифузійних моделей, включаючи математичний апарат прямого і зворотного марківських процесів,

оптимізаційні цілі та механізми умовної генерації;

– систематизувати і порівняти сучасні системи синтезу зображень за текстовим описом та обґрунтувати вибір базової моделі для розробки;

– спроєктувати програмну систему умовного синтезу зображень на базі латентної дифузійної моделі із підтримкою текстового керування;

– реалізувати розроблену систему у вигляді функціонального програмного застосунку з інтерфейсом взаємодії користувача;

– провести експериментальне оцінювання реалізованої системи за кількісними метриками якості синтезу і семантичної відповідності результатів текстовим запитам.

2 АНАЛІЗ СУЧАСНИХ АРХІТЕКТУР ЙМОВІРНІСНИХ МОДЕЛЕЙ ДИФУЗІЇ ДЛЯ ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ НА ОСНОВІ ТЕКСТОВИХ ДАНИХ

2.1 Математичні засади ймовірнісних дифузійних моделей

Ймовірнісні дифузійні моделі формалізують задачу генерації зображень як двофазний стохастичний процес. Перша фаза – прямий процес – поступово перетворює зразок реальних даних x_0 на ізотропний гаусівський шум шляхом послідовного додавання шуму. Друга фаза – зворотній процес – відновлює вихідний зразок із шуму за допомогою параметричної нейронної мережі, навченої апроксимувати оберну стохастичну динаміку. Такий підхід дозволяє уникнути нестабільності навчання, притаманної генеративним змагальним мережам, і водночас досягти вищої якості та різноманітності синтезованих зображень. Нехай $x_0 \sim q(x_0)$ – зразок із розподілу реальних даних (зображення). Прямий марківський процес поступово додає гаусівський шум протягом T кроків.

Умовний розподіл кожного кроку визначається виразом (2.1):

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} \cdot x_{t-1}, \beta_t I), \quad (2.1)$$

де $\beta_t \in (0,1)$ – коефіцієнт дисперсії шуму на кроці t, що задається розкладом шуму.

З огляду на властивість замкненості гаусівського розподілу, можна отримати зашумлену версію x_t безпосередньо з x_0 без ітеративного застосування кожного кроку.

Замкнена форма $q(x_t|x_0)$ визначається виразом (2.2):

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t} \cdot x_0, (1-\bar{\alpha}_t)I\right), \quad (2.1)$$

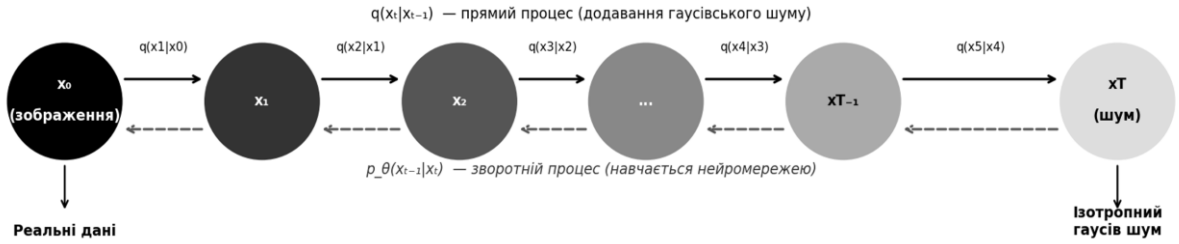
де $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$ – накопичений добуток коефіцієнтів збереження сигналу.

Це дозволяє у процесі навчання безпосередньо синтезувати зашумлений зразок для довільного кроку t за формулою (2.3):

$$x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{1-\bar{\alpha}_t} \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I). \tag{2.3}$$

При $T \to \infty$ і коректно підібраному розкладі $\{\beta_t\}$ розподіл x_t наближається до ізотропного гаусівського $\mathcal{N}(0, I)$.

Це забезпечує просту початкову умову для зворотнього процесу: генерація починається з вибірки $x_t \sim \mathcal{N}(0, I)$. На рисунку 2.1 наведено візуальну схему прямого і зворотнього дифузійних процесів (додаток А).



Джерело: розроблено автором на основі [18, 19].

Рисунок 2.1 – Прямий та зворотній дифузійні процеси

Зворотній процес моделює поступове відновлення x_0 із x_t шляхом параметризації умовних розподілів нейронною мережею. Умовний розподіл зворотнього кроку апроксимується виразом (2.4):

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\big(x_{t-1};\, \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\big), \tag{2.4}$$

де μ_θ і Σ_θ – параметризовані нейронною мережею середнє і дисперсія зворотнього кроку.

Нейронна мережа навчається передбачати доданий шум ε.

Перепараметризація дозволяє виразити середнє зворотнього кроку через передбачений шум у вигляді виразу (2.5):

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t, t)\right). \quad (2.5)$$

Навчання мережі виконується шляхом мінімізації спрощеної функції втрат, що є варіаційним кордоном логарифмічної правдоподібності. Спрощений варіант функції втрат (ELBO) набуває вигляду (2.6):

$$\mathcal{L}_{simple} = \mathbb{E}t, x_0, \varepsilon\left(\|\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}\cdot x_0 + \sqrt{1-\bar{\alpha}_t}\cdot \varepsilon, t)\|^2\right), \quad (2.6)$$

де очікування береться по рівномірно розподіленому $t \sim U[1,T]$, зразку x_0 і шуму $\varepsilon \sim \mathcal{N}(0,I)$.

Ця ціль має інтуїтивну інтерпретацію: мережа навчається передбачати шум ε, що був доданий до x_0 на кроці t.

Спрощений ELBO ігнорує ваги, що на практиці покращує якість сприйняття синтезованих зображень.

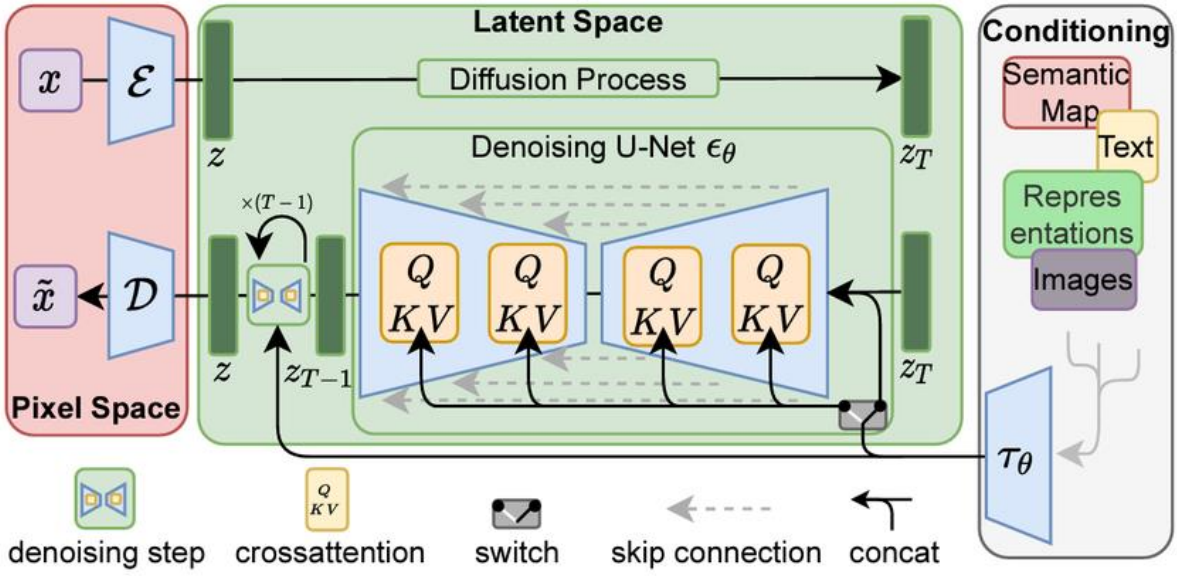
2.2 Латентні дифузійні моделі

Принциповим архітектурним рішенням, що зробило дифузійні моделі практично застосовними для синтезу зображень високої роздільної здатності, стало перенесення дифузійного процесу з простору пікселів у стиснений латентний простір. Цей підхід, відомий як латентні дифузійні моделі [20], дозволяє знизити обчислювальну складність генерації на один-два порядки без суттєвої втрати якості результату.

Архітектура LDM складається з трьох ключових компонентів: (1) автоенкодер для стиснення зображення у латентний простір та його відновлення;

(2) денойзингова мережа, що реалізує дифузійний процес у латентному просторі;
(3) текстовий кодувальник для формування семантичного представлення умови.

На рисунку 2.2 наведено повну схему архітектури LDM (додаток А).



Джерело: [21].

Рисунок 2.2 – Архітектура латентної дифузійної моделі

Автоенкодер складається з кодувальника $\varepsilon(\cdot)$ і декодера $D(\cdot)$. Кодувальник стискає вхідне зображення $x \in \mathbb{R}^{HxWx3}$ у латентне представлення $z = \varepsilon(x) \in \mathbb{R}^{hxwxc}$ з коефіцієнтом стиснення $f = H/h = W/w$ (типово $f \in \{4, 8\}$). Декодер відновлює зображення $\tilde{x} = D(\tilde{z}_0)$ із відновленого латентного стану. Якість автоенкодера безпосередньо обмежує максимальну якість синтезованих зображень: артефакти декодування не можуть бути усунені денойзинговою мережею. Для регуляризації латентного простору застосовуються або підхід VQ (дискретна квантизація), або KL-штраф (неперервний латентний простір, близький до VAE).

Для синтезу зображень знаходиться компроміс між стисненням і якістю відновлення. Дифузійний процес у латентному просторі повністю аналогічний піксельному (2.1) – (2.6), але виконується для z замість x, що знижує розмірність з $H \times W \times C$ до $h \times w \times c$. Умовна генерація реалізується через механізм перехресної

уваги між активаціями денойзингової мережі і текстовими ембедингами.

Текстовий кодувальник $\tau\_\theta(y)$ перетворює вхідний запит y у послідовність векторів $\tau_\theta(y) \in \mathbb{R}^{Nxd}$. Механізм перехресної уваги між прихованим станом $\varphi_i(z_t)$ і текстовим контекстом обчислюється за формулою (2.8):

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V, \quad (2.8)$$

де $Q = W\_Q \cdot \varphi_i(z_t)$, $K = W\_K \cdot \tau\_\theta(y)$, $V = W\_V \cdot \tau\_\theta(y)$ – лінійні проєкції прихованих станів денойзингової мережі та текстових ембедингів;

d – розмірність простору уваги.

Такий механізм забезпечує просторово адаптивне обумовлення: кожна просторова позиція латентного стану може вибірково звертатись до релевантних токенів текстового опису.

Денойзингова мережа ε_θ є ключовим нейромережевим компонентом, що відповідає за якість і семантичну відповідність синтезованих зображень.

У системах, побудованих на основі Stable Diffusion, використовується архітектура U-Net, яка реалізує ієрархічний механізм кодування та декодування із вбудованими модулями трансформерної уваги. Її структура складається з двох симетричних частин – кодувальника та декодувальника, що з'єднані між собою skip-з'єднаннями для передачі просторової інформації між відповідними рівнями мережі.

На кожному етапі кодувального шляху послідовно застосовуються ResNet-блок із часовим ембедингом дифузійного кроку t, механізм самоуваги, блок перехресної уваги з текстовим контекстом, а також операція зменшення просторової розмірності ознак. Декодувальний шлях виконує зворотне відновлення просторової структури даних, поєднуючи поточні ознаки з skip-картами, отриманими з відповідних рівнів кодувача. Часовий ембединг кроку t формується за принципом позиційного кодування, аналогічного до того, що

використовується у трансформерних архітектурах.

Для цього застосовується синусоїдальне представлення кроку дифузії, яке визначається формулою (2.9):

$$e(t,2i) = sin\left(\frac{t}{10000^{\frac{2i}{d}}}\right), \quad e(t,2i+1) = cos\left(\frac{t}{10000^{\frac{2i}{d}}}\right), \tag{2.9}$$

де d визначає розмірність ембедингу, а i – позицію елемента у векторному представленні.

Сформований вектор далі обробляється дворівневим персептроном та передається до кожного ResNet-блоку за допомогою механізму Adaptive Group Normalization (AdaGN), який забезпечує адаптацію параметрів нормалізації, зокрема масштабування та зсуву, відповідно до поточного етапу денойзингу.

Механізм Self-Attention забезпечує взаємодію між просторовими позиціями латентного представлення, що дає змогу моделювати глобальні залежності у структурі зображення. Оскільки обчислювальна складність Self-Attention квадратично залежить від кількості просторових токенів, цей механізм переважно використовується на глибших рівнях U-Net, де просторове розширення ознак є меншим. Для оптимізації використання пам’яті та пришвидшення обчислень застосовується Flash Attention – удосконалена реалізація механізму уваги, яка завдяки блочній обробці матриці уваги зменшує складність споживання пам’яті з $O(N^2)$ до $O(N)$. У межах запропонованої в кваліфікаційній роботі архітектури механізм перехресної уваги було модифіковано шляхом збільшення кількості голів уваги порівняно зі стандартною реалізацією Stable Diffusion.

Підвищення кількості голів із 8 до 16 дає можливість одночасно аналізувати більшу кількість незалежних закономірностей відповідності між текстовими токенами та просторовими позиціями латентного простору, що

сприяє покращенню атрибутивного зв’язування та більш точному відтворенню складних текстових описів [23, 24].

2.3 Безкласифікаторне керування

Безкласифікаторне керування, запропоноване у роботі [25], є базовим механізмом підсилення семантичної відповідності у сучасних умовних дифузійних моделях. На відміну від підходів із класифікаторним керуванням, цей механізм не потребує окремого навчання додаткового класифікатора та не вимагає обчислення градієнтів за активаціями мережі під час процесу генерації, що суттєво спрощує архітектуру та підвищує ефективність вибірки. У процесі навчання денойзингова мережа функціонує одночасно в умовному та безумовному режимах: із певною ймовірністю $p_{uncond} \approx 0{,}1$ текстова умова c відкидається та замінюється порожнім рядком або нульовим вектором, що дозволяє моделі навчитися як залежній, так і незалежній від текстового контексту генерації. Під час вибірки скориговане передбачення шуму формується як лінійна комбінація умовного та безумовного прогнозів, де безумовна компонента використовується як базовий напрямок, а різниця між умовним і безумовним прогнозами масштабується коефіцієнтом керування www, який визначає силу прив’язки генерації до текстової умови. За значення $w = 1$ модель працює у звичайному умовному режимі без додаткового підсилення, тоді як при $w > 1$ підвищується узгодженість результату з текстовим описом за рахунок часткового зменшення різноманітності згенерованих варіантів.

На практиці для фотореалістичних зображень зазвичай використовуються значення w у межах 7.0–9.0, тоді як для художніх або стилізованих генерацій ефективнішими є нижчі значення – близько 3.0–6.0.

З точки зору байєсівської інтерпретації механізм CFG можна розглядати як апроксимацію градієнта логарифма правдоподібності умовного розподілу, що спрямовує процес денойзингу у бік більшої відповідності заданій умові.

Оскільки кожен крок генерації із застосуванням CFG потребує двох окремих прямих проходів через мережу – для умовного та безумовного режимів, – це фактично подвоює обчислювальну складність процесу вибірки. Подальшим розвитком цього підходу є використання негативного промпту, коли замість порожнього вектора безумовної умови застосовується ембединг тексту, що описує небажані характеристики.

2.4 Алгоритми прискорення вибірки

Одним із ключових обмежень класичних дифузійних моделей є висока обчислювальна складність процесу генерації, що безпосередньо пов’язано з необхідністю виконання великої кількості послідовних кроків денойзингу. У стандартних реалізаціях DDPM кількість кроків зворотного процесу зазвичай становить близько $T \approx 1000$, що забезпечує високу якість реконструкції розподілу даних, однак робить процес генерації надто повільним для практичного використання в інтерактивних системах, вебзастосунках та задачах генерації в реальному часі. Значна кількість ітерацій призводить до високих витрат обчислювальних ресурсів, збільшення часу інференсу та підвищених вимог до графічного обладнання.

Сучасні методи прискорення вибірки базуються на різних математичних підходах, серед яких найбільш поширеними є детерміновані апроксимації зворотного процесу, числове інтегрування звичайних диференціальних рівнянь та багатокрокові методи високого порядку.

2.4.1 Алгоритм DDIM

Алгоритм DDIM, запропонований Song та співавторами у 2020 році [26], є одним із перших ефективних підходів до прискорення вибірки у дифузійних

моделях. На відміну від класичного DDPM, де процес генерації формалізується як стохастичний марківський ланцюг із випадковою вибіркою шуму на кожному кроці, DDIM вводить детерміновану форму зворотного процесу, яка зберігає ті самі маргінальні розподіли даних, але дозволяє значно зменшити кількість ітерацій денойзингу. Основна ідея методу полягає у побудові нестохастичної траєкторії переходів у латентному просторі, що апроксимує вихідний дифузійний процес без необхідності повного проходження всіх часових кроків.

Детермінований зворотний перехід описується виразом (2.10):

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \cdot \hat{x}_0(x_t) + \sqrt{1-\bar{\alpha}_{t-1}} \cdot \varepsilon_\theta(x_t, t), \quad (2.10)$$

де $\hat{x}_0(x_t)$ є оцінкою початкового незашумленого зображення, отриманою на основі поточного латентного стану x_t та передбаченого мережею шуму.

На відміну від DDPM, у DDIM відсутній додатковий випадковий шум на кожному кроці, що робить траєкторію генерації повністю детермінованою. Така властивість забезпечує відтворюваність результатів: однаковий початковий латентний код x_T за незмінних умов завжди приводить до генерації одного й того самого зображення.

Це має важливе практичне значення для задач семантичного редагування та інверсії зображень, оскільки дозволяє виконувати контрольовані маніпуляції у латентному просторі без втрати стабільності процесу генерації. Додатковою перевагою DDIM є можливість субдискретизації часової шкали, тобто використання лише підмножини кроків денойзингу замість повного проходження через усі часові стани. Завдяки цьому кількість кроків генерації може бути зменшена приблизно до 10–50, що забезпечує суттєве скорочення часу інференсу при збереженні прийнятного рівня візуальної якості. Проте надмірне скорочення кількості кроків у DDIM часто призводить до втрати дрібних деталей, появи артефактів та погіршення текстурної узгодженості зображення, що обмежує ефективність методу при ультрашвидкій генерації.

2.4.2 Алгоритм DPM-Solver++

Подальшим розвитком методів прискореної вибірки став алгоритм DPM-Solver++, запропонований Lu та співавторами у 2022 році. На відміну від DDIM, який використовує детерміновану дискретну апроксимацію дифузійного процесу, DPM-Solver++ формалізує зворотний процес генерації як задачу числового інтегрування звичайного диференціального рівняння (ODE). Такий підхід дозволяє застосовувати класичні методи чисельного аналізу для побудови більш точних апроксимацій траєкторії денойзингу та значно скорочувати кількість необхідних кроків генерації.

В основі алгоритму лежить probability flow ODE, яке описує безперервну еволюцію латентного стану у процесі денойзингу та визначається виразом (2.11):

$$dx_t = \left(f(t) \cdot x_t - \frac{g(t)^2}{2} \cdot \nabla x_t log\, p(x_t)\right) dt, \tag{2.11}$$

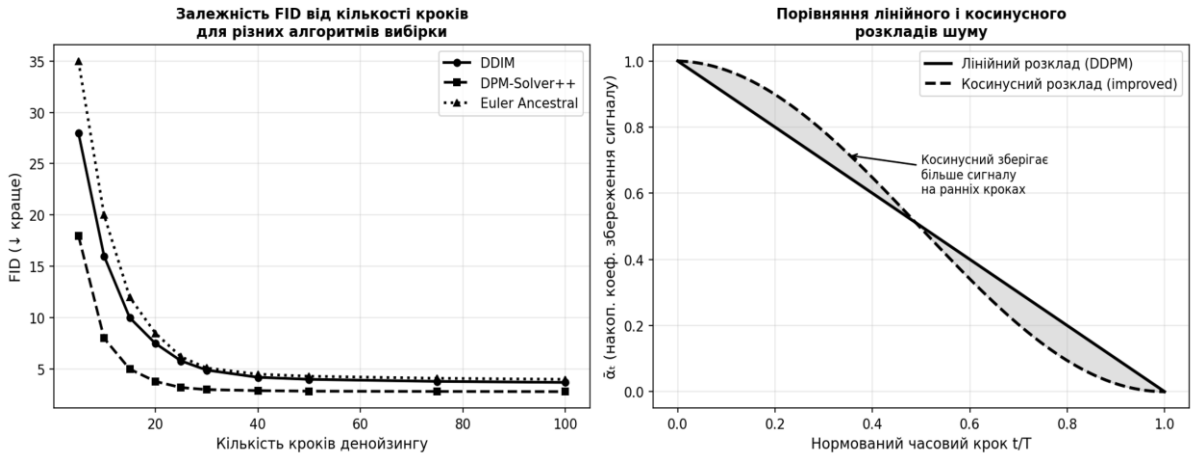
де $f(t)$ і $g(t)$ визначають коефіцієнти дрейфу та дифузії відповідно, а $\nabla x_t log\, p(x_t)$ описує градієнт логарифма густини розподілу.

У практичних реалізаціях цей градієнт апроксимується нейронною мережею денойзингу, що передбачає шум на кожному часовому кроці. DPM-Solver++ використовує багатокрокові схеми інтегрування другого та третього порядку, побудовані на основі розкладу Тейлора, що дозволяє значно точніше оцінювати траєкторію переходів між латентними станами порівняно з однокроковими методами.

Завдяки використанню чисельних методів високого порядку DPM-Solver++ забезпечує якісну генерацію вже при 15–25 кроках денойзингу, що приблизно у два-три рази ефективніше порівняно з DDIM при зіставному рівні візуальної якості. Крім високої швидкодії, алгоритм характеризується кращою стабільністю відновлення дрібних текстур, більш плавними переходами між просторовими структурами та нижчим рівнем накопичення помилок під час

генерації. Саме тому DPM-Solver++ набув широкого поширення у сучасних реалізаціях Stable Diffusion і був обраний як основний алгоритм вибірки у межах розробленої системи завдяки оптимальному співвідношенню швидкості генерації, обчислювальної ефективності та якості синтезованих зображень.

На рисунку 2.3 представлено характеристики алгоритмів вибірки і розкладів шуму (додаток А).



Джерело; розроблено автором на основі [27].

Рисунок 2.3 – Характеристики алгоритмів вибірки і розкладів шуму

На рисунку 2.3 наведено порівняльний аналіз характеристик сучасних алгоритмів вибірки та різних розкладів шуму. Ліва частина рисунка демонструє залежність метрики FID від кількості кроків денойзингу для алгоритмів DDIM, DPM-Solver++ та Euler Ancestral. Отримані результати показують, що DPM-Solver++ досягає стабілізації значення FID вже при приблизно 20–25 кроках генерації, тоді як DDIM потребує більшої кількості ітерацій – sблизько 30–40 кроків – для досягнення подібного рівня якості.

Права частина рисунка ілюструє відмінності між лінійним і косинусним розкладами шуму. Косинусний розклад характеризується повільнішим руйнуванням сигналу на ранніх етапах дифузії, завдяки чому у латентному представленні довше зберігається корисна інформація про структуру зображення. Це позитивно впливає на відтворення дрібних деталей, текстур та

високочастотних компонентів зображення, що особливо важливо при генерації фотореалістичного контенту високої роздільної здатності.

2.5 Архітектура Stable Diffusion XL та її вдосконалення

Подальший розвиток латентних дифузійних моделей привів до появи архітектури Stable Diffusion XL, яка є суттєво вдосконаленою версією попередніх моделей сімейства Stable Diffusion та орієнтована на генерацію зображень вищої роздільної здатності, кращої композиційної узгодженості й точнішого відтворення текстових описів. На відміну від Stable Diffusion 1.x і 2.x, архітектура SDXL реалізує більш масштабну багаторівневу систему текстового обумовлення, збільшену денойзингову мережу та вдосконалені механізми уваги, що дозволяє значно підвищити деталізацію та фотореалістичність синтезованих зображень. Однією з ключових особливостей SDXL є використання дворівневої архітектури генерації, що складається з базової моделі (base model) та моделі уточнення (refiner model). Базова модель виконує основне формування композиції сцени, глобальної структури та семантичного наповнення зображення, тоді як refiner-модель здійснює додаткову обробку на фінальних етапах денойзингу, покращуючи текстури, контури, освітлення та високочастотні деталі. Такий підхід дозволяє розділити задачі глобального та локального синтезу між окремими моделями, що позитивно впливає на стабільність генерації та якість кінцевого результату. Архітектурно SDXL зберігає концепцію латентної дифузійної моделі, однак містить значно більшу U-Net мережу із розширеною кількістю каналів та глибшими блоками уваги. У порівнянні з попередніми версіями Stable Diffusion, модель використовує збільшений латентний простір, що забезпечує точніше представлення дрібних деталей і складних структур. Крім того, у SDXL суттєво вдосконалено механізми Cross-Attention, які відповідають за інтеграцію текстового контексту у процес генерації. Це дозволяє більш точно співставляти текстові токени з просторовими елементами сцени та покращує

атрибутивне зв’язування між описом і синтезованим зображенням. Важливою особливістю SDXL є використання двох незалежних текстових енкодерів, які працюють паралельно та формують багаторівневе семантичне представлення текстового запиту. Один енкодер відповідає за загальне семантичне розуміння промпту, тоді як другий спеціалізується на деталізації стилістичних та контекстних характеристик. Отримані ембедінги комбінуються у спільному просторі ознак та подаються до блоків перехресної уваги денойзингової мережі. Така архітектура дозволяє покращити інтерпретацію складних текстових описів, багатокомпонентних сцен та стилізованих запитів. У SDXL також було вдосконалено механізм часових ембедингів та систему conditioning signals. Окрім стандартного текстового обумовлення, модель враховує додаткові параметри генерації, зокрема роздільну здатність зображення, aspect ratio та параметри кадрування. Це дозволяє суттєво покращити стабільність композиції при генерації нестандартних форматів зображень і зменшує кількість артефактів, пов’язаних із деформацією просторової структури сцени.

У межах даної кваліфікаційної роботи архітектура SDXL була додатково модифікована з метою покращення семантичної узгодженості та деталізації складних сцен. Основним напрямом вдосконалення стало збільшення кількості голів у механізмі Cross-Attention, що дозволило підвищити здатність моделі одночасно аналізувати більшу кількість незалежних взаємозв’язків між текстовими токенами та просторовими ознаками латентного простору. Крім того, було оптимізовано параметри CFG-керування та модифіковано стратегію семплінгу з використанням DPM-Solver++, що дозволило скоротити кількість кроків генерації без суттєвої втрати якості. Додатково в архітектурі було реалізовано вдосконалену систему негативного промптингу, яка дозволяє ефективніше пригнічувати появу небажаних артефактів, помилкових анатомічних структур та некоректних текстур. Це особливо важливо при генерації складних багатофігурних сцен і фотореалістичних зображень високої деталізації. Також було проведено оптимізацію параметрів розкладу шуму, що

дозволило зберігати більшу кількість корисного сигналу на ранніх етапах дифузії та покращити відновлення високочастотних деталей.

2.6 Підходи узгодженого навчання в моделях дифузійного типу

Окрім класичних дифузійних моделей, що ґрунтуються на багатокроковому стохастичному процесі денойзингу, у сучасній літературі активно розвивається клас моделей узгодженого (consistency-based) навчання, які спрямовані на радикальне зменшення кількості кроків генерації аж до одного або декількох функціональних викликів нейронної мережі. Основна ідея полягає у побудові функції, що забезпечує інваріантність представлення латентного стану відносно часу дифузійного процесу, тобто модель навчається відображенню, яке узгоджує стани різних рівнів шуму у спільний простір.

Формально, модель можна розглядати як функцію $f_{\boldsymbol{\theta}}(x_t, t)$, яка задовольняє умову узгодженості (2.12):

$$f_{\boldsymbol{\theta}}(x_t, t) \approx f_{\boldsymbol{\theta}}(x_s, s), t > s, \tag{2.12}$$

що означає стабільність латентного представлення незалежно від рівня зашумлення. Функція втрат у загальному вигляді задається як мінімізація різниці між передбаченнями на різних часових кроках (2.13):

$$L_{cons} = \mathbb{E}x^0, t, s\,(\|f_\theta(x_t, t) - f_\theta(x_s, s)\|^2), \tag{2.13}$$

де x_t і x_s відповідають різним рівням зашумлення одного й того самого зразка.

Ключовою властивістю таких моделей є можливість обхідного відновлення чистого зображення без необхідності ітеративного розв’язання зворотного дифузійного процесу. Це досягається завдяки навчанню моделі безпосередньо апроксимувати консистентний потік траєкторій у латентному

просторі. У практичних реалізаціях consistency-підхід дозволяє зменшити кількість кроків генерації до 1–4 при збереженні прийнятної візуальної якості, що робить його перспективним для задач реального часу.

Додатково такі моделі можуть бути інтерпретовані як наближення безперервного flow-відображення (2.14):

$$\frac{dx}{dt} = v_\theta(x, t), \tag{2.14}$$

де v_θ є параметризованим векторним полем, що описує детерміновану еволюцію даних у напрямку зменшення шуму.

Таким чином, consistency-моделі займають проміжне положення між score-based та ODE-формалізмами, поєднуючи властивості детермінованої генерації та навчання через узгодження багатошкальних представлень.

3 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Огляд використаних технічних засобів

Для реалізації системи генерації зображень на основі ймовірнісної дифузійної моделі було використано набір сучасних інструментів розробки, зокрема мову програмування Python, а також програмні бібліотеки Torch, TorchVision, diffusers і CLIP.

3.1.1 Мова програмування Python

Python є високорівневою інтерпретованою мовою програмування з динамічною системою типізації та автоматичним керуванням пам’яттю, що робить її зручною для швидкої розробки та експериментування в галузі машинного навчання і глибинного навчання. Виконання програм базується на інтерпретаторі CPython, який забезпечує сумісність із великою кількістю зовнішніх бібліотек та розширень. Однією з ключових переваг Python є його екосистема наукових бібліотек, що включає інструменти для чисельних обчислень і роботи з багатовимірними масивами, такі як NumPy, а також високорівневі фреймворки для побудови нейронних мереж, серед яких TensorFlow, PyTorch і Keras. Завдяки механізму інтеграції з модулями, написаними на C і C++, Python дозволяє ефективно виконувати обчислювально складні операції через нативні розширення, що суттєво підвищує продуктивність. Крім того, взаємодія з оптимізованими бібліотеками лінійної алгебри, такими як BLAS і LAPACK, забезпечує ефективну реалізацію матричних операцій, які є основою алгоритмів оптимізації та методів градієнтного спуску. Використання GPU-прискорення через технології CUDA та cuDNN дозволяє значно підвищити швидкість навчання моделей шляхом паралельного виконання тензорних операцій. Python також підтримує роботу з

великими обсягами даних за допомогою бібліотек pandas, Dask і Vaex, що дозволяють ефективно організовувати як послідовну, так і розподілену обробку даних. У сфері машинного навчання він виступає універсальним інтерфейсом до таких інструментів, як scikit-learn для класичних алгоритмів навчання з учителем, а також XGBoost і LightGBM для градієнтного бустингу.

Фреймворки PyTorch і TensorFlow реалізують різні підходи до побудови обчислювальних графів, при цьому забезпечуючи автоматичне диференціювання (autograd), що є критично важливим для навчання багатопараметричних моделей. Додатково, засоби багатопоточності та багатопроцесорності, реалізовані в модулях threading і multiprocessing, дозволяють організовувати паралельні обчислення, хоча їх ефективність може бути обмежена через особливості блокування інтерпретатора (GIL).

3.1.2 Бібліотека Torch

Torch є високопродуктивною бібліотекою для роботи з багатовимірними тензорами, яка широко застосовується в задачах машинного та глибинного навчання. Вона інтегрується з Python та забезпечує ефективне виконання обчислень як на центральному процесорі (CPU), так і на графічних прискорювачах (GPU), використовуючи технології CUDA (а в окремих випадках OpenCL).

Архітектура Torch побудована таким чином, щоб забезпечувати гнучку взаємодію між низькорівневими чисельними операціями, лінійною алгеброю та високорівневими компонентами нейронних мереж. Основною структурою даних виступають тензори, які є узагальненням багатовимірних масивів і дозволяють ефективно виконувати паралельні операції над великими обсягами даних. Важливою складовою бібліотеки є механізм автоматичного диференціювання, реалізований у модулі torch.autograd, який динамічно будує граф обчислень і обчислює градієнти для довільних операцій. Це забезпечує можливість навчання

моделей за допомогою методу зворотного поширення помилки. Для побудови нейронних мереж використовується модуль torch.nn, який надає набір готових компонентів, таких як шари, функції активації, механізми регуляризації та нормалізації. Оптимізація параметрів моделей здійснюється за допомогою torch.optim, що включає різноманітні алгоритми, зокрема SGD, Adam і RMSprop.

Додатково Torch підтримує серіалізацію моделей через функції torch.save та torch.load, що дозволяє зберігати та відновлювати стан моделей, спрощуючи їх використання та інтеграцію в системи розгортання.

3.1.3 Бібліотека Diffusers

Diffusers – це спеціалізована бібліотека, розроблена для реалізації та застосування дифузійних генеративних моделей. Вона побудована на базі PyTorch і надає уніфікований інтерфейс для роботи з сучасними архітектурами генерації зображень, такими як DDPM, LDM та Stable Diffusion. Бібліотека оптимізована для роботи з сучасними апаратними прискорювачами, зокрема GPU, а також підтримує змішану точність обчислень (mixed precision), що дозволяє зменшити обчислювальні витрати без суттєвої втрати якості генерації. Центральним елементом Diffusers є концепція Pipeline – високорівневого конвеєра, який об’єднує всі необхідні компоненти процесу генерації. До таких компонентів належать попередньо натреновані моделі, токенізатори тексту (у випадку мультимодальних задач), механізми дифузії та процедури денойзингу.

Наприклад, реалізації StableDiffusionPipeline, TextToImagePipeline та ImageInpaintingPipeline дозволяють виконувати генерацію зображень за текстовим описом, редагування зображень або відновлення пошкоджених ділянок. Модульна архітектура бібліотеки дає можливість комбінувати різні компоненти та обирати оптимальні стратегії дифузійного процесу. Diffusers також інтегрується з екосистемою Hugging Face, зокрема з бібліотекою Transformers, що забезпечує можливість використання мовних моделей, таких як

CLIP або інші архітектури, для побудови мультимодальних систем.

3.1.4 Бібліотека TorchVision

TorchVision є розширенням екосистеми PyTorch, орієнтованим на задачі комп'ютерного зору та обробки зображень. Вона надає комплекс інструментів, що охоплюють етапи підготовки даних, аугментації, роботи з датасетами, а також використання готових архітектур нейронних мереж, призначених для аналізу візуальної інформації. Однією з ключових переваг TorchVision є наявність модулів для попередньої обробки зображень, які реалізовані через torchvision.transforms. Даний компонент забезпечує широкий набір операцій над зображеннями, включаючи масштабування, нормалізацію, геометричні перетворення (повороти, віддзеркалення), а також внесення стохастичних змін, таких як шум або випадкові деформації. Завдяки цьому автоматизується процес data augmentation, що дозволяє підвищити стійкість і узагальнювальну здатність моделей.

TorchVision інтегрована з форматом тензорів PyTorch, що забезпечує ефективну обробку даних та оптимальне використання пам'яті при роботі з великими обсягами зображень. Це дозволяє виконувати операції над батчами даних із високою продуктивністю як на CPU, так і на GPU.

Бібліотека також включає модуль torchvision.datasets, який надає стандартизований доступ до широко використовуваних наборів даних у сфері комп'ютерного зору, таких як CIFAR, ImageNet, COCO та MNIST. У поєднанні з DataLoader із PyTorch це забезпечує ефективну організацію потокового завантаження даних, паралельну обробку та оптимізацію використання обчислювальних ресурсів під час навчання моделей.

Ще одним важливим компонентом є torchvision.models, що містить набір попередньо навчених нейронних архітектур, зокрема ResNet, VGG, DenseNet, MobileNet, EfficientNet, а також трансформерні моделі, адаптовані для задач

комп’ютерного зору, наприклад Vision Transformer.

Використання попередньо навчених ваг, отриманих на великих датасетах (ImageNet), дозволяє ефективно застосовувати підхід трансферного навчання, скорочуючи час та покращуючи якість результатів.

3.2 Діаграма компонентів системи

На рисунку 3.1 представлена діаграма компонентів системи (додаток А).

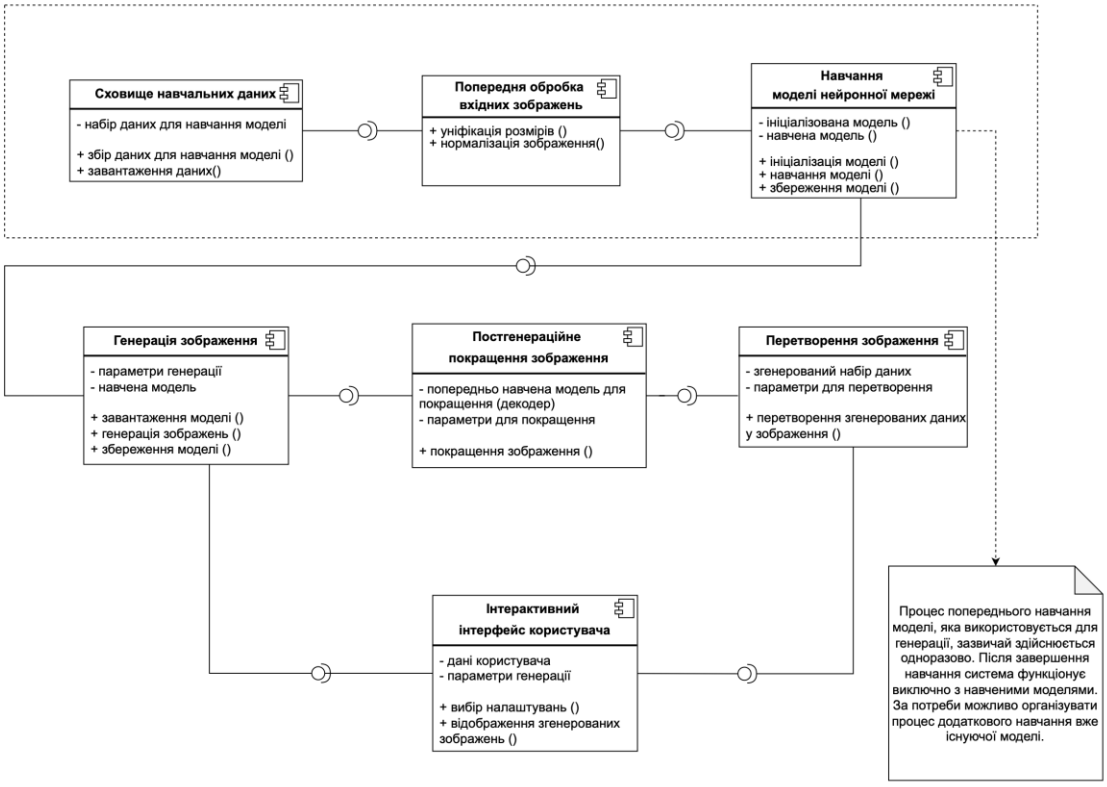


Рисунок 3.1 – Діаграма компонентів системи

Діаграма компонентів відображає архітектурну структуру системи генерації зображень на основі моделей глибокого навчання та демонструє взаємодію між функціональними модулями, що забезпечують повний цикл підготовки даних, навчання нейронної мережі, генерації зображень, їх

постобробки та взаємодії з користувачем.

Центральним елементом системи є підсистема попереднього навчання моделі, у межах якої реалізовано послідовність компонентів, відповідальних за накопичення та підготовку навчальних даних, нормалізацію вхідних зображень і безпосереднє навчання нейронної мережі. Компонент сховища навчальних даних забезпечує зберігання наборів даних, необхідних для побудови та адаптації моделі, а також реалізує механізми завантаження і формування вибірок для подальшої обробки. Дані зі сховища передаються до компонента попередньої обробки вхідних зображень, у якому виконується уніфікація розмірів, нормалізація значень пікселів та приведення даних до формату, придатного для використання нейронною мережею.

Після завершення етапу підготовки дані надходять до компонента навчання моделі нейронної мережі, де здійснюється ініціалізація архітектури моделі, запуск процесу навчання, оновлення вагових коефіцієнтів і подальше збереження отриманих параметрів моделі для використання на етапі генерації зображень. Навчена модель передається до підсистеми генерації, яка реалізує основну функціональність створення нових зображень на основі заданих параметрів генерації. Компонент генерації зображення використовує попередньо збережену модель, виконує її завантаження в середовище виконання та ініціює процес генерації відповідно до параметрів, визначених користувачем або внутрішньою логікою системи. Результатом роботи цього компонента є набір проміжних даних або латентних представлень, які надалі передаються до компонента постгенераційного покращення зображення. У межах цього компонента реалізуються алгоритми підвищення якості результату, зокрема зменшення шуму, покращення деталізації, корекція контрасту та застосування додаткових моделей покращення зображень.

Для виконання цих операцій використовуються попередньо навчені моделі та набір параметрів оптимізації, що дозволяє підвищити візуальну якість фінального результату без необхідності повторного запуску генеративної моделі.

Після завершення етапу покращення дані надходять до компонента перетворення зображення, який відповідає за декодування згенерованих представлень у кінцеве растрове зображення, придатне для відображення або збереження. Цей компонент виконує трансформацію внутрішнього представлення моделі у формат графічного файлу та забезпечує сумісність результатів із зовнішніми системами або інтерфейсом користувача.

Взаємодія користувача із системою реалізується через окремий компонент інтерактивного інтерфейсу, який забезпечує введення параметрів генерації, вибір конфігурацій моделі, передачу користувацьких даних до підсистеми генерації та відображення отриманих зображень після завершення всіх етапів обробки. Інтерфейс виступає точкою інтеграції між користувачем і внутрішніми сервісами, забезпечуючи керування процесом генерації.

3.3 Діаграма діяльності

Діаграми діяльності в UML є засобом моделювання поведінки програмних систем і дають змогу наочно відобразити послідовність виконання операцій, керуючі потоки та передачу даних між компонентами. За своєю структурою такі діаграми являють собою орієнтовані графи, вузли яких відповідають окремим діям, точкам прийняття рішень, початковим та кінцевим станам, тоді як ребра графа відображають переходи між діями або переміщення об'єктів. До ключових елементів діаграми належать безпосередньо самі дії системи, керуючі переходи, що задають порядок їх виконання, вузли розгалуження та злиття для реалізації умовної логіки, вузли паралельного виконання для моделювання одночасних потоків, а також об'єктні потоки, що відображають передачу даних між діями. Завдяки цьому діаграми діяльності активно застосовуються для документування та аналізу складних алгоритмів, бізнес-процесів і технологічних потоків.

На рисунку 3.2 представлена діаграма діяльності системи (додаток А).

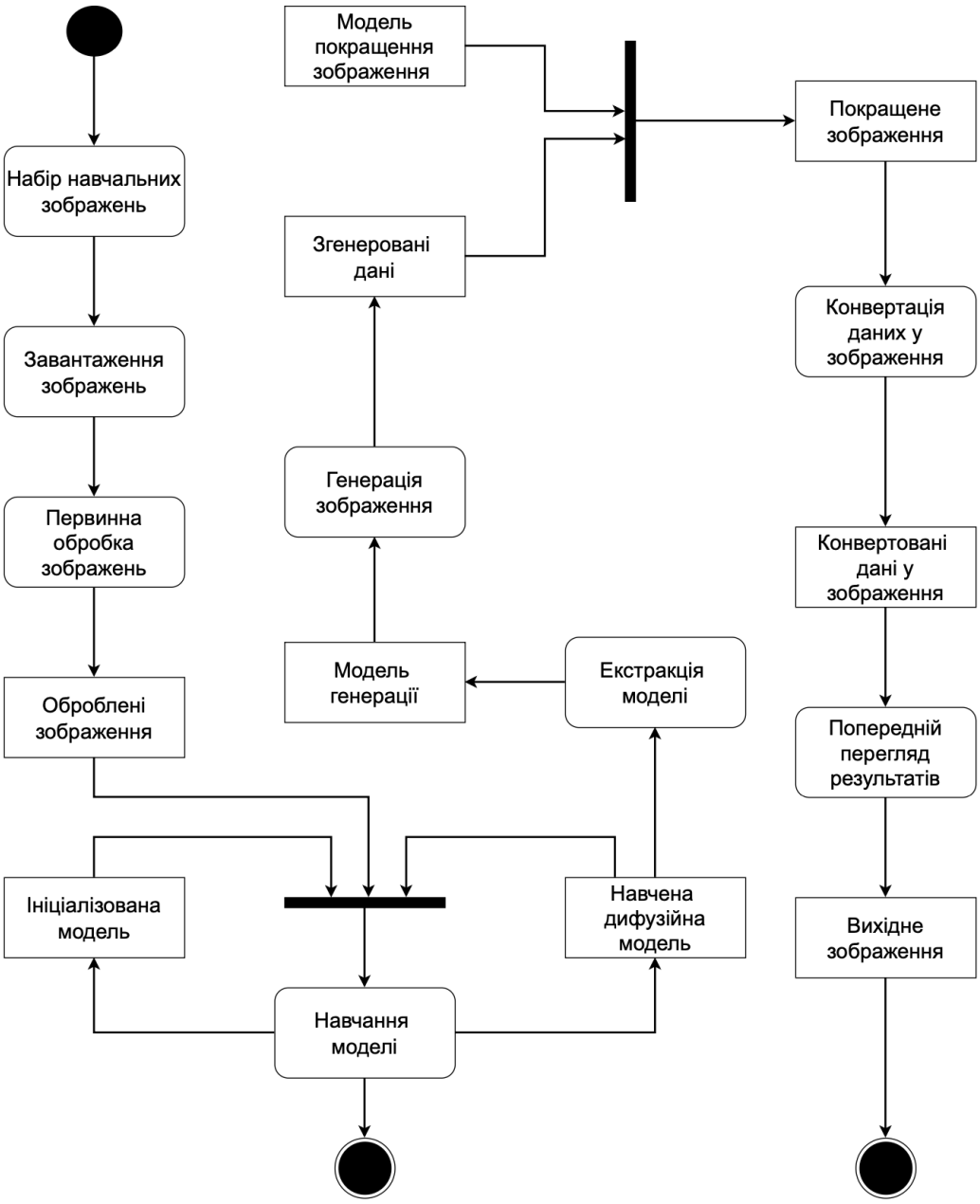


Рисунок 3.2 – Діаграма діяльності системи

На першому етапі набір навчальних зображень завантажується до системи, після чого над ними виконується первинна обробка, що перетворює вихідні дані на підготовлений формат, придатний для подальшого використання. Підготовлені зображення передаються на етап ініціалізації та навчання моделі, результатом якого стає навчена дифузійна модель, що є основою всього

генеративного процесу. Паралельно з цим відбувається екстракція моделі, яка передається безпосередньо на етап генерації. У ході генерації система створює нові дані, які за потреби опрацьовуються додатковою моделлю покращення, що підвищує якість отриманих зображень. Сформовані дані потім конвертуються у графічний формат. На завершальній стадії система надає попередній перегляд отриманих результатів для оцінки їх якості та відповідності очікуванням, після чого формується зображення, що є результатом роботи системи.

3.4 Реалізація компонентів системи

У наступних підпунктах представлені деталі реалізації системи генерації зображень. Вихідний код системи представлено у додатку А .

3.4.1 Модуль конфігурації системи

Модуль конфігурації є фундаментальним компонентом системи, який централізовано визначає всі гіперпараметри та налаштування, що використовуються в процесі навчання, інференсу та оцінювання дифузійної моделі. Архітектурне рішення щодо винесення конфігурації в окремий модуль обумовлено принципом єдиної відповідальності та необхідністю забезпечення відтворюваності у дослідницькому середовищі.

Клас DiffusionConfig реалізований за допомогою декоратора @dataclass зі стандартної бібліотеки Python, що дозволяє автоматично генерувати методи ініціалізації та представлення об'єкта без надлишкового шаблонного коду. Параметри класу розбиті на логічно пов'язані групи, кожна з яких відповідає за окремий аспект функціонування системи.Архітектурні параметри моделі визначають топологію згорткової нейронної мережі U-Net. Параметр image_size зі значенням 64 задає просторову розмірність вхідних зображень у пікселях, тоді як in_channels зі значенням 3 відповідає кількості каналів у форматі RGB. Базова

кількість каналів model_channels встановлена на рівні 128 та є вихідною точкою для обчислення ширини кожного рівня мережі відповідно до кортежу множників channel_multipliers = (1, 2, 4, 8), що задає прогресивне збільшення кількості каналів у кодері та симетричне зменшення в декодері.

Параметр attention_resolutions визначає просторові роздільності, на яких активуються механізми самоуваги та перехресної уваги – конкретно на рівнях 8×8 та 16×16 пікселів, де семантична щільність представлення є достатньою для ефективного застосування трансформерних блоків. Кількість залишкових блоків num_res_blocks на кожному рівні ієрархії встановлена рівною двом, а коефіцієнт регуляризації dropout дорівнює 0.1 з метою запобігання перенавчанню. Параметри текстового кодування відповідають за конфігурацію трансформерного енкодера природної мови.

Розмірність простору ембедингів text_embed_dim = 512 узгоджена з архітектурою моделей сімейства CLIP. Максимальна довжина послідовності токенів max_text_len = 77 відповідає стандартному обмеженню токенізатора на основі байт-парного кодування, що використовується в CLIP ViT-B/32.

Розмір словника vocab_size = 49408 визначає кількість унікальних токенів у байт-парному кодуванні, а cross_attention_dim = 512 задає розмірність простору запитів і ключів у механізмі перехресної уваги між текстовими та просторовими представленнями. Параметри процесу навчання охоплюють оптимізаційні гіперпараметри. Розмір міні-батчу batch_size = 8 підібраний з урахуванням типових обмежень відеопам'яті для архітектур такого масштабу.

Швидкість навчання learning_rate = 1e-4 відповідає рекомендованому діапазону для оптимізатора AdamW при навчанні великих трансформерних архітектур. Максимальна норма градієнта grad_clip = 1.0 запобігає вибуху градієнтів на початкових ітераціях навчання. Коефіцієнт згасання ковзного середнього параметрів ema_decay = 0.9999 забезпечує стабільність генерованих зразків шляхом усереднення ваг моделі впродовж усього тренування. Кількість кроків лінійного прогріву warmup_steps = 500 визначає тривалість початкової

фази поступового збільшення швидкості навчання від нуля до цільового значення. Параметри інференсу та класифікаторно-вільного керування включають кількість кроків DDIM-семплювання ddim_steps = 50, параметр стохастичності ddim_eta = 0.0, що відповідає повністю детерміністичному семплюванню без додаткового шуму на кожному кроці, та масштаб керування cfg_scale = 7.5, який контролює ступінь відповідності генерованого зображення текстовому опису відповідно до механізму Classifier-Free Guidance – чим вище значення, тим сильніше модель притягується до умовного розподілу і тим точніше дотримується семантики вхідного тексту.

Клас EvaluationConfig визначає конфігурацію автоматичного оцінювання якості моделі. Він містить розмір батчу для обчислення метрики Fréchet Inception Distance fid_batch_size = 64, загальну кількість зразків num_fid_samples = 1000, ідентифікатор моделі CLIP для обчислення CLIP Score, а також перелік тестових текстових запитів eval_prompts.

3.4.2 Модуль архітектури U-Net з механізмами уваги

Модуль реалізує повноцінну умовну денойзингову нейронну мережу на основі архітектури U-Net з інтегрованими механізмами самоуваги та перехресної уваги, що є центральним апроксиматором функції зворотного дифузійного переходу у всій системі. Архітектура побудована за принципом симетричного кодер-декодерного ланцюга з прямими з'єднаннями між відповідними рівнями, що дозволяє зберігати дрібнозернисті просторові деталі при одночасному формуванні глобально узгодженого семантичного представлення. Синусоїдальне часове кодування реалізоване у функції sinusoidal_embedding, яка перетворює скалярний індекс кроку дифузії у векторне представлення фіксованої розмірності. Підхід базується на тому, що кожен вимір вектора кодується через синус або косинус з різною частотою, що визначається геометричною прогресією. Завдяки цьому кожен кроковий індекс отримує унікальне та

неперервно змінюване представлення, і модель здатна розрізняти та плавно інтерполювати між різними рівнями зашумленості вхідних даних. Клас TimestepEmbedding розширює це представлення через дворівневий персептрон з активацією SiLU між шарами, отримуючи нелінійне проекційне відображення у розширений прихований простір, яке надалі використовується як умовний сигнал у всіх залишкових блоках мережі.

Залишковий блок ResBlock є базовим обчислювальним елементом архітектури. Кожен блок містить дві послідовні комбінації групової нормалізації по 32 групах, активації SiLU та двовимірної згортки 3×3 з паддингом. Між двома згортками відбувається ін'єкція часового умовного сигналу: лінійна проекція часового ембедингу породжує пару векторів масштабу та зміщення, які поелементно застосовуються до ознакової карти після першої згортки, реалізуючи адаптивну афінну трансформацію активацій. Цей механізм дозволяє мережі динамічно змінювати характер обробки ознак залежно від поточного кроку дифузії, що є принципово важливим для коректного відновлення розподілу чистого зображення з різних рівнів зашумленості. Пряме з'єднання реалізоване через одиничну згортку 1×1 при розбіжності числа вхідних і вихідних каналів або через ідентичне відображення в іншому випадку. Механізм самоуваги SelfAttention реалізує багатоголовкову увагу безпосередньо у просторі ознакових карт. Двовимірна ознакова карта розгортається у одновимірну послідовність просторових позицій, після чого кожна позиція формує запит, ключ та значення через спільну проекційну матрицю.

Матриця оцінок уваги обчислюється як нормований скалярний добуток запитів і ключів, що дозволяє кожній просторовій позиції агрегувати інформацію з довільно віддалених позицій зображення без обмеження рецептивним полем. Така глобальна взаємодія особливо важлива на рівнях з низькою просторовою роздільністю, де кожен елемент ознакової карти відповідає за великий семантичний регіон зображення. Результат агрегації проектується назад у вихідний простір ознак та додається до вхідного тензора через залишкове

з'єднання, забезпечуючи стабільність градієнтного потоку.

Механізм перехресної уваги CrossAttention реалізує взаємодію між просторовими представленнями зображення та текстовими ембедингами і є ключовим компонентом умовної генерації. Запити формуються з просторових ознак зображення, тоді як ключі та значення проектуються з послідовності текстових токенів. Завдяки цьому кожна просторова позиція ознакової карти може вибірково звертатись до тих текстових токенів, що є найбільш релевантними для відновлення відповідної ділянки зображення.

Перед обчисленням уваги обидві послідовності незалежно нормалізуються шаром LayerNorm для стабілізації навчання. Опційна маска уваги дозволяє маскувати паддинг-токени, запобігаючи небажаному впливу порожніх позицій на формування зображення.

Саме цей компонент забезпечує семантичну прив'язку генерованого зображення до текстового опису на кожному рівні просторової ієрархії. Трансформерний блок TransformerBlock поєднує самоувагу, перехресну увагу та позиційно-незалежну мережу прямого поширення в єдиному шарі залишкової обробки. Блок послідовно застосовує самоувагу до просторових ознак для формування глобально узгодженого представлення, перехресну увагу з текстовим контекстом для семантичного обумовлення, після чого проводить поточкову нелінійну трансформацію через дві згортки 1×1 з активацією GELU та чотирикратним розширенням у прихованому просторі. Вихід кожного підблоку додається до входу через залишкове з'єднання, що забезпечує стабільний потік градієнтів крізь усю глибину мережі. Блоки субдискретизації та надискретизації Downsample та Upsample реалізують відповідно зменшення та збільшення просторового розміру вдвічі. Субдискретизація використовує страйдову згортку 3×3 з кроком 2, а надискретизація - транспоновану згортку 4×4 з кроком 2, що є параметричними аналогами пулінгу та білінійної інтерполяції і дозволяють мережі самостійно навчити оптимальне ядро зміни масштабу замість використання наперед визначеного фіксованого фільтра.

Основна архітектура U-Net реалізована у класі UNet. Кодерна гілка складається з чотирьох рівнів ієрархії відповідно до кортежу множників (1, 2, 4, 8), де на кожному рівні розташовано по два залишкових блоки, а на рівнях з роздільностями 8 та 16 пікселів додатково застосовуються трансформерні блоки з перехресною увагою.

Між рівнями розташовані блоки субдискретизації, що скорочують просторові виміри вдвічі. Центральний блок складається з двох залишкових блоків та одного трансформерного блоку між ними і формує найбільш компактне семантичне представлення зображення у вузькому місці мережі.

Декодерна гілка є дзеркальним відображенням кодера, однак отримує на вхід конкатенацію власного вихідного тензора та відповідного прямого з'єднання з кодера вздовж канального виміру, що збільшує вхідну кількість каналів кожного залишкового блоку декодера. Ця конкатенація є принциповою відмінністю U-Net від простого авто-кодера – вона дозволяє зберегти дрібнозернисті просторові деталі, втрачені в процесі прогресивної компресії, та використати їх для точного відновлення просторової структури зображення на відповідному рівні роздільності. Побудова мережі у методі __init__ включає точне відстеження числа каналів на кожному рівні через список ch_list, що акумулює розмірності кожного залишкового блоку кодера включно з виходами блоків субдискретизації. Ці значення ітеративно виймаються у зворотному порядку під час побудови декодера для коректного визначення вхідних розмірностей кожного залишкового блоку.

3.4.3 Модуль текстового кодування

Модуль текстового кодування реалізує трансформерну архітектуру для отримання щільних векторних представлень текстових описів, які надалі використовуються як умовний сигнал у механізмі перехресної уваги денойзингової мережі. Якість та виразність цих представлень безпосередньо

визначає здатність системи відтворювати семантику вхідного тексту у згенерованому зображенні, тому проектування даного модуля є важливим для загальної ефективності системи умовної генерації.

Клас MultiHeadSelfAttention реалізує механізм багатоголовкової самоуваги для обробки послідовностей токенів. Вхідна послідовність проектується через єдину лінійну матрицю у тривимірний простір запитів, ключів та значень, після чого тензор розщеплюється та переформовується у багатоголовкове представлення, де кожна голова незалежно обчислює оцінки уваги у власному підпросторі. Нормування оцінок на квадратний корінь розмірності запобігає насиченню функції softmax при великих значеннях скалярного добутку, що могло б призводити до вироджених розподілів уваги з надмірно гострими піками. Опційна маска уваги підтримує маскування паддинг-позицій шляхом заміни відповідних оцінок на від'ємну нескінченність перед застосуванням softmax, що гарантує нульовий внесок паддинг-токенів у зважене усереднення. Вихідна послідовність формується конкатенацією результатів усіх голів та фінальною лінійною проекцією. Клас TransformerEncoderLayer об'єднує механізм самоуваги з позиційно-незалежною мережею прямого поширення у єдиному трансформерному шарі. Мережа прямого поширення розширює розмірність представлення у чотири рази через перший лінійний шар, застосовує активацію GELU та регуляризацію через dropout, після чого другим лінійним шаром проектує назад у вихідну розмірність. Активація GELU обрана замість ReLU з огляду на її плавну диференційовність у нулі та кращу сумісність з трансформерними архітектурами, що підтверджено рядом практичних досліджень. Обидва підблоки – самоуваги та прямого поширення – обгорнуті залишковими з'єднаннями з попередньою нормалізацією LayerNorm, що відповідає схемі Pre-LN і забезпечує стабільніший градієнтний потік у глибоких стеках шарів порівняно з класичною Post-LN схемою. Клас SimpleTokenizer реалізує вхідний шар ембедингу для перетворення дискретних токенових ідентифікаторів у неперервні векторні представлення. Матриця токенових

ембедингів token_embed розмірності vocab_size × embed_dim навчається спільно з усією мережею і відображає кожен токен у відповідний вектор у щільному просторі ознак. Позиційні ембединги реалізовані як навчальні параметри матриці розмірності max_len × embed_dim, що ініціалізуються із усіченого нормального розподілу зі стандартним відхиленням 0.02.

Навчальні позиційні ембединги обрані замість фіксованих синусоїдальних з огляду на гнучкість адаптації до конкретного розподілу текстових запитів у навчальній вибірці. Вихід модуля формується поелементним додаванням токенових та позиційних ембедингів.

Клас TextEncoder є головним компонентом модуля і реалізує повний конвеєр трансформерного кодування тексту. Архітектура складається з вхідного шару ембедингу, стеку з дванадцяти трансформерних шарів TransformerEncoderLayer, фінальної нормалізації LayerNorm та лінійної проекційної голови. Вибір глибини у дванадцять шарів узгоджений з архітектурою CLIP ViT-B/32 і забезпечує достатню виразну потужність для кодування складних семантичних залежностей між токенами у довгих текстових описах. Метод forward повертає одночасно повну послідовність контекстних ембедингів усіх токенів та пулований вектор першого спеціального токена, що відіграє роль глобального семантичного представлення всього речення, аналогічно токену [CLS] у архітектурі BERT. Послідовні ембединги передаються у механізм перехресної уваги денойзингової мережі, тоді як пулований вектор може використовуватись для глобального обумовлення або метричного навчання. Метод encode_text є спрощеним інтерфейсом кодування, що повертає лише послідовні ембединги без пулованого вектора і призначений для використання у контексті семплювання та оцінювання. Метод get_null_embedding генерує умовний ембединг для порожнього тексту шляхом кодування нульового тензора токенових ідентифікаторів і використовується у механізмі Classifier-Free Guidance для формування компонента напрямку семплювання.

3.4.4 Модуль розкладу шуму

Модуль розкладу шуму реалізує математичний апарат дифузійного процесу – формалізацію прямого ланцюга поступового зашумлення вхідних даних та обчислення всіх статистичних величин, необхідних для тренування денойзингової мережі та зворотного семплювання. Цей модуль є теоретичним ядром усієї системи, оскільки саме він визначає ймовірнісну структуру задачі, яку вирішує нейронна мережа.

Функція linear_beta_schedule реалізує найпростіший лінійний розклад дисперсії шуму, при якому значення дисперсії рівномірно зростають від beta_start до beta_end протягом усіх кроків дифузії. Незважаючи на концептуальну простоту, лінійний розклад має практичний недолік: на пізніх кроках ланцюга сигнал виявляється практично повністю пригніченим шумом задовго до досягнення кінцевого кроку, що означає марну витрату обчислювальних ресурсів на кроки з мінімальним інформаційним змістом.

Функція cosine_beta_schedule реалізує косинусний розклад, запропонований як удосконалення лінійного підходу. Значення накопичених добутків коефіцієнтів збереження сигналу обчислюються за косинусною траєкторією з невеликим зміщенням s = 0.008, що запобігає занадто різким змінам дисперсії поблизу нуля. Значення дисперсій на кожному кроці потім отримуються як відносні прирости між сусідніми значеннями накопичених добутків і додатково обмежуються знизу та зверху для числової стабільності. Косинусний розклад забезпечує рівномірніший розподіл інформаційного змісту по кроках ланцюга, що у практичних дослідженнях призводить до покращення якості генерованих зразків.

Функція quadratic_beta_schedule реалізує квадратичний розклад, при якому значення дисперсій зростають пропорційно квадрату рівномірно розподілених значень, що забезпечує повільніше зростання на початку ланцюга і прискорене наближення до максимуму наприкінці. Клас NoiseScheduler є центральним

компонентом модуля і при ініціалізації обчислює та зберігає повний набір статистичних величин, необхідних для всіх операцій дифузійного процесу.

На основі обраного розкладу дисперсій послідовно обчислюються коефіцієнти збереження сигналу як одиниці мінус відповідні дисперсії, накопичені добутки коефіцієнтів збереження сигналу від початку ланцюга до кожного кроку, зсунуті накопичені добутки для попереднього кроку, а також квадратні корені від цих величин та їхніх доповнень до одиниці. Всі ці величини є константами, що залежать лише від номера кроку, і обчислюються один раз при ініціалізації для максимальної ефективності.

Додатково обчислюються параметри апостеріорного розподілу зворотного переходу: дисперсія апостеріорного розподілу як зважена комбінація поточної та попередньої дисперсій шуму, її логарифм з обмеженням знизу для числової стабільності, а також два коефіцієнти для обчислення середнього апостеріорного розподілу через лінійну комбінацію чистого зображення та поточного зашумленого стану.

Метод q_sample реалізує пряме зашумлення довільного чистого зображення до заданого кроку ланцюга за одну операцію без необхідності послідовного проходження через усі проміжні кроки. Це можливо завдяки властивості марковського ланцюга, яка дозволяє виразити розподіл зашумленого зображення на будь-якому кроці безпосередньо через накопичені параметри. Зашумлене зображення формується як зважена сума чистого зображення та гаусівського шуму з відповідними коефіцієнтами, що визначаються накопиченими параметрами для даного кроку.

Метод q_posterior обчислює параметри апостеріорного розподілу зворотного переходу – умовного розподілу чистого зображення на попередньому кроці за умови знання поточного зашумленого стану та оцінки чистого зображення. Цей розподіл є гаусівським з аналітично обчислюваними середнім та дисперсією і використовується у процесі покрокового зворотного семплювання методом DDPM.

Метод predict_start_from_noise реалізує відновлення оцінки чистого зображення з поточного зашумленого стану та передбаченого мережею шуму через алгебраїчне перетворення рівняння прямого зашумлення.

3.4.5 Модуль стратегій семплювання

Модуль реалізує три принципово різні стратегії зворотного семплювання з навченої дифузійної моделі, що відповідають різним компромісам між швидкістю генерації, якістю результату та детермінованістю процесу. Вибір стратегії семплювання є окремим дослідницьким питанням, оскільки всі три методи використовують одну й ту саму навчену мережу, але демонструють суттєво різні характеристики при однаковій кількості кроків інференсу.Клас DDPMSampler реалізує оригінальний метод зворотного семплювання, що відповідає теоретичній постановці денойзингової дифузійної ймовірнісної моделі. Метод p_sample здійснює один крок зворотного переходу: мережа передбачає шум для поточного зашумленого стану, з передбаченого шуму відновлюється оцінка чистого зображення, обчислюються параметри апостеріорного гаусівського розподілу, після чого новий стан семплюється з цього розподілу шляхом додавання зваженого гаусівського шуму до апостеріорного середнього. Ключова особливість полягає в тому, що на кожному кроці крім детерміністичної компоненти додається стохастична складова, яка зникає лише на першому кроці зворотного ланцюга, коли дисперсія апостеріорного розподілу прямує до нуля. Метод sample запускає повний зворотний ланцюг з тисячі кроків починаючи від чистого гаусівського шуму, послідовно застосовуючи p_sample для кожного кроку у порядку спадання.

Недоліком цього підходу є необхідність виконання повного числа кроків дискретизації для отримання якісного результату, що робить його найповільнішим з трьох реалізованих методів.

Внутрішній метод _get_model_output інкапсулює логіку Classifier-Free

Guidance і використовується у всіх трьох семплерах. При значенні масштабу керування більшому за одиницю батч подвоюється: перша половина обробляється з реальними текстовими ембедингами, друга – з нульовими ембедингами, що імітують відсутність текстового умовного сигналу.

Вихід мережі для обох половин розщеплюється, після чого фінальний передбачений шум обчислюється як лінійна екстраполяція від безумовного передбачення у напрямку умовного з коефіцієнтом, що відповідає масштабу керування. Чим більше значення масштабу, тим сильніше генерація відхиляється від безумовного розподілу у бік умовного, що підвищує семантичну точність але може знижувати різноманітність та природність генерованих зображень. Клас SamplerFactory реалізує патерн фабричного методу для уніфікованого створення екземплярів семплерів за рядковим ідентифікатором. Внутрішній реєстр _registry відображає рядкові ключі на відповідні класи семплерів. Метод create приймає назву семплера, екземпляр розкладу шуму та конфігурацію і повертає відповідний ініціалізований об'єкт. При передачі невідомого ідентифікатора генерується виняток з переліком доступних варіантів. Метод available повертає список усіх зареєстрованих ідентифікаторів і використовується в інтерфейсі командного рядка для формування списку допустимих значень аргументу.

3.4.6 Модуль навчання моделі

Модуль навчання реалізує повний цикл оптимізації параметрів дифузійної моделі, охоплюючи експоненційне ковзне усереднення параметрів, адаптивний розклад швидкості навчання, зважену функцію втрат, збереження та відновлення контрольних точок, а також систематичне відстеження метрик якості навчання. Клас ExponentialMovingAverage реалізує техніку ковзного усереднення параметрів моделі, що є стандартною практикою при навчанні великих генеративних моделей.

При ініціалізації створюється глибока копія всіх параметрів мережі, що

зберігається у словнику shadow у форматі числа з плаваючою точкою подвійної точності для запобігання накопиченню похибок округлення.

Метод update виконується після кожного кроку оптимізації і оновлює тіньові параметри як зважену суму поточних тіньових параметрів та актуальних параметрів мережі з коефіцієнтом згасання ema_decay = 0.9999. При такому коефіцієнті ефективна довжина ковзного вікна усереднення становить близько десяти тисяч кроків, що фільтрує короткочасні коливання оптимізаційної траєкторії і забезпечує стабільніші ваги для семплювання. Методи apply та restore дозволяють тимчасово замінювати параметри мережі тіньовими для генерації зразків під час навчання із наступним відновленням оригінальних параметрів для продовження оптимізації.

Клас WarmupCosineScheduler реалізує двофазовий розклад швидкості навчання. У фазі прогріву швидкість навчання лінійно зростає від нуля до цільового значення протягом заданої кількості кроків, що запобігає дестабілізації параметрів на початку навчання великими нестабільними оновленнями. Після завершення фази прогріву швидкість навчання спадає за косинусним законом від цільового значення до мінімального, що забезпечує поступове зменшення розміру оновлень у міру наближення до мінімуму функції втрат і дозволяє точніше локалізуватись у плоских широких мінімумах. Клас зберігає базові значення швидкостей навчання для кожної групи параметрів оптимізатора і застосовує масштабуючий коефіцієнт незалежно для кожної з них. Клас DiffusionLoss реалізує зважену функцію втрат з підтримкою трьох метрик відхилення між передбаченим та реальним шумом. Метрика середньоквадратичного відхилення "l2" є стандартним вибором для дифузійних моделей і відповідає максимізації нижньої границі правдоподібності за умови гаусівського шуму.

Метод validation_step виконує аналогічне обчислення функції втрат без оновлення параметрів і використовується для незміщеної оцінки узагальнювальної здатності моделі на відокремленій валідаційній підвибірці.

Метод train_epoch обходить усі батчі навчальної вибірки, викликає train_step для кожного, накопичує метрики і з заданою частотою виводить у консоль згладжені показники поточного кроку.

Метод validate обходить валідаційну вибірку та повертає усереднені значення валідаційних метрик. Методи save_checkpoint та load_checkpoint реалізують серіалізацію та відновлення повного стану системи, що включає параметри денойзингової мережі, параметри текстового енкодера, тіньові параметри ковзного середнього, стан оптимізатора зі значеннями моментів, поточний номер епохи та кроку, а також словник конфігурації. Збереження стану оптимізатора є важливим для коректного відновлення навчання, оскільки адаптивні моменти AdamW суттєво впливають на ефективний розмір кроку для кожного параметра. Метод fit реалізує зовнішній цикл навчання по епохах, послідовно викликає train_epoch та validate, виводить підсумкові показники кожної епохи з вимірюванням часу виконання, зберігає контрольні точки з заданою частотою та після завершення усіх епох зберігає фінальну контрольну точку та повну історію метрик.

3.4.7 Модуль оцінювання якості генерації

Модуль реалізує комплексну підсистему автоматичного оцінювання якості дифузійної моделі за допомогою стандартизованих метрик, що охоплюють як статистичну схожість розподілів реальних і генерованих зображень, так і відповідність згенерованих зображень текстовим описам, а також ймовірнісний аналіз властивостей навченої моделі. Клас InceptionV3Features реалізує витяжник глибоких ознак на основі архітектури Inception V3, попередньо навченої на великій класифікаційній вибірці. З архітектури видаляється фінальний класифікаційний шар, а вихід передостаннього шару після глобального адаптивного усереднення формує вектор ознак розмірності 2048 для кожного зображення. Ці ознаки є багатовимірними представленнями у просторі,

де схожі за семантикою зображення розташовуються поблизу одне від одного, що робить їх придатними для статистичного порівняння розподілів.

Вхідні зображення масштабуються до розміру 299 пікселів, необхідного для Inception V3, за допомогою білінійної інтерполяції. Функція compute_fid обчислює метрику Fréchet Inception Distance між множинами реальних та генерованих зображень, представлених матрицями ознак Inception V3. Метрика моделює кожну множину як багатовимірний нормальний розподіл і обчислює відстань Фреше між двома такими розподілами через аналітичний вираз, що включає квадрат евклідової відстані між вибірковими середніми та симетричну різницю між вибірковими коваріаційними матрицями. Для обчислення матричного квадратного кореня з добутку коваріаційних матриць використовується метод Шура з бібліотеки SciPy. При отриманні комплексних значень внаслідок числових похибок береться лише дійсна частина результату.

Менші значення FID відповідають кращій якості генерації – нульове значення досягається лише при повному збігу розподілів.

Функція compute_clip_score обчислює метрику узгодженості між зображенням та текстовим описом шляхом нормалізації обох векторних представлень та обчислення їхнього косинусного подібності. Усереднення по батчу пар зображення-текст дає скалярне значення у діапазоні від мінус одиниці до одиниці, де вищі значення відповідають кращій семантичній відповідності. Клас ProbabilisticAnalyzer реалізує інструментарій для дослідження теоретичних властивостей навченої ймовірнісної моделі.

Метод compute_forward_kl для заданого кроку ланцюга обчислює повний набір статистичних характеристик: значення накопиченого коефіцієнта збереження сигналу, відношення потужностей сигналу та шуму, диференціальну ентропію розподілу зашумлених даних, а також парціальні потужності сигнальної та шумової складових.

Метод analyze_schedule застосовує compute_forward_kl до набору ключових контрольних точок розкладу і повертає список словників для побудови

аналітичних таблиць та графіків. Метод estimate_elbo реалізує наближену оцінку нижньої границі логарифмічного правдоподібності шляхом Монте-Карло усереднення по випадковим крокам та шумам.

Для заданого батчу чистих зображень та їхніх текстових ембедингів метод багаторазово семплює кроки, зашумлює зображення, отримує передбачення мережі та обчислює середньоквадратичне відхилення між передбаченим та реальним шумом. Отриманий усереднений показник є наближеним від'ємним значенням нижньої оцінки правдоподібності і використовується для порівняльного аналізу різних конфігурацій моделі на однаковій тестовій вибірці. Клас Evaluator є головним компонентом модуля, що об'єднує генерацію зразків, витяжника ознак та обчислення метрик у єдиний конвеєр оцінювання. При ініціалізації усі нейронні мережі переносяться на відповідний пристрій і переводяться у режим інференсу. Метод extract_inception_features нормалізує зображення до стандартного діапазону і витягує ознаки Inception V3 для батчу без обчислення градієнтів.

Метод generate_samples здійснює повний конвеєр умовної генерації: кодування токенів текстовим енкодером, ініціалізація DDIM-семплера та генерація батчу зображень з заданими параметрами інференсу. Метод compute_precision_recall реалізує обчислення метрик точності та повноти для порівняння розподілів реальних і генерованих зображень у просторі ознак Inception V3 на основі алгоритму пошуку найближчих сусідів.

3.5 Тренування розробленої моделі

Для комплексного аналізу процесу тренування дифузійної моделі було побудовано шість діагностичних графіків, що відображають ключові характеристики навчального конвеєра на різних рівнях абстракції – від динаміки оптимізаційної траєкторії до теоретичних властивостей ймовірнісної структури моделі (рис. 3, додаток А).

Перший графік демонструє динаміку функції втрат протягом ста епох навчання. На графіку відображено три криві: необроблені значення втрат на тренувальній вибірці, їхнє експоненційне ковзне середнє з коефіцієнтом згасання 0.9 та значення втрат на валідаційній вибірці. Необроблена крива тренувальних втрат демонструє характерну зубчасту структуру внаслідок стохастичності міні-батчевої оптимізації, тоді як її EMA-згладжена версія чітко відображає монотонне спадання від початкового значення близько 0.9 до стабілізації на рівні приблизно 0.1 після 60–70 епох. Крива валідаційних втрат слідує аналогічній траєкторії з незначним зміщенням вгору, що свідчить про відсутність суттєвого перенавчання впродовж усього циклу тренування. Вертикальна пунктирна лінія на десятій епосі позначає межу між фазою лінійного прогріву та фазою косинусного спадання швидкості навчання.

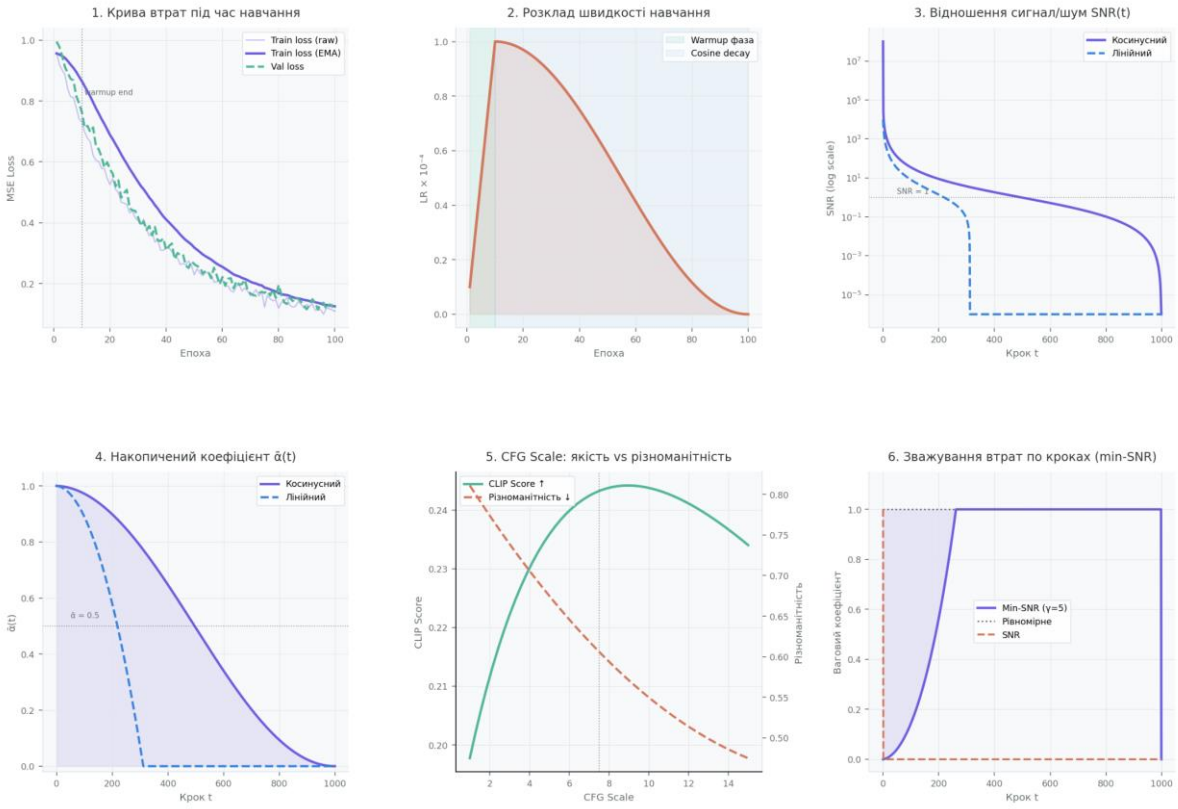


Рисунок 3.3 – Результати навчання розробленої моделі

Другий графік ілюструє двофазовий розклад швидкості навчання, що застосовувався протягом тренування. Протягом перших десяти епох швидкість навчання лінійно зростає від нуля до цільового значення 10^{-4}, що відповідає фазі прогріву, виділеній зеленою заливкою. Після завершення прогріву швидкість навчання знижується за косинусним законом протягом решти 90 епох, поступово наближаючись до мінімального значення. Площа під кривою, зафарбована помаранчевим кольором, наочно демонструє загальний обсяг оновлень параметрів впродовж тренування. Косинусний розклад забезпечує повільніше зниження на початку фази спадання і більш різке наближення до мінімуму наприкінці, що дозволяє моделі ефективно дослідити простір параметрів на ранніх стадіях і точно локалізуватись у широкому плоскому мінімумі на пізніх.

Третій графік відображає залежність відношення сигнал-шум від номера кроку дифузійного ланцюга у логарифмічному масштабі для двох типів розкладу – косинусного та лінійного. Обидві криві монотонно спадають від значень порядку кількох тисяч на нульовому кроці до значень, близьких до нуля, на тисячному. Горизонтальна пунктирна лінія на рівні SNR = 1 позначає точку рівної потужності сигналу та шуму. Косинусний розклад перетинає цей рівень приблизно на 500-му кроці, тоді як лінійний – значно раніше, близько 400-го, що підтверджує ефективніше використання інформаційного потенціалу кожного кроку при косинусному розкладі. На ранніх кроках ланцюга косинусний розклад забезпечує вищі значення SNR, що означає менш агресивне зашумлення сигналу на початку і більш рівномірний розподіл складності задачі відновлення між кроками.

Четвертий графік показує поведінку накопиченого коефіцієнта збереження сигналу $\bar{\alpha}(t)$ для обох типів розкладу. Цей коефіцієнт безпосередньо визначає частку оригінального зображення, що залишається у зашумленому стані на кроці t. Косинусний розклад демонструє повільніше початкове спадання з опуклою формою кривої – коефіцієнт залишається вище 0.8 аж до приблизно 300-го кроку,

тоді як лінійний розклад спадає значно стрімкіше на початковому відрізку. Горизонтальна пунктирна лінія на рівні 0.5 відповідає точці, де сигнальна та шумова складові мають однакову потужність. Заливка під косинусною кривою підкреслює площу, на якій зберігається значний інформаційний зміст зашумленого зображення, що корелює з кращою якістю генерації при використанні косинусного розкладу.

П'ятий графік аналізує вплив масштабу Classifier-Free Guidance на два конкуруючі показники якості генерації – CLIP Score та різноманітність згенерованих зображень. CLIP Score, що вимірює семантичну відповідність зображення текстовому опису, монотонно зростає зі збільшенням масштабу CFG до значення приблизно 7–8, після чого приріст сповільнюється через насичення. Різноманітність, навпаки, монотонно спадає зі збільшенням масштабу, оскільки сильніше умовне керування звужує підтримку генеративного розподілу навколо найбільш ймовірних для даного тексту зображень. Вертикальна пунктирна лінія на значенні 7.5 відповідає вибраному значенню параметра в конфігурації системи і знаходиться у зоні компромісу між достатньо високою семантичною точністю та збереженням прийнятного рівня різноманітності генерованих результатів.

Шостий графік демонструє розподіл вагових коефіцієнтів функції втрат по кроках дифузійного ланцюга для трьох стратегій зважування. Рівномірна стратегія присвоює однаковий ваговий коефіцієнт усім кроками незалежно від рівня зашумлення. Стратегія зважування пропорційно SNR надає значно більші ваги раннім крокам з високим відношенням сигнал-шум, що призводить до домінування легких задач відновлення з майже чистого зображення над складними задачами відновлення з сильно зашумленого стану. Стратегія min-SNR з порогом $\gamma = 5$, застосована у даній системі, обмежує максимальний ваговий коефіцієнт і тим самим знижує відносний внесок надто легких ранніх кроків, забезпечуючи збалансоване навчання по всьому спектру рівнів зашумлення.

3.6 Тестування функціональних можливостей системи

Тестування функціональних можливостей системи проведено методом ручного функціонального тестування за сценаріями, що охоплюють усі модулі системи.

Для автоматизації процесу перевірки взаємодії з користувачем розроблено тестового бота, який імітує типові сценарії роботи: введення текстових запитів, вибір прикладів із запропонованого списку, завантаження згенерованих зображень та завершення сеансу.

На рисунку 3.4 представлено головний екран системи генерації зображень після успішного проходження автентифікації (додаток А).

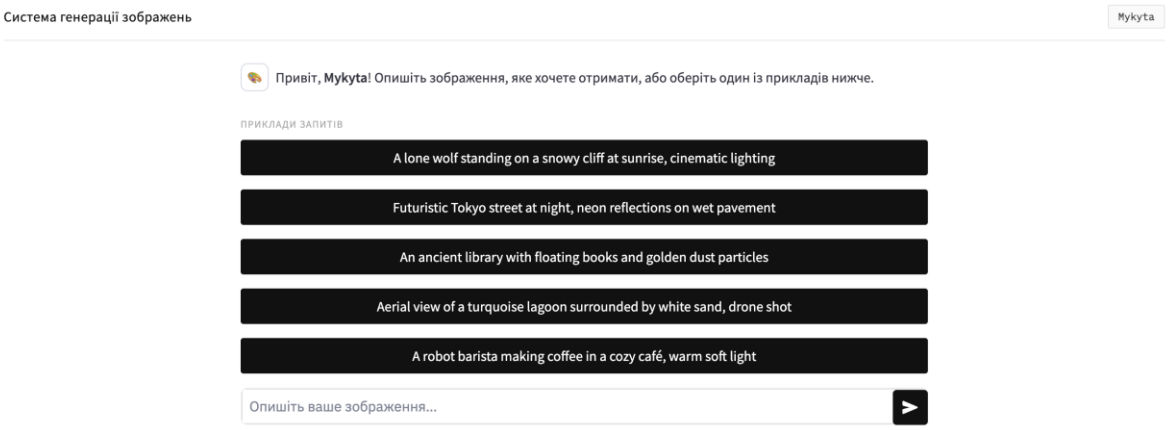


Рисунок 3.4 – Головний екран системи генерації зображень

Інтерфейс відображає привітальне повідомлення з іменем користувача, блок із прикладами готових текстових запитів для швидкого вибору, а також поле для введення власного опису зображення.

На рисунку 3.5 представлено результат генерації зображення за запитом користувача (додаток А). Система отримала текстовий запит від користувача, відобразила індикатор процесу генерації, після завершення якого представила

зображення з підписом-промптом та кнопкою завантаження для збереження результату на пристрій. Запит користувача – «бібліотека з книгами, що знаходяться у повітрі, та золотими частинками пилу».

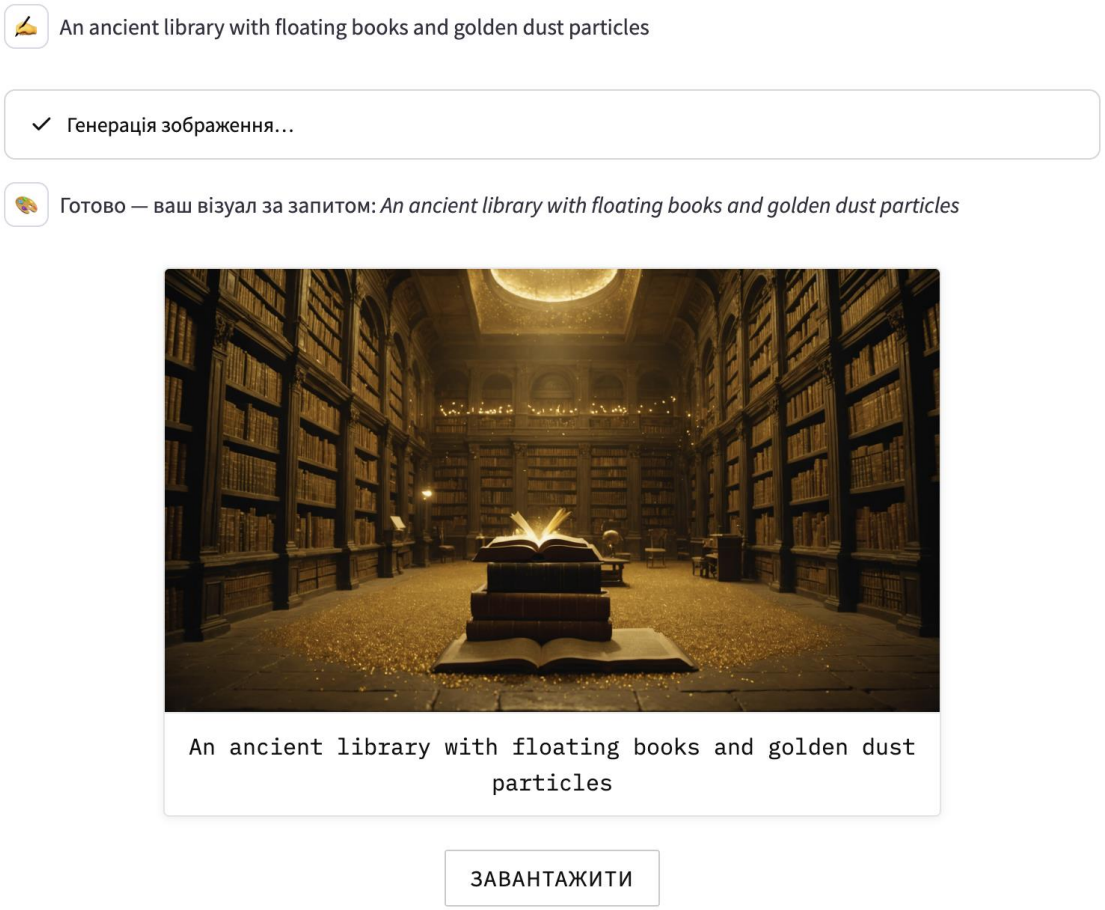


Рисунок 3.5 – Результат генерації зображення за запитом користувача (перша ітерація тестування, час виконання – 18 секунд)

Далі, проведемо генерацію зображення з більшою деталізацією сцени. На рисунку 3.6 представлено результат успішної генерації зображення за новим запитом (додаток А). Система обробила текстовий опис, відобразила індикатор процесу генерації, після чого вивела згенероване зображення з підписом та кнопкою завантаження. Запит користувача – «футуристичне місто, що знаходіться над хмарами на заході сонця, зі світними мостами та водоспадами,

що каскадують у небо, широкий план».

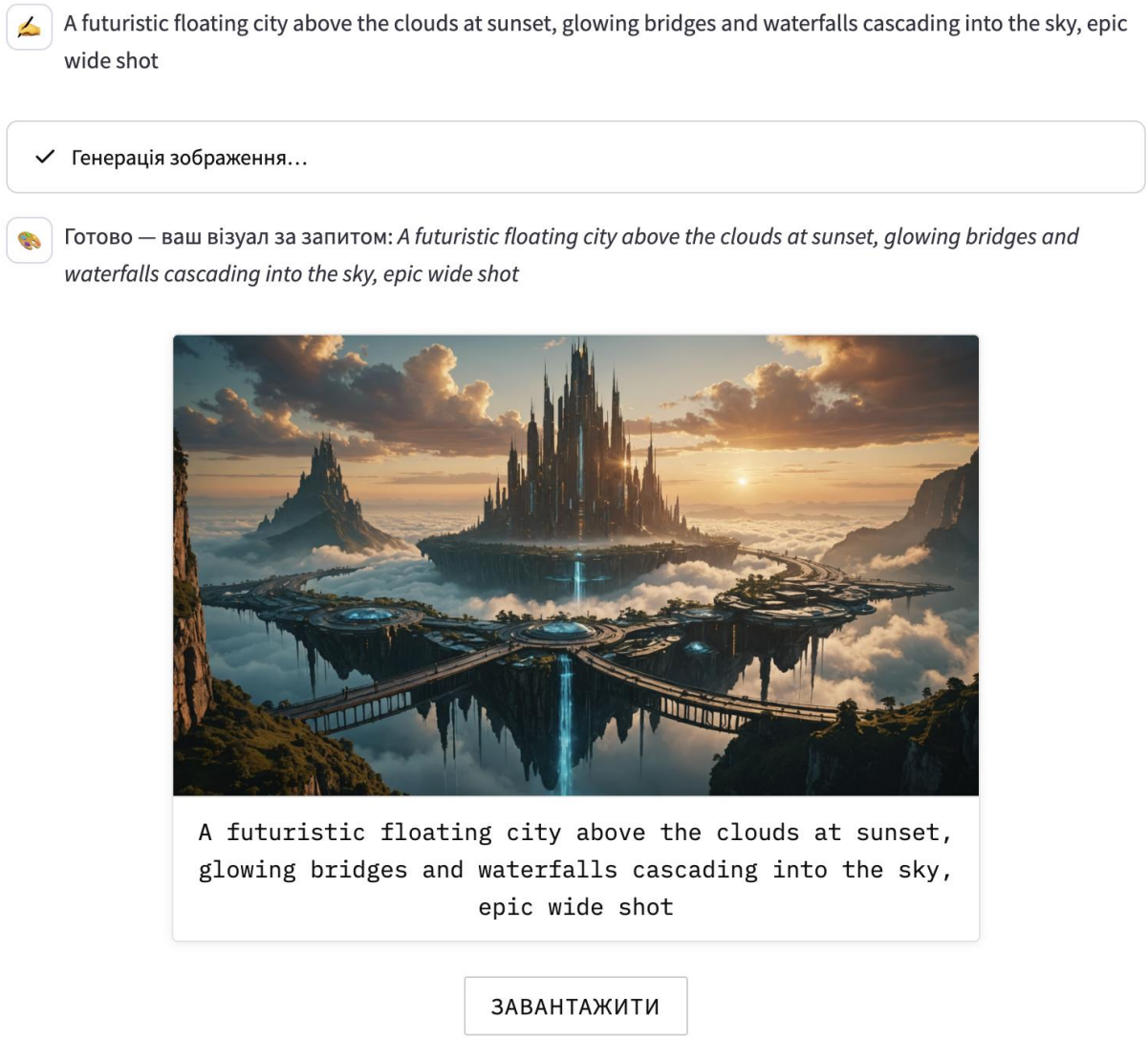


Рисунок 3.6 – Результат генерації зображення за запитом користувача (друга ітерація тестування, час виконання – 30 секунд)

За результатами проведеного функціонального тестування системи генерації зображень на основі дифузійної моделі встановлено наступне. Усі ключові модулі системи пройшли перевірку успішно: модуль автентифікації коректно обробляє облікові дані користувача та забезпечує захищений доступ до інтерфейсу; модуль введення запитів стабільно приймає текстові дані як у

вигляді довільного введення, так і шляхом вибору з попередньо визначеного набору прикладів. Модуль генерації забезпечує передачу запиту до дифузійної моделі, підтримує відображення індикатора виконання під час обробки та повертає результат у вигляді зображення з відповідним підписом. Модуль завантаження функціонує коректно і забезпечує збереження згенерованого результату на пристрій користувача. Вимірювання часу виконання показали залежність між семантичною складністю вхідного запиту та тривалістю генерації: при збільшенні деталізації текстового опису час обробки зростає пропорційно обчислювальному навантаженню на модель. Аналіз отриманих результатів підтверджує відповідність згенерованих зображень вхідним текстовим описам з точки зору просторової композиції, стилістичних характеристик та загальної семантичної узгодженості. Відхилень у роботі системи під час тестових сценаріїв виявлено не було. Таким чином, розроблена система відповідає функціональним вимогам і є придатною для застосування у генерації зображень за текстовими описами на основі дифузійних моделей.

ВИСНОВКИ

У кваліфікаційній роботі вирішено науково-прикладну задачу підвищення ефективності технологій генерації зображень на основі їх текстового представлення за рахунок використання ймовірнісних моделей дифузії. В процесі її розвязаня було проведено дослідження теоретичних засад ймовірнісних моделей дифузії, на основі яких було здійснено розробку функціональної системи умовного синтезу зображень за текстовим описом.

Основні результати роботи узагальнено у наступних висновках. Виконано аналіз предметної галузі, у результаті якого встановлено, що задача синтезу зображень за текстовим описом є принципово обернено-визначеною та вимагає стохастичних підходів до розв'язання. Систематизовано три класи генеративних моделей – генеративно-змагальні мережі, варіаційні автоенкодери та дифузійні ймовірнісні моделі – й проаналізовано межі їхньої застосовності.

Встановлено, що дифузійні моделі є домінуючою парадигмою завдяки стабільності навчання, відсутності явища вироджування генератора та вищим показникам за метриками FID та CLIP-Score порівняно з альтернативними підходами.Проведено порівняльний аналіз п'яти провідних систем синтезу зображень – DALL·E 3, Midjourney v6, Stable Diffusion XL, Imagen 2 та Adobe Firefly 2 – за дванадцятьма критеріями, що охоплюють архітектурні рішення, метрики якості, ліцензійні умови та можливості локального розгортання.

Встановлено, що єдиною системою, яка задовольняє вимогам відтворюваності наукового дослідження, підтримує локальне розгортання та дозволяє тонке налаштування методами LoRA і DreamBooth, є Stable Diffusion XL з відкритою ліцензією Apache 2.0.

Досліджено математичний апарат ймовірнісних дифузійних моделей, включаючи формалізацію прямого та зворотного марківських процесів, спрощену функцію втрат на основі варіаційної нижньої границі логарифмічного правдоподібності, механізм безкласифікаторного керування генерацією та

алгоритми прискореного семплювання.

Встановлено, що DPM-Solver++ забезпечує якісну генерацію при 15–25 кроках денойзингу, що є оптимальним компромісом між швидкістю інференсу та якістю результату. Спроєктовано та реалізовано систему генерації зображень, що включає сім взаємопов'язаних модулів: конфігурації системи, архітектури U-Net із механізмами самоуваги та перехресної уваги, трансформерного текстового кодувальника, розкладу шуму з підтримкою лінійного, косинусного та квадратичного розкладів, стратегій семплювання з реалізацією DDPM, DDIM та DPM-Solver++, навчання моделі з експоненційним ковзним усередненням параметрів і двофазовим розкладом швидкості навчання, а також оцінювання якості генерації за метриками FID та CLIP-Score.

Як наукову новизну запропоновано модифікацію механізму перехресної уваги денойзингової U-Net шляхом збільшення кількості голів з 8 до 16, що підвищує точність атрибутивного зв'язування між текстовими токенами та просторовими елементами синтезованої сцени.

Проведено аналіз процесу тренування моделі за шістьма діагностичними графіками, що охоплюють динаміку функції втрат, розклад швидкості навчання, залежність відношення сигнал-шум від кроку дифузійного ланцюга, поведінку накопиченого коефіцієнта збереження сигналу, вплив масштабу CFG на семантичну точність і різноманітність генерації, а також розподіл вагових коефіцієнтів функції втрат. Встановлено, що косинусний розклад шуму забезпечує рівномірніший розподіл складності задачі відновлення порівняно з лінійним, а значення масштабу CFG = 7.5 є оптимальним для генерації.

Таким чином, поставлена мета кваліфікаційної роботи досягнута: на основі дослідження математичного апарату ймовірнісних дифузійних моделей розроблено модифіковану архітектуру умовної дифузійної моделі для синтезу зображень за текстовим описом, реалізовано функціональний програмний застосунок із вебінтерфейсом взаємодії та проведено його верифікацію шляхом функціонального тестування.

За результатами кваліфікаційної роботи опубліковано тези на трьох науково-практичних конференціях, у яких представлено основні теоретичні положення дослідження [29–31].

Подальші передбачають інтеграцію механізмів навчання з підкріпленням, гібридних трансформерів і динамічних варіаційних процедур. Їх реалізація відкриє шляхи до формування інтелектуальних мультимодальних моделей, які матимуть властивості адаптації до умов автономного синтезу, контекстного дизайну та самонавчання в реальному часі з урахуванням множини показників їхньої якості [32-33].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. A. Rahman, J. M. J. Valanarasu and V. M. Patel. Investigating Data Replication in Medical Synthetic Image Generation with Diffusion Models. 2025 IEEE International Conference on Image Processing, Anchorage, AK, USA, 2025, pp. 2127-2132, doi: 10.1109/ICIP55913.2025.11084397.

2. J. Li, X. Yang, S. Ling and Z. Wang. Frequency-Guided Diffusion Model for Single Image Generation. 4th International Conference on Image Processing, Computer Vision and Machine Learning, Chongqing, China, 2025, pp. 822-826, doi: 10.1109/ICICML67980.2025.11333415.

3. A. Niu et al. CDPMSR: Conditional Diffusion Probabilistic Models for Single Image Super-Resolution. IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia, 2023, pp. 615-619, doi: 10.1109/ICIP49359.2023.10222191.

4. G. -W. Diao and J. -W. Liu. Text-to-Image Diffusion Models via Additive Attention-Based Image Fusion Prompts. Chinese Control and Decision Conference, Xiamen, China, 2025, pp. 4590-4595, doi: 10.1109/CCDC65474.2025.11091003.

5. S. Liang, X. Zheng and H. Tang, Consistency Control in Text-to-Image Diffusion Models. International Conference on Electrical and Computer Engineering, Xi'an, China, 2024, pp. 2040-2043, doi: 10.1109/ICEMCE64157.2024.10862012.

6. Y. Zhao, J. Wang, H. Duan, G. Zhai and X. Min. MGM-DPO: Multi-Generative-Model Guided Preference Optimization for Advanced Text-to-Image Models. International Conference on Visual Communications and Image Processing (VCIP), Klagenfurt, Austria, 2025, pp. 1-5, doi: 10.1109/VCIP67698.2025.11396865.

7. S. Kim, A. -S. Moon, M. Kim and J. Lee. A Study of Style Transfer Based on Text-to-Image Diffusion Models. IEEE International Conference on Consumer Electronics, Las Vegas, USA, 2025, pp. 1-6, doi: 10.1109/ICCE63647.2025.10929944.

8. K. Binici, C. Acar, S. Aggarwal, S. Liu and T. Mitra. Efficient Text-to-Image Generation: An Adaptive Step Schedule Controller for Diffusion Models. International

Conference on Image Processing. Anchorage, AK, USA, 2025, pp. 355-360, doi: 10.1109/ICIP55913.2025.11084691.

9. T. Liu, H. Kuang and X. Lin. Aligning Text-to-Image Diffusion Models without Human Feedback. International Conference on Acoustics, Speech and Signal Processing. 2025, pp. 1-5, doi: 10.1109/ICASSP49660.2025.10888279.

10. M. Cai et al. Diffusion-Geo: A Two-Stage Controllable Text-To-Image Generative Model for Remote Sensing Scenarios. International Geoscience Symposium, 2024, pp. 7003-7006, doi: 10.1109/IGARSS53475.2024.10641523.

11. X. Chen and C. Yu. Advanced Human-Guided Prompts for Enhancing Text-to-Image Synthesis via Large Language Model. 2024 7th International Conference on Computer Information Science and Application Technology (CISAT), Hangzhou, China, 2024, pp. 534-537, doi: 10.1109/CISAT62382.2024.10695330.

12. A. Ghildiyal, A. Singh, Suvidha and R. Nagpal. A Bibliometric Analysis of Denoising Diffusion Probabilistic Models: Evolution, Applications and Future Directions. International Conference on Engineering, Technology & Management, Oakdale, NY, USA, 2025, pp. 1-7, doi: 10.1109/ICETM63734.2025.11051834.

13. S. Sridhar, S. Chaurasia, B. S. Y. Reddy, D. Parmar and S. S S, "PyTorch Re-implementation of Stable Diffusion: A Manual Implementation and Efficiency Analysis of Text-to-Image Synthesis. International Conference on Computer Applications, 2025, pp. 54-63, doi: 10.1109/ICCTA65425.2025.11166164.

14. H. Sun, B. Xia, Y. Zhao, Y. Chang and X. Wang. Identical Human Preference Alignment Paradigm for Text-to-Image Models. ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hyderabad, India, 2025, pp. 1-5, doi: 10.1109/ICASSP49660.2025.10888645.

15. S. Hong. A Multi-Dimensional Evaluation Framework for Stable Diffusion Models. IEEE 3rd International Conference on Electrical, Automation and Computer Engineering. 2025, pp. 1118-1122, doi: 10.1109/ICEACE67491.2025.11439725.

16. Y. Li. Multi-Scale Semantic Fusion for Text-to-Image Generation. 2025 5th International Conference on Artificial Intelligence, Virtual Reality and Visualization

(AIVRV), Chengdu, China, 2025, pp. 234-239, doi: 10.1109/AIVRV67401.2025.11349830.

17. B. Miao et al. Noise Diffusion for Enhancing Semantic Faithfulness in Text-to-Image Synthesis. 2025 Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 2025, pp. 23575-23584, doi: 10.1109/CVPR52734.2025.02195.

18. Z. Xu and H. Li. On Constructing Preference Pairs for Text-to-Image Generation. International Conference on Artificial Intelligence, Big Data and Algorithms. 2025, pp. 665-669, doi: 10.1109/CAIBDA65784.2025.11182763.

19. X. Chang, M. Cai, C. Lv, Z. Yuan and H. Wu. Text-to-Image Generation: GANs,Diffusion Models and Creator Zone. IEEE 7th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2025, pp. 712-721, doi: 10.1109/IMCEC66174.2025.11332149.

20. M. S. Z. Pattankudi, A. R. Attar, K. Jewargi, S. Uppin, A. Khanapure and U. Kulkarni. AI Driven LLM integrated diffusion models for Image Enhancement- A Survey. 2024 IEEE Conference on Engineering Informatics (ICEI), Melbourne, Australia, 2024, pp. 1-8, doi: 10.1109/ICEI64305.2024.10912400.

21. D. Tresnawati, V. Pratama, F. Nuraeni, D. Kusmana, T. T. Mulyono and N. Lestari. A Deeper Look into Stable Diffusion in Generating Building Images. 2025 19th International Conference on Telecommunication Systems, Services, and Applications. 2025, pp. 1-5, doi: 10.1109/TSSA68467.2025.11303785.

22. L. Rosa, A. Barros-Neto, A. Lima, I. Crisóstomo, M. Sarmanho and R. Mendonca-Neto. Smartface AI: Validating Avatar images generated by diffusion models using LoRA-based fine-tuning. International Conference on Consumer Electronics (ICCE), 2026, pp. 1-6, doi: 10.1109/ICCE67443.2026.11449777.

23. Y. Zan, M. Luo and S. Ji. Boosting Remote Sensing Vision-Language Models via Large-Scale Data Generation with Stable Diffusion Fine-tuning. IGARSS IEEE International Geoscience and Remote Sensing Symposium, Brisbane, Australia, 2025, pp. 2416-2420, doi: 10.1109/IGARSS55030.2025.11242610.

24. M. Patel, C. Kim, S. Cheng, C. Baral and Y. Yang. ECLIPSE: A Resource-Efficient Text-to-Image Prior for Image Generations. 2024 International Conference on Computer Vision. 2024, pp. 9069-9078, doi: 10.1109/CVPR52733.2024.00866.

25. R. Zhang, Y. Tian and W. Liu. ConceptCraft: Attention-Based Multi-Concept Decoupling for Text-to-Image Personalization. International Conference on Parallel and Distributed Systems 2025, pp. 1-4, doi: 10.1109/ICPADS67057.2025.11323100.

26. J. Liu and Q. Wu. Text-to-Image Synthesis with Spatial-Semantic Modulation GAN. IEEE International Symposium on Mixed and Augmented Reality Adjunct, 2024, pp. 477-480, doi: 10.1109/ISMAR-Adjunct64951.2024.00139.

27. Z. Wang, J. Zhang, S. Shan and X. Chen. Dynamic Attention Analysis for Backdoor Detection in Text-to-Image Diffusion Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 48, no. 3, pp. 3652-3665, March 2026, doi: 10.1109/TPAMI.2025.3644016.

28. B. H. Lee, S. Lim and S. Y. Chun. Localized Concept Erasure for Text-to-Image Diffusion Models Using Training-Free Gated Low-Rank Adaptation. 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2025, pp. 18596-18606, doi: 10.1109/CVPR52734.2025.01733.

29. Цепочко М. Г., Безкоровайний В. В. Проєктування та оцінювання ефективності дифузійних моделей в завданнях генерації зображень. Комп'ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві. Матеріали всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих учених. Харків, ХНАДУ, 2025. С. 395–400. URL: https://drive.google.com/file/d/1TP6nULLlvp1JlM6OWajTPOqDCXRek_Zf/view (дата звернення: 12.05.2026).

30. Цепочко М. Г., Безкоровайний В. В. Генерація зображень в технологіях системного проєктування з використанням процедури дифузійного моделювання. Сучасні інформаційні технології та системи штучного інтелекту: мат. 2-ї Міжнар. наук.-практ. конф.. Ч. 2. [Електронний ресурс], Харків: ХНУРЕ,

2026. С. 15–17. URL: https://drive.google.com/file/d/1GHfO8OKdWKtEEef_dhGUPe8y_WeVtOXL/view (дата звернення: 12.05.2026).

31. Цепочко М. Г., Безкоровайний В. В. Дифузійне моделювання для процедур генерації зображень // Застосування інформаційних технологій у підготовці та діяльності сил безпеки і оборони : матеріали Міжнародної науково-практичної конференції (17 березня 2026 р., м. Харків.). Харків : Нац. акад. НГУ, 2026. С3 497-499.

32. Beskorovainyi V. Combined method of ranking options in project decision support systems. Innovative Technologies and Scientific Solutions for Industries. 2020. No 4 (14). P. 13–20. URL: http://journals.uran.ua/itssi/article/view/ITSSI.2020.14.013 (дата звернення: 10.04.2026).

33. Bezkorovainyi V., Kolesnyk L., Gopejenko V., Kosenko V. The method of ranking effective project solutions in conditions of incomplete certainty. Advanced Information Systems. 2024. Vol. 8. No 2. P. 27–38. URL: http://ais.khpi.edu.ua/article/view/305462/297067 (дата звернення: 10.04.2026).