ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«DIFFUSION-МОДЕЛІ У ГЕНЕРАЦІЇ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ»

на здобуття освітнього ступеня магістр
за спеціальності 124 Системний аналіз
(код,найменування спеціальності)
освітньо-професійної програми Інтелектуальні системи управління
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Гліб КОШКАРЬОВ
(підпис) (ім'я, ПРІЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
група САДМ-61 Гліб КОШКАРЬОВ
(ім'я, ПРІЗВИЩЕ)
Керівник Олексій ШУШУРА
д.т.н.
доцент
(ім'я, ПРІЗВИЩЕ)

Рецензент: _____
(ім'я, ПРІЗВИЩЕ)

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти магістр
Спеціальність 124 Системний аналіз
Освітньо-професійна програма Інтелектуальні системи управління

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ

Каміла СТОРЧАК _____

"____" _____2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Кошкарьову Глібу Олександровичу
(прізвище, ім'я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Diffusion-моделі у генерації мультимедійного контенту
керівник кваліфікаційної роботи: Олексій ШУШУРА д.т.н., доцент
(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від "30 " жовтня 2025 р. № 467

2. Строк подання кваліфікаційної роботи «26» грудня 2025 р.

3. Вихідні дані кваліфікаційної роботи:
1. Генерація контенту.
2. Архітектура дифузійних моделей.
3. Методи навчання дифузійних моделей.
4. Науково-технічна література

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1. Дослідження тенденцій розвитку та поширення Штучного інтелекту.
2. Огляд методів генерації мультимедійного контенту.
3. Аналіз результатів впровадження ШІ в робочу, наукову та суспільні сфери.

5.Перелік ілюстраційного матеріалу: презентація

6. Дата видачі завдання «30» жовтня 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	15.09-26.09.25	
2.	Дослідження тенденцій останніх моделей штучного інтелекту	29.09-10.10.25	
3.	Дослідження роботи з дифузійними моделями	13.10-31.10	
4.	Створення та тренування власної моделі	3,11-14.11	
5.	Висновки по роботі	17.11-21.11	
6.	Оформлення магістерської роботи	24.11.18.12	
7.	Підготовка демонстраційних матеріалів, доповідь.	19.12-24.12	

Здобувач вищої освіти _____ Гліб КОШКАРЬОВ
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи _____ Олексій ШУШУРА
(підпис) (ім'я, ПРІЗВИЩЕ)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ ЩОДО ЗАХИСТУ
КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Кошкарьов Г.О. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 124 Системний аналіз
(код, найменування спеціальності)
освітньо-професійної програми Інтелектуальні системи управління
на тему: «Diffusion-моделі у генерації мультимедійного контенту».

Кваліфікаційна робота і рецензія додаються.

Директор ННІІТ _____ Катерина НЕСТЕРЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Кошкарьов Гліб Олександрович обрав тему кваліфікаційної роботи, метою якої була розробка розробка моделі для генерації контенту за допомогою дифузії. Під час виконання кваліфікаційної роботи Кошкарьов Г.О. показав добру теоретичну та практичну підготовку, вміння вирішувати самостійно питання та проводити дослідження. Робота виконана сумлінно та вчасно за планом.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Кошкарьов Г.О. на оцінку «відмінно» та присвоїти йому кваліфікацію магістр з інтелектуальних системи управління.

Керівник кваліфікаційної роботи _____ Олексій ШУШУРА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*
"___" _____20__ року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Кошкарьов Г.О. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри Інформаційних систем та технологій _____ Каміла СТОРЧАК
(назва) *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу
на здобуття освітнього ступеня магістра
здобувача вищої освіти ***Кошкарьов Гліба Олександровича***
на тему: «Diffusion-моделі у генерації мультимедійного контенту».

Актуальність.

Тема роботи «Diffusion-моделі у генерації мультимедійного контенту» є вкрай актуальною через стрімкий розвиток генеративного штучного інтелекту в медійну індустрію та необхідність автоматизації творчих процесів. Застосування таких методів для створення зображень, відео та аудіо дозволяє суттєво знизити часові та фінансові витрати при створенні унікального цифрового продукту. Дослідження теоретичних основ та практична розробка алгоритмів для генерації контенту відповідає викликам сучасної цифровізації та має вагому практичну значущість для розвитку інтелектуальних систем.

Позитивні сторони.

1. Зміст роботи повністю відповідає завданню, а поставлені задачі виконані у повному обсязі.
2. У роботі проведено глибокий аналіз сучасних генеративних архітектур.
3. Досліджено та навчено власну модель для генерації зображень.

Недоліки.

1. Не розглянуте питання безпеки і протидії технологіям з заміни облич.
2. Доцільно було б створити повноцінний мультимедійний продукт.

Відзначені зауваження не впливають на загальну позитивну оцінку магістерської кваліфікаційної роботи.

Висновок: *кваліфікаційна робота на здобуття ступеня магістра заслуговує оцінку "відмінно", а здобувач Кошкарьов Гліб Олександрович заслуговує присвоєння кваліфікації: магістр з Інтелектуальних системи управління.*

Рецензент: _____ _____
науковий ступінь, вчене звання *підпис* *Ім'я, ПРІЗВИЩЕ*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступня магістр: 64 стор., 35 рис., 28 джерел.

Мета роботи – розробка моделі для генерації контенту за допомогою дифузії.

Об'єкт дослідження – Системи штучного інтелекту для створення мультимедійного контенту за допомогою генеративних моделей.

Предмет дослідження – Архітектура та алгоритми дифузійних моделей (наприклад, DDPM, Stable Diffusion) для формування візуального контенту.

Короткий зміст роботи. Перший розділ це історичний розвиток GAI та теоретичні основи дифузійних моделей. У цьому розділі розглядається еволюція штучного інтелекту, починаючи від правило-орієнтованих систем 1950-х років (наприклад, Deep Blue) до статистичних методів навчання та сучасних нейронних мереж. Другий розділ присвячений технічній базі дифузійних моделей. Порівнюються GAN та VAE, описуються принципи прямої та зворотної дифузії (DDPM), де зображення спочатку перетворюється на шум, а потім нейромережа вчиться його поступово видаляти.

Третій розділ фокусується на складніших типах медіа. Розглядаються моделі Sora та Veo 3 для генерації відео. Описано методи генерації аудіо, зокрема принцип V-дифузії, та наведено огляд сервісів для створення музики (наприклад, Dance Diffusion) і покращення записаного голосу.

Четвертий розділ це практична частина роботи, де описано розробку власної моделі на мові Python за допомогою бібліотек PyTorch та DeepInverse. продемонстровано процес тренування мережі на датасеті рукописних цифр MNIST, налаштування «бета-графіка» шуму та результати генерації нових зразків із чистого гаусового шуму.

КЛЮЧОВІ СЛОВА: DIFFUSION-МОДЕЛІ, ДИФУЗІЙНІ МОДЕЛІ, МОДЕЛЬ, ГЕНЕРАЦІЯ, МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, ТЕКСТ-В-ЗОБРАЖЕННЯ, ГЕНЕРАЦІЯ ВІДЕО, ГЕНЕРАЦІЯ ЗОБРАЖЕННЯ.

ABSTRACT

The text part of the qualifying work for obtaining a bachelor's degree: 64 pp., 35 fig., 28 sources.

Objective – to develop a model for diffusion-based content generation.

Object of study – Artificial Intelligence systems for multimedia content creation using generative models.

Subject of study – Architectures and algorithms of diffusion models (e.g., DDPM, Stable Diffusion) for visual content generation.

Abstract. The first chapter explores the historical development of Generative AI (GAI) and the theoretical foundations of diffusion models. This section examines the evolution of AI, from the rule-based systems of the 1950s (e.g., Deep Blue) to statistical learning methods and modern neural networks. The second chapter is dedicated to the technical framework of diffusion models. It compares GANs and VAEs and describes the principles of forward and reverse diffusion (DDPM), where an image is first transformed into noise, and the neural network then learns to gradually remove it.

The third chapter focuses on more complex types of media. It discusses the Sora and Veo models for video generation. It also describes audio generation methods, specifically the V-diffusion principle, and provides an overview of services for music creation (e.g., Dance Diffusion) and recorded voice enhancement.

The fourth chapter represents the practical part of the work, describing the development of an original model in Python using the PyTorch and DeepInverse libraries. It demonstrates the process of training the network on the MNIST dataset of handwritten digits, configuring the "beta-schedule" for noise, and presents the results of generating new samples from pure Gaussian noise.

KEYWORDS: DIFFUSION MODELS, MODEL, GENERATION, MACHINE LEARNING, NEURAL NETWORKS, TEXT-TO-IMAGE, VIDEO GENERATION, IMAGE GENERATION.

ЗМІСТ

ВСТУП

Генеративний штучний інтелект з недавніх пір досяг неабиякого успіху, надавши можливість людям будь якого віку чи статусу створювати цікаві і захоплюючі зображення, комп'ютерні коди, музику чи навіть відео.

Засоби штучного інтелекту, які перетворюють текстові prompt запити на різного вигляду мультимедійний продукт набули неабиякої популярності за останній роки. Для багатьох людей це можуть бути просто спосіб скоротити час, проте існує верста людей для яких генеративні диф'южн моделі це є повноцінний заробіток або як мінімум серйозне хобі, і поки хтось не звертає уваги на подібні забавки, то є люди які вже в повну використовують блага сучасних генеративних моделей штучного інтелекта. А як всі знають, де щось популяризується там обов'язково з'явиться пустий простір наживи яку швидко наповнять різні версти населення нашої планети з метою набуття неправомірної вигоди.

Великі корпорації і іменити бренди розширюють частку залученності різних інструментів штучного інтелекту, в тому числі генеративного ШІ. У зв'язку з чим виникає питання глибше зануритись в цю тему і дослідити її починаючи з самих основ, щоб зрозуміти розвиток і куди рухається розвиток цього класу моделей ШІ, оскільки де фінансовий інтерес там і більші впливи спонсорів і зацікавленість ширшого кола сторін. Все частіше і частіше замовники звертаються спершу не до професійних дизайнерів чи розробників інтерфейсів, а до різних додатків чи чатів в яких прописавши короткий влучний prompt запит людина отримує декілька різних варіантів продукту чи дизайну продукту, це займає менше часу і створює відчуття безпосереднього виконання роботи, хоча й усю її виконав ШІ.

Популяризація і доступність text-to-image сервісів відкриває для користувачів нові інструменти, і нові проблеми, а саме не розуміння людьми суті запитів і як правильно їх сконструювати, як подати символи, щоб алгоритми спрацювали на твою користь, а не злились не робочий макет. Часто це стає не питанням "яких слів слід уникати" чи "які слова спрямують ШІ до правильного результат", велика частка людей не розуміє, що чатбот не "бачить" і не "розуміє", чатбот реагує чітко за алгоритмами і з тим як він навчений працювати. ШІ не

розуміє, що таке “пальці” або “вилки у виделки”.

Виникає тоді питання, що робити і як правильно працювати з ШІ, що йому писати і як. Для цього і написана ця робота, ця теми в ній також досліджено і вирішено. Питання і неточності в роботі з генеративними моделями буде досліджено і перевірено.

1. ІСТОРИЧНИЙ РОЗВИТОК GAI ТА ТЕОРЕТИЧНІ ОСНОВИ ДИФУЗІЙНИХ МОДЕЛЕЙ

1.1. Правило-орієнтовні системи

Генеративний штучний інтелект (GAI) - це модель чи група алгоритмів, які спроможні створювати контент, включно з текстами, зображеннями і відео імітуючи людську креативність.

Символічним прародителем ШІ можна вважати перші генеративні правило-орієнтовані системи (Rule-based generative systems) розроблені орієнтовно у 1950-х роках. Для того, щоб ці програми могли генерувати якісь дані, перед тим люди мали створити набір правил. Другим же компонентом можна назвати, як генеративний двигун, який мав створювати дані за допомогою різних формальних операцій. Він побудований на базі знань, що включає правила і факти. Далі вступають в силу вбудовані за допомогою кодування правила у формі простих описів (рис.1).

Рис.1. Схема роботи правило-орієнтованих систем

Також в цей період можна зазначити створення перших систем перекладу. Орієнтовно наприкінці 1950-х років було створено систему яка на базі граматичних правил і фактів з лінгвістичних знань було запроваджену першу таку систему.

У 1960 роках було створено першу систему синтезу мовлення. Використовуючи фонетичні і лінгвістичні правила вона дозволяла перетворювати у звук текст, хоча це і було характерним механічним,

комп’ютерним тоном і не було плавним, якими є сучасні аналоги синтезу мовлення.

Завдяки такому незначному інструментарію тодішнім технологам вдалося досягти перших успіхів в виконанні конкретних завдань. Схожість на сучасні генеративні моделі мінімальна і загалом умовна, оскільки кожна розроблена така програма чи виконане правило було ефективне виключно в межах конкретного завдання. Але для реального широкого використання такі моделі, на жаль, не прижились, оскільки їх використання завжди вело до проблеми виходу за межі правил, які були заздалегідь визначені розробником, який не може передбачити усі можливі сценарії поведінки користувача.

Якщо розглядати як приклад найраніший приклад вдалого масштабного використання алгоритмічних ШІ на жорстких запрограмованих правилах та символьній логіці з деревом рішень, то таким прикладом буде Deep Blue від компанії IBM. Досягненням цієї моделі ШІ була перемога Гаррі каспарова в шахи у 1977 році. Це відрізняється від генеративних чи дифузійних ШІ сьогодні, які зі сторони для користувача виглядають як симуляція “думки” чи “креативу”, ця версія перебирала мілйони можливих варіацій комбінацій ходів, оцінюючи можливі наслідки і можливі рішення супротивника і його відповідь на кожен крок. У моделі були чіткі правила і чіткі інструкції і велика база фактів.

1.2. Статистичні моделі машинного навчання

Статистичні методи машинного навчання стали революцією дозволивши ШІ частково вийти за межі інструкцій шляхом навчання на даних, а не чітких правилах. Такі методи поділяють на дискримінативні, які роблять прогнози, та генеративні, що описують розподіл даних і здатні відтворювати нові зразки. Із 1960-х років розвивалися різні підходи до побудови генеративних моделей, спрямовані на відтворення даних або наближене ймовірнісне виведення.

Ранні генеративні моделі формували явні статистичні описи даних. Найпоширенішим напрямом стали ймовірнісні графові моделі, де дані подаються як змінні у вигляді вузлів графа, а ймовірні зв’язки — як ребра.

Процес генерації інтерпретується як обчислення невідомих змінних за відомими. Застосовувалися методи максимізації правдоподібності, зокрема алгоритм EM. Значна частина робіт спиралася на марковські процеси, включно з прихованими марковськими моделями, які вводили латентні змінні для опису спостережень. Подальший розвиток призвів до моделей з двонаправленими залежностями й байєсовими мережами, що ефективно генерують дані високої розмірності, наприклад зображення.

Інший клас явних генеративних моделей представлено авто регресивними підходами, які створюють елементи послідовно, враховуючи попередні значення. Вони стали природними для обробки мови і звуку, а згодом — основою великих нейромережевих моделей, зокрема GPT.

До імпліцитних моделей належать нормалізовані потоки, які трансформують простий розподіл у складний через оборотні перетворення. У випадку стохастичних перетворень ці процеси можна інтерпретувати як дифузію. З часом такі підходи привели до появи сучасних дифузійних моделей та методів flow matching, що стали популярними в епоху глибинного навчання.

Зі зростанням складності моделей ключову роль почали відігравати штучні нейронні мережі, здатні апроксимувати будь-які розподіли. Еволюція архітектур (від перцептронів до CNN і RNN) суттєво розширила можливості генеративних систем. Розробка LSTM значно покращила генерацію тексту, а енергетичні моделі, включно з обмеженими Больцманівськими машинами, дозволили описувати розподіли через функції енергії.

Навчання таких моделей стало можливим завдяки алгоритму зворотного поширення похибки, який у 1980-х сформував основу оптимізації нейромереж.

Генеративні моделі потребують процедури виведення. Оскільки точне обчислення часто є недосяжним, дослідження сконцентрувалися на наближених методах — стохастичній апроксимації (зокрема MCMC) та варіаційному виведенні, що мінімізує розбіжність між реальним і наближеним розподілом.

Моделі які використовують статистичні методи звільнили користувачів від необхідності обмежуватись лише правилами для роботи і дозволили оперувати великими масивами даних. Також це відкрило можливість до генерації різного виду мультимедійного контенту, але все ще у простому вигляді, це були мовні чи голосові дані. Це стало хорошим підґрунтям для сучасни генераційних та дифузійних моделей. В свою чергу не можна ігнорувати необхідність обробляти велику кількість баз даних, залежність від якості цих даних. Оскільки якщо бази даних будуть неякісними, то і результат роботи з ними буде відповідним.Розробникам доводилось стикатись з проблемами вибору правильного алгоритму збору та аналізу даних і на ходу вирішувати проблему перенавчання моделі, оскільки модель могла стати занадто залежною від навчальних даних і погано працювати з новими даними.

1.3. Нейронні мережі та глибоке навчання

Перші ідеї штучних нейронів з’явилися ще в середині XX століття. Модель Маккалоха — Піттса (1943) заклала математичну основу для побудови штучних нейронів, хоча її виразна здатність залишалася вкрай обмеженою. Наступним важливим кроком став перцептрон Розенблатта (1957), який уперше продемонстрував можливість автоматичного навчання на основі прикладів.

Попри початковий ентузіазм, розвиток нейронних мереж на тривалий час сповільнився через теоретичні обмеження. М. Мінський та С. Пейперт у 1969 році показали, що одношарові перцептрони неспроможні розв’язувати низку фундаментальних нелінійних задач (наприклад, XOR). Це спричинило період, який увійшов в історію як «зима штучного інтелекту», коли інтерес і фінансування в цій сфері суттєво скоротилися.

Відродження досліджень стало можливим завдяки появі алгоритму зворотного поширення помилки (backpropagation), запропонованого Румельхартом, Хінтоном і Вільямсом у 1986 році. Він дав змогу ефективно тренувати багатошарові мережі й відкрив шлях до побудови складніших архітектур. Проте навіть після цього розвиток залишався обмеженим через

проблеми згасання та вибуху градієнтів, нестачу обчислювальних потужностей і недостатність великих масивів даних для навчання.

Справжній прорив у глибокому навчанні став можливим лише з появою потужних графічних прискорювачів (GPU) та спеціалізованих процесорів (TPU). Ще два десятиліття тому жодні доступні обчислювальні системи не могли б забезпечити тренування моделей на обсягах даних, характерних для сучасних нейромереж. Саме поєднання апаратного прогресу та значного зростання доступних даних зробило можливим перехід до глибоких архітектур.

Наявність великих відкритих і корпоративних датасетів стала одним із ключових чинників стрімкого розвитку глибокого навчання. Компанії на кшталт Google (Alphabet), Meta, Microsoft, OpenAI та Amazon за роки накопичили величезні масиви даних, які використали як у власних дослідженнях, так і зробили доступними для спільноти. Набори даних, такі як ImageNet, Open Images, MS COCO, LAION-5B, Common Crawl, YouTube-8M, забезпечили можливість тренувати моделі на безпрецедентно великих і різноманітних вибірках. Це дало змогу перейти від експериментальних невеликих наборів до масштабних представницьких датасетів, що відображають різноманітність реального світу. У результаті моделі стали точнішими, здобули кращу здатність до узагальнення та стали менш схильними до перенавчання.

У галузі комп'ютерного зору зростання масштабів даних радикально змінило підходи до побудови моделей. Зокрема, завдяки ImageNet стала можливою поява AlexNet (2012), яка започаткувала сучасну епоху глибокого навчання. Це стимулювало створення CNN-архітектур, таких як VGG, ResNet, EfficientNet, які стали стандартами в обробці зображень. Подібним чином набори на кшталт MS COCO, Open Images і LAION-5B стали фундаментом для розвитку генеративних моделей і мультимодальних систем. Наприклад, саме LAION забезпечив базу для тренування Stable Diffusion — моделі, що відкрила новий етап у генерації зображень.

У сфері обробки природної мови вплив великих корпусів був не менш

значущим. Common Crawl, Wikipedia, WebText та інші масштабні текстові бази стали основою розвитку трансформерів — GPT, BERT, T5 та багатьох інших моделей, здатних опановувати складні мовні структури, семантичні залежності та контекстні закономірності. Здатність сучасних мовних моделей виконувати логічні міркування, аналіз, генерацію тексту й мультимодальні завдання безпосередньо пов’язана з масштабом використаних даних.

Для генеративних моделей, особливо дифузійних, широкий спектр варіативних даних є фундаментально важливим. Дифузійні моделі відновлюють структуру зображення з шуму, тому якість генерації прямо залежить від різноманітності контенту в навчальному датасеті. Саме завдяки даним на рівні LAION-5B стало можливим створення моделей на кшталт Stable Diffusion, DALL·E чи Midjourney.

Загалом поява великих, різноманітних і якісно анотованих датасетів стала одним із визначальних факторів розвитку глибокого навчання та сучасних генеративних систем. Дані перетворилися на стратегічний ресурс, що визначає точність, гнучкість та творчий потенціал сучасних ШІ-технологій.

1.4. Ризики та етичні недоліки генерації мультимедійного контенту

З активним розвитком генеративних технологій, на жаль, не встигають так же швидко розвиватись і етичні норми пов’язані з цією технологією, або не встигає розвиватись суспільний договір і аморальні версти користувачів не отримують чинного покарання за свої дії. Також слабким залишається правове поле, оскільки воно ще не готове регулювати цю галузь, а за генеративними сервісами часто стоять малочисельні компанди. Слід не забувати, що генеративні моделі навчаються на іншому контенті, цей контент часто це чужі роботи, чуже надбання і чиясь власність. Тому для більшості митців використання їх робіт, це значна проблема оскільки потім роботодавець відмовляється від їх дорогих послуг на користь більш дешевої робочої сили ШІ. Ці всі аспекти пробуджують в суспільстві жваві обговорення де люди, як зазвичай буває, діляться на різні табори.

1.4.1. Технологія DeepFake

Одне з найбільших і найперших обурень, від небезпечних і зневажливих актів використання штучного інтелекту, стало таке явище як “DeepFake”. Ця технологія дозволяє змінювати обличчя та голоси людей на відео й аудіозаписах, досягла такого рівня реалізму, що відрізнити фальшивку від оригіналу стає майже неможливо. Якість заміни відкрив неабиякий потенціял для різних маніпуляцій, це були різні сфери : медійна, шантаж, шахрайство, створення компроматів та навіть військові і політичні цілі під час війни і нападу на іншу державу. Є численні позови зірок обличчя яких були використані без їхньої згоди не дуже доброчесними ентузіастами. Хоча у технології з заміною облич є і свої очевидні плюси у вигляді використання облич померлих акторів для фільмів, але для цього, на думку людей, потрібна згода за життя. Оскільки мало хто хоче, щоб після його смерті його обличчя стало не більше ніж забавкою яка може опинитись не в тих руках.

1.4.2. Навчання на чужій власності і взнечінення людської творчості

Мабуть, найгучніша етична проблема пов'язана з тим, як саме навчаються ці потужні моделі ШІ. Вони "тренуються" на мільярдах зображень, текстів, музичних треків та відео, зібраних з інтернету, більшість із яких захищені авторським правом. Це призводить до ситуації, коли комерційний продукт ШІ фактично базується на неоплачуваній праці тисяч митців.

Це питання вже стало предметом серйозних судових розглядів. Кілька колективних позовів було подано проти розробників популярних моделей генерації зображень, зокрема Stability AI (Stable Diffusion) та Midjourney, а також компанії OpenAI. Художники та фотоагентства стверджують, що використання їхніх робіт без дозволу та компенсації для навчання комерційних інструментів є прямим порушенням авторських прав.

Особливого розголосу набув випадок, коли генеративні моделі почали імітувати унікальний стиль відомих митців та студій. Відомий випадок пов’язаний зі студією Ghibli, легендарний японський анімаційний гігант, публічно висловлювала своє обурення та занепокоєння з приводу створення зображень "у стилі Ghibli" штучним інтелектом. Це не лише моральна проблема

використання чужого напрацьованого стилю без згоди, але й економічна загроза: якщо ШІ може створити контент, що імітує роботу конкретного митця, це знецінює оригінальну людську працю та зменшує потребу в послугах самого автора.

Окрім юридичних аспектів, існує глибока етична дискусія про вплив ШІ на професії у творчих індустріях. Здатність ШІ генерувати високоякісні ілюстрації, тексти, музику та код за лічені секунди викликає обґрунтовані побоювання щодо масової втрати робочих місць дизайнерами, копірайтерами, композиторами та аніматорами.

Ризик полягає не тільки в автоматизації рутинних завдань, але й у потенційній стагнації людської креативності. Якщо "швидко і дешево" стає домінуючим фактором, ринок може почати віддавати перевагу згенерованому контенту, знецінюючи унікальний людський досвід, інтуїцію та роки навчання, які стоять за справжнім мистецтвом.

1.4.3. Використання в анімаційних серіалах

Ще однією проблемою зі створення мультимедійного продукту є неодноразові випадки, коли глядачі ловили аніме на випадках користуванням штучним інтелектом. Є безліч матеріалів (рис.1.2) і випадків як користувачі помічали найбільш очевидний дефект, який чітко вказує на використання штучного інтелекту - кількість пальців, якщо вона більша чи менша за 5 (в тих аніме де кількість пальців у персонажів дорівнює п’яти). Це відома проблема генеративних моделей, вони не завжди розуміють окремі елементи на зображеннях, це може бути кількість гострих кінців у виделки, кількість пальці на руках чи ногах, або китайські палички в тарілці з раменом. Оскільки штучний інтелект знає як має виглядати щось, але не розумі тонкощів, то він часто додає зайві пікселі, або не здатен прибрати те що просить замовник.

Рис. 1.2. Приклад як відомі студії використовують ШІ для аніме

Подібні історії вдаряють по репутації анімаційних студій і псують враження від продукту, оскільки глядач відчуває, що на ньому зекономили і студія не ставиться до свого продукту серйозно.

1.4.4. Актуальні можливості для користувачів генеративних ШІ

Поговоривши про недоліки генеративних ШІ, неможливо не поговорити, про надбання генеративних моделей у вигляді можливостей які вони відкривають перед користувачем.

Користь дифузійних моделей полягає у простоті реалізації бажання користувача створити мультимедійний продукт різною складності. Якби більшості людей сказали, що вони можуть створити зображення в стилі їх улюбленого аніме, мультика, або художника, то вони б у це ніколи не повірили. Але реальність в тому, що це можливо. Будь хто з доступом до браузера, в будь якій точці світу, в якій присутнє покриття інтернету, може створити дивовижний

арт у стилі студії Ґіблі, або будь якого популярного художника чи в будь якому стилі мистецтва, без прив'язки до конкретного автора. Для цього потрібен лише короткий промпт. Або ви можете створити власний автопортрет завантаживши власне фото чи фото кота, що перетворити його на персонажа гри чи фентезійної історії. Для цього не треба мати спеціальну освіту чи довго тренувати і відточувати навички малювання чи дизайну, не треба проходити курси чи художню школу. Звісно, ніхто не применшує необхідність, якщо весь світ буде розраховувати виключно на штучний інтелект, то людство занепаде і це буде найгірший сценарій штучного інтелекта, але й не слід тішити себе думкою, що кожен тепер став Пікассо тільки від того, що написав промпт в чат боті. Все описане вище лише стосується приємного доповнення до коротких маленьких веселощів кожної людини. Мова йдеться про те, що кожен щоб отримати посмішку і повеселити коло друзів може витратити трохи часу і отримати картинку, яка підніме настрій чи збере певну кількість лайків. Це не пошук визнання чи овацій, це реалізація маленьких бажань, пошук припливу кортизолу, спосіб потішити близьких людей і привернути чиюсь увагу. І ці маленькі радощі можливі лише завдяки популяризації і розвитку генеративних моделей. Компанія друзів може створити зображення кота у стилі аніме персонажа, щоб потім створити пост на Linkedin і використати його як обкладинку. Ще років 10 назад для цього треба було б шукати і наймати художника, який міг би за це взятись, витрачати на це не малі гроші, пояснювати завдання з надією, що він розуміє про що ви говорите і очікувати, що він правильно реалізує ваше бажання. І всі ці переваги появи генеративних моделей було описано лише на найпростішому – генерації зображень. Якщо мова заходить за генерацію відео і аудіо, то це стає ще більш важким і ресурсозатратним процесом, про який легше сказати – неможливо і ніколи навіть не думати про те щоб створити власний відеоролик. Але сьогодні і це можливо. Велика кількість людей просто щоб повеселитись створюють короткі відео ролики. Люди ведуть тікток канали зі згенерованим контентом і заробляють на цьому гроші. Є цілі сервіси які пропонують генерувати brainrot контент ряд для фонового програвання під час аудіоряду. Окрім цих видів заробітку, поява ШІ створила цілу нову професію – промпт-

інжиніринг. І сьогодні спеціалісти, які вміють ефективно створювати запити для ШІ-моделей, відомі як промпт-інженери, користуються зростаючим попитом. У 2025 році ця навичка розглядається не просто як доповнення, а як повноцінна і цінна компетенція, за яку платять гроші. Ця професія слугує мостом між людиною та штучним інтелектом. Промпт-інженери досконало зрозуміли, як “спілкуватись” з великими мовними моделями, щоб отримувати максимально точні та якісні результати.

Для більшості людей “спілкування” з мовними моделями це щось нове і невідоме, часто результат залишається далеким від того, що собі уявляла людина, і отримати правильний очікуваний результат радше вдале співпадіння ніж точне розуміння, що написане правильно спрацювало. Тому промпт-інженери допомагають розкрити повний потенціал ШІ. Хоча кожен може написати простий запит, створення складних, багаторівневих промптів для виконання специфічних завдань — це мистецтво. Особливо цінно, коли потрібно, щоб ШІ згенерував матеріал у певному стилі, форматі або з урахуванням безлічі нюансів.

Нові можливості для фахівців. Ця навичка вже потрібна в різних сферах, включаючи:

- маркетинг (створення контенту для соцмереж);
- SEO (оптимізація текстів);
- розробка (створення коду та документації);
- дизайн (створення візуальних матеріалів).

Важливо зазначити, що хоча попит на окремих "чистих" промпт-інженерів зростає, ще частіше ця навичка стає додатковою перевагою для фахівців в інших галузях. Якщо ви вже працюєте у сфері розробки, маркетингу або дизайну, володіння промпт-інжинірингом зробить вас більш цінним спеціалістом.

Наприклад, розробник, який вміє ефективно використовувати ШІ, може прискорити свою роботу, а маркетолог — генерувати креативний контент швидше та якісніше. Тому на сьогодні можна говорити, про негативний вплив ШІ на ринок, але ігнорувати можливості які несуть мовні моделі неможливо

ігнорувати. Не можна відмовлятись від прогресу.

2. DIFFUSION ГЕНЕРАЦІЯ ЗОБРАЖЕНЬ

2.1. Генеративні моделі GAN і VAE

Одним з найближчих етапів у розвитку штучного інтелекту є Generative Adversarial Nets і основна ідея цього методу полягає в одночасному тренуванні двох нейронних мереж: дискримінаторної і генераторної, які навчаються в конкурентному середовищі. Основна мета генератора це створювати нові екземпляри даних, це можуть бути зображення, відео тощо. Ціль ж дискримінатора ідеально до найменших дрібниць розрізняти нові згенеровані дані від оригіналу. І ці дві моделі постійно змагаються в тому щоб перевершити і перемогти один одного, для генераторної перемогою буде створити таке зображення, яке неможливо буде відрізнити від оригінала, а для другої моделі ідеально навчитись відрізняти згенеровані зображення від справжніх.

Проте жодна моделей не має перемогти, а вони мають завжди перебувати в балансі, оскільки якщо переможе дискримінатор і навчиться ідеально розрізняти всі згенеровані дані, то генератор почне створювати нісенітницю і процес навчання зупиниться, оскільки відрізняти такі дані буде занадто легко. Цей сценарій називається vanishing gradients. Якщо ж генератор перемагає, тоді створюючи ідеальні підробки дискримінатор через відсутність можливості відрізнити дані перестає вчитись і генератор перестає покращуватись. Тому ця модель ідеально існує лише в одному сценарії - в рівновазі Неша.

Ось як побудована це змагання (рис.1.4.1) :

1. Генератор отримує випадковий шум z і намагається перетворити це на реальні дані.
2. Дискримінатор отримує справжній екземпляр з реальних даних і згенерований екземпляр від генератора.
3. Дискримінатор оновлюється щоб краще розпізнати згенеровані дані від справжніх
4. Генератор оновлюється щоб підвищити шанс видати згенероване за справжнє.

5. Цикл повторюється, моделі стабілізуються

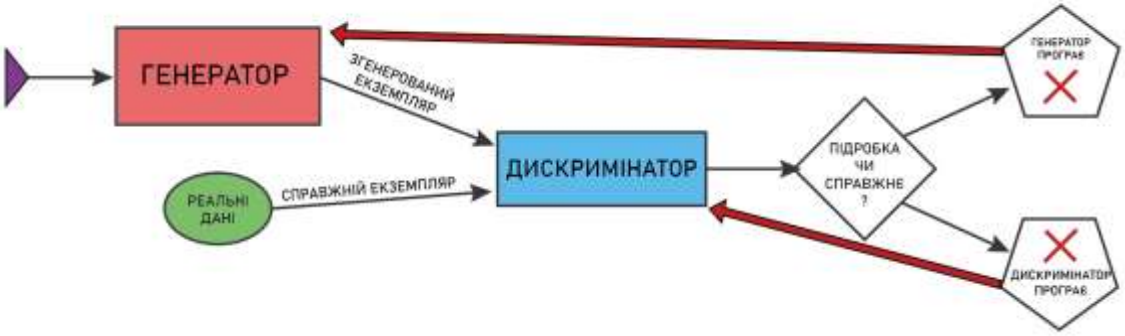


Рис. 1.4.1. Модель гри GAN

Вигляд математичної формули описаного процесса гри GAN :

$$\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}[log\ D(x)] + E_{z\sim p_z}[log\ (1 - D(G(\text{z})))]$$

Наразі GAN широко використовуються для створення зображень для навчання інших моделей створювати реалістичні фотографії, або генерувати картинки по типу Текст-в-Зображення.

Принцип роботи: VAE складається з двох частин: кодувальника (encoder), що стискає дані у стиснутий латентний простір, і декодувальника (decoder), що відновлює дані з цього простору. Латентна змінна: На відміну від дифузійних моделей, VAE навчається відображати дані на безперервний, низьковимірний, ймовірнісний латентний простір. Генерація: Нові дані генеруються шляхом вибірки точки з цього латентного простору (зазвичай, з гауссівського розподілу) і передачі її декодувальнику.

VAE складається з кодувальника (encoder) і декодувальника (decoder). Кодувальник стискає дані у латентний простір, який собою набір числе - вектор. Декодувальник відновлює ці дані в даному випадку у пікселі, щоб утворити потрібне зображення. На відміну від дифузійних моделей VAE навчається відображати дані на безперервний, низьковимірний, ймовірнісний латентний простір і нові дані генеруються вибіркою з цього латентного простору і передачі точки з цього простору декодувальнику.

На відміну від GAN варіаційні автокодери (VAE) складаються з кодера та декодера. Кодер відображає вхідні дані у прихований простір, коли як декодер відображає точки у прихованому просторі назад у простір вхідних даних. Основна відмінність VAE від GAN, який нагадаю намагається досягти “ідеального” результату в грі змагання, в тому, що VAE намагається отримати середнє очікування, а не ідеальну правдоподібність деталей, через що зображення згенеровані за допомогою VAE виходять часто розмитими. Якщо описати цей процес на прикладі, то це матиме наступний вигляд. Почавши реконструкцію з трьох зображень. : з світлішим котом, темнішим котом і третім кольоровим, модель почне виводити усереднений варіант цих трьох зображень, тому що це мінімізує похибку (MSE) або Гаусівську ймовірність. В свою чергу GAN не намагається оптимізувати MSE.

2.2. Основи дифузійних архітектур і навчання моделей

2.2.1 Принцип прямої і зворотної дифузії і тренування

Наукова робота з якою дифузійні моделі було представлено світу була видана у 2020 і називалась “Denoising Diffusion Probabilistic Models” написана Джонатаном Хо і іншими співавторами. В цій роботі дифузійні моделі не були представлені, самі моделі було представлено ще у 2015 році у роботі під назвою “Deep Unsupervised Learning using Nonequilibrium Thermodynamics” і в цей період світ захоплювався результатами які генерувала інша модель, зображення які виходили при роботі з GAN моделлю набирали популярність і через це дифузійні моделі набули малого розголосу і уваги, також слід зазначити, що моделі 2015 року були дуже складними в обчисленнях і якість не була значно кращою за результати роботи GAN моделей. Проте все змінилось з виходом більш потужних GPU і з виходом у 2020 році DDPM, наразі це є найбільш цитована робота в галузі ШІ, яку цитували понад двадцять тисяч разів на період 2025. Завдяки цій роботі було досягнуто ряд покращень в роботу з генераціями мультимедійного контенту за допомогою ШІ. Стан генерації зображень значно покращився, а мета навчання значно спростилась, це зробило цей метод більш доступним порівняно з попередніми роками і підсвітило цікавіший взаємозв’язок

з попередніми роботами дифузійних моделей.

Основною відмінністю дифузійних моделей від інших моделей з латентними змінними полягає в умовному розміщенні (approximate posterior) $q(x_{1:T}|x_0)$, суть якого полягає в поступовому додаванні шуму, поки початкове зображення x_0 не перетвориться на чистий шум xt і зворотному декодінгу, це процес прибирання шуму. Виглядає це наступним чином :

$$q(x_t|x_{t-1}) := N(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

Сам процес додавання шуму є Марківським ланцюгом. Стан зображення в цьому випадку на кроці *t* залежить лише від попереднього кроку *t-1*.

β - контролює кількість шуму, що вноситься на кожному кроці, параметр β не є сталим, він змінюється від *t=0* до *t=T*. Цей процес називається розклад шуму (Noise Schedule), спочатку додається трохи шуму, а під кінець багато. Цей процес буває лінійним та косинусним., x_0 - це початкове зображення, x_1- зображення до якого додали шум, що є нашою наступною точкою в нормальному розподілі (рис.1.5.1) $q(x_t|x_{t-1}) := N(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$.

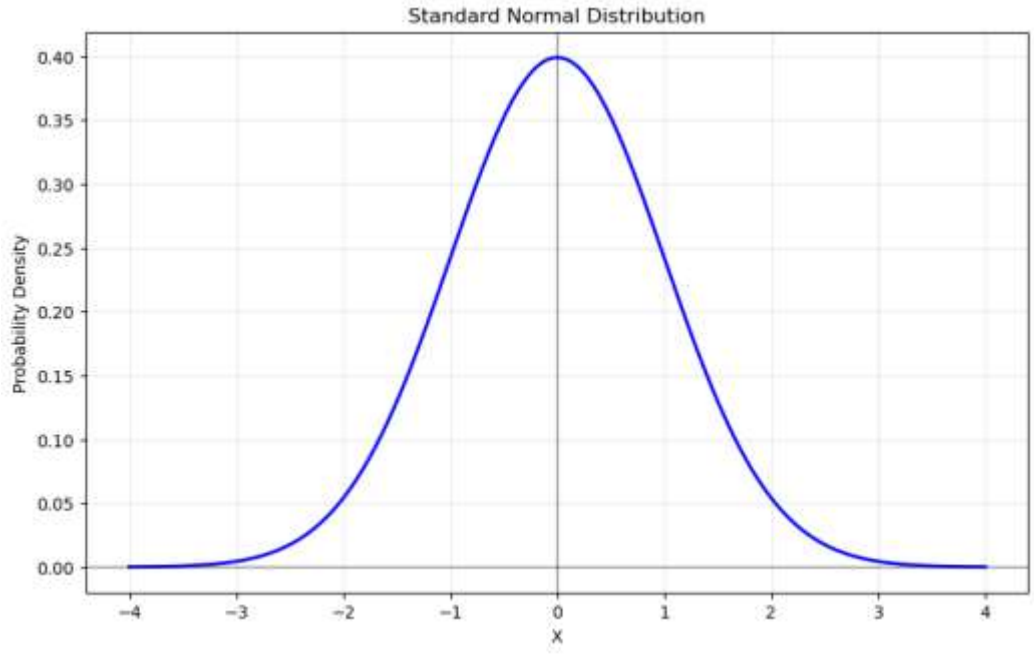


Рис. 1.5.1. Нормальний розподіл

x_2 це наступна випадкова точка з нормального розподілу з математичним очікуванням $\sqrt{1-\beta_2}x_1$, x_Tчистий шум. Якщо показувати це зображеннями, то

ми маємо наше початкове зображення x_0 (рис. 1.5.2)

рис. 1.5.2. Перше зображення x_0

до нього додається контрольована кількість шуму β і отримуємо x_1(рис.1.5.3)

рис. 1.5.3. Друга ітерація зображення x_1

далі ще шуму використовуючи щільність ймовірності q і отримуємо x_2

$$q(x_t|x_{t-1})$$

рис. 1.5.4. третій крок додавання шуму x_2

і так далі поки не вийде чистий шум x_T(рис. 1.5.4)

рис. 1.5.5. Чистий шум x_T

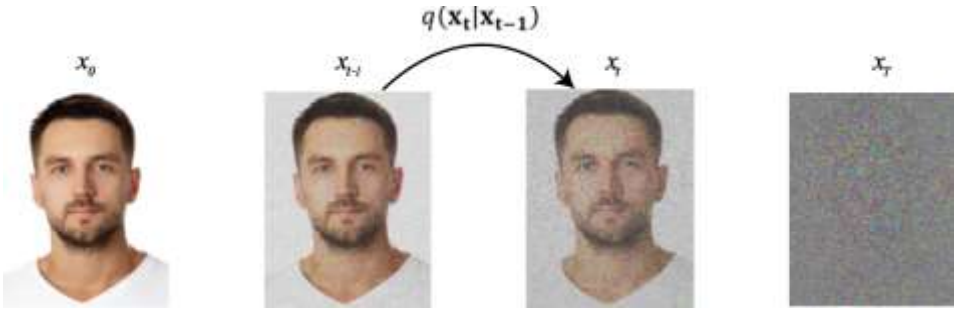


Рис.1.5.6. Прямий дифузійний процес

Щоб не рахувати шум крок за кроком, існує “трюк репараметризації” (The Reparameterization Trick). Суть цього трюку полягає в тому, що якщо ми не хочемо рахувати 499 кроків додавання шуму, а хочемо одразу подивитись результат на картинці під номером 500, то слід використати параметр $a_t = 1 - \beta_t$ та кумулятивний добуток $\bar{a}_t$. В такому випадку формула виглядатиме :

$$q(x_t|x_0) = N(x_t; \sqrt{\bar{a}_t}x_0, (1 - \bar{a}_t)I$$

Це значно пришвидшить тренування, оскільки дозволить не рахувати всі кроки, а переходити до будь якого як в прямому дифузійному процесі так і в зворотньому.

Після щойно описаного прямого дифузійного процесу (рис. 1.5.6), відбувається зворотній процес $p_0(x_{t-1}|x_t)$, який спрямований на скасування шуму з метою навчання нейронної мережі його поступовому усуненню і реконструкції максимально наближених до вихідних даних. Описується це так :

$$p_\theta(x_{t-1}|x_t) \sim N(x_{t-1}; \mu_\theta(x_t), \Sigma_\theta)$$

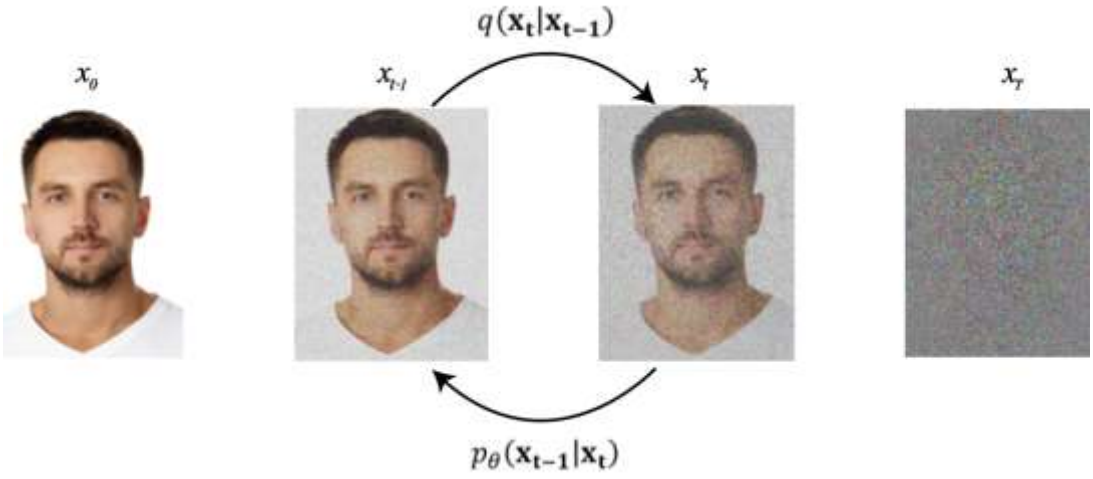


Рис.1.5.7. Прямий і зворотній процеси

Символ θ (тета) вказує на те, що нейромережа має сама визначити параметри, щоб найбільш ефективно видалити шум, цей параметр вказує, на навчання нейромережі в цьому процесі. Теоретичним фундаментом для цього виступає Баєсова статистика (Variational Inference), хоча й на практиці вона значно спрощується. З точки зору статистики, метою є знайти параметри тета θ, але проблемою є неможливість прорахувати це напряму, оскільки для цього доведеться проінтегрувати усі можливі шляхи шуму. Тоді слід використати баєвський трюк (ELBO) і замість розрахунку точної ймовірності прорахуємо її нижню межу. В згаданій раніше публікації DDPM автори поділились цікавим спостереженням, яке і слід використати. Дивергенція Кульбака-Лейблера – це спосіб вимірювання того, наскільки близькі два розподіли ймовірностей, його можна уявити як відстань у просторі ймовірностей (рис. 1.5.8)

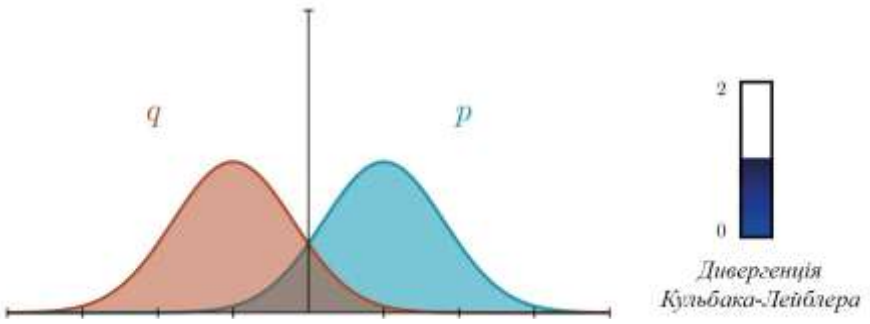


Рис. 1.5.7. Візуалізаця Дивергенції Кульбака-Лейблера

Якщо середні значення розподілів дуже різні, то розбіжність ДКЛ буде значною (рис. 1.5.8)

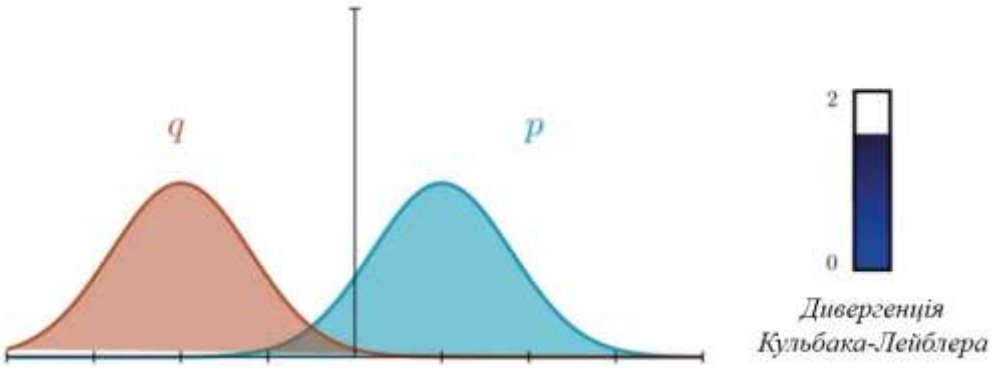


Рис. 1.5.8. Збільшена розбіжність ДКЛ

В іншому випадку, якщо розподіли будуть дуже схожими, то розбіжність буде не значною (рис. 1.5.9).

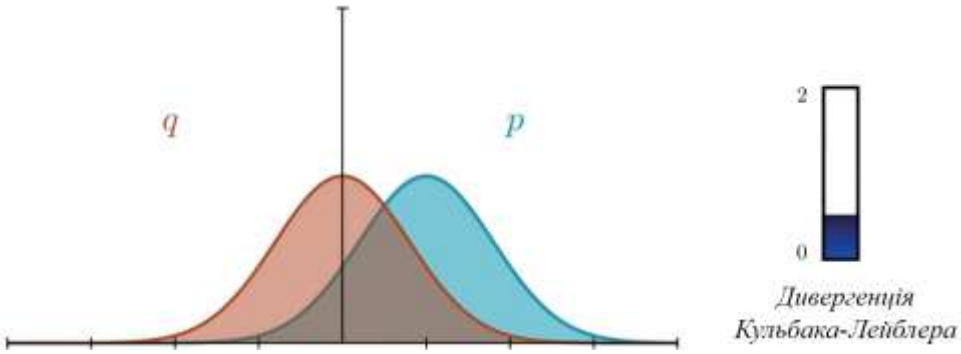


Рис. 1.5.9. мала розбіжність ДКЛ

І щоб розбіжність ДКЛ досягла нуля два розподіли мають бути абсолютно ідентичними (рис. 1.5.10)

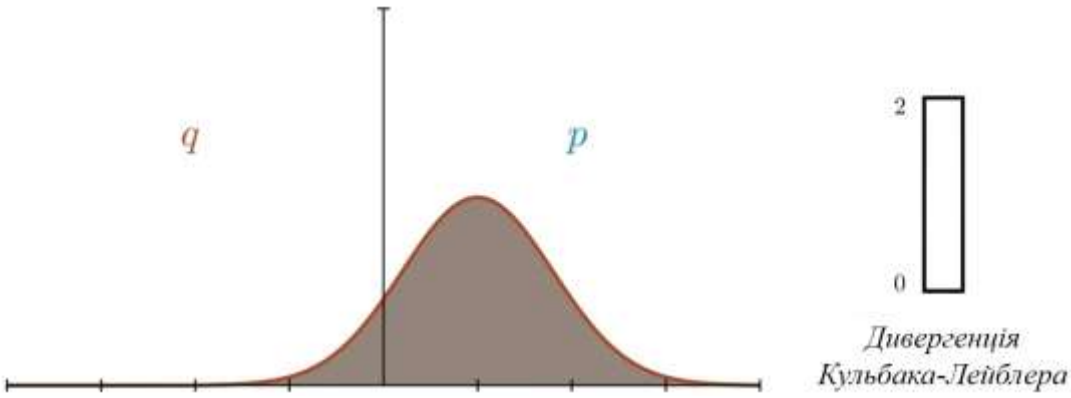


Рис. 1.5.10. Нульова розбіжність ДКЛ

Оскільки обидва наші розподіли q і p є Гауссівськими, то мінімізація Кульбака-Лейблера дивергенції між ними математично зводяться до віднімання середніх значень а оскільки середня значення залежить від доданого шуму, то формула Баєсова спрощується до :

$$L_{simple} = ||\epsilon - \epsilon_\theta(x_t, t)||^2$$

ϵ це справжній шум, який було додано (його величина відома оскільки він був доданий нами) ϵ_θ це шум який визначила нейрона мережа. І ми використовуємо Баєвську статистику для формули і доведення працездатності метода. Хоча вже при роботі з кодом дифузійної моделі і навчанні складні ймовірності та інтеграли не рахуються.

Відкритим залишається питання : “Як же відбувається процес тренування нейронних мереж?”

Для цього використовується велика сума ДКЛ між деякими розподілами.

$$-\mathbb{E}_q \left[\sum_{t>1} D_{\mathrm{KL}}(q(x_{t-1} \mid x_t, x_0) \parallel p_\theta(x_{t-1} \mid x_t)) \right]$$

де $p_\theta(x_{t-1} \mid x_t))$ це крок зворотної дифузії, а $q(x_{t-1}|x_t, x_0)$ це справжній апостеріорний розподіл.

Якщо ми маємо прямий процей $q(x_t|x_{t-1})$, то вже відомо, що існує зворотній процес $p_\theta(x_{t-1}|x_t)$ і якщо накласти умову на вихідне зображення x_0, то це дозволить фактично обчислити точний зворотній процес. Цей справжній зворотній процес, який використовує чисте зображення x_0 називається апостеріорним (рис. 1.5.11).

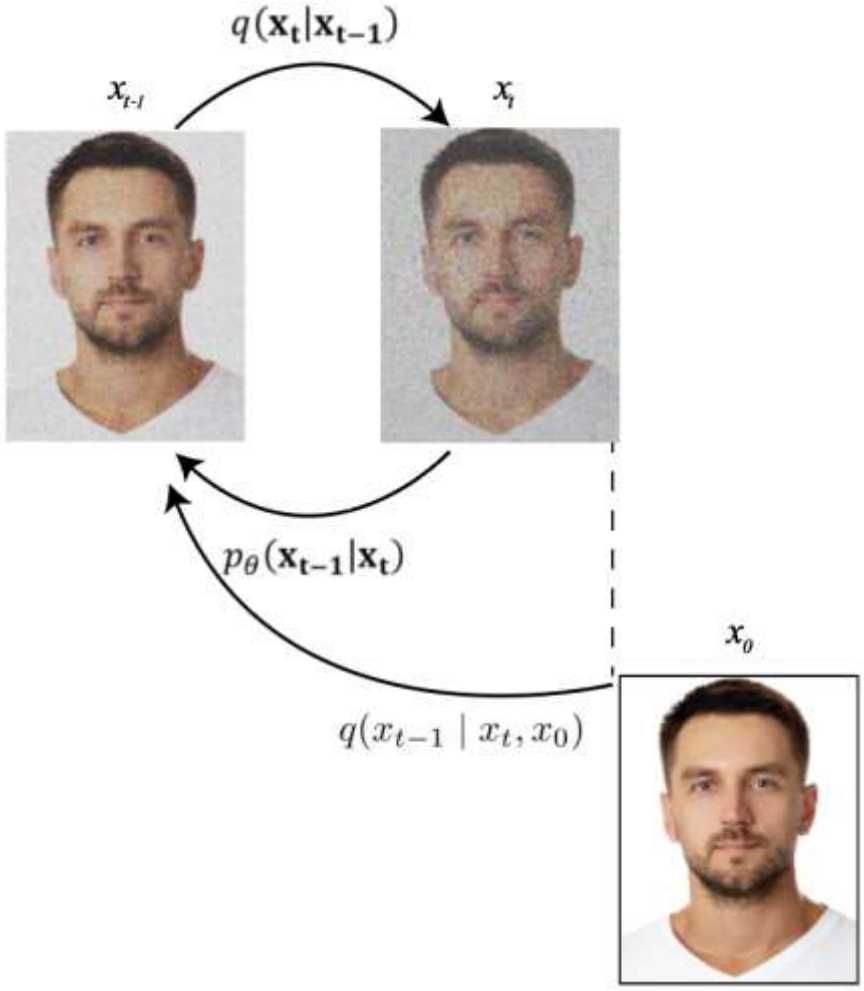


Рис. 1.5.11. Апостеріорна функція

Але використання апостеріорної функції підходить лише для навчання моделі, її використання не дозволяє отримувати зображення, оскільки генерація зображення відбувається з чистого шуму з метою отримати x_0. І те що відбувається це навчання нейронної мережі відповідати справжньому апостеріорному значенню (рис. 1.5.12), тому яке б можна було обчислити, якби мали доступ до x_0, навіть якщо на практиці мережа бачить лише x_t.

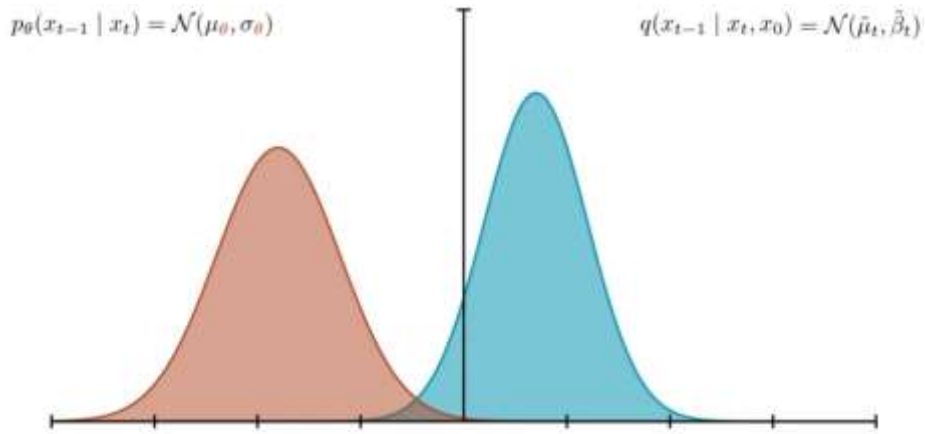


Рис.1.5.12. Гаусове Апостеріорне значення

Нейрона мережа передбачить параметри гаусової функції і оскільки ми намагаємось зіставити два гаусові розподіли, але можливо контролювати лише середнє значення найближенного, найкраще, що можна зробити, це наблизити його середнє значення якомога ближче до середнього значення апостеріорного розподілу (рис. 1.5.13), отже мінімізація суми ДКЛ у цільовій функції насправді означає просто наближення кожного наближеного апостеріорного значення до його справжнього аналога. А оскільки можна контролювати лише середнє значення кожного приблизного апостеріорного значення, то найкращим варіантом є зменшення відстані до середнього значення кожного справжнього апостеріорного знання.

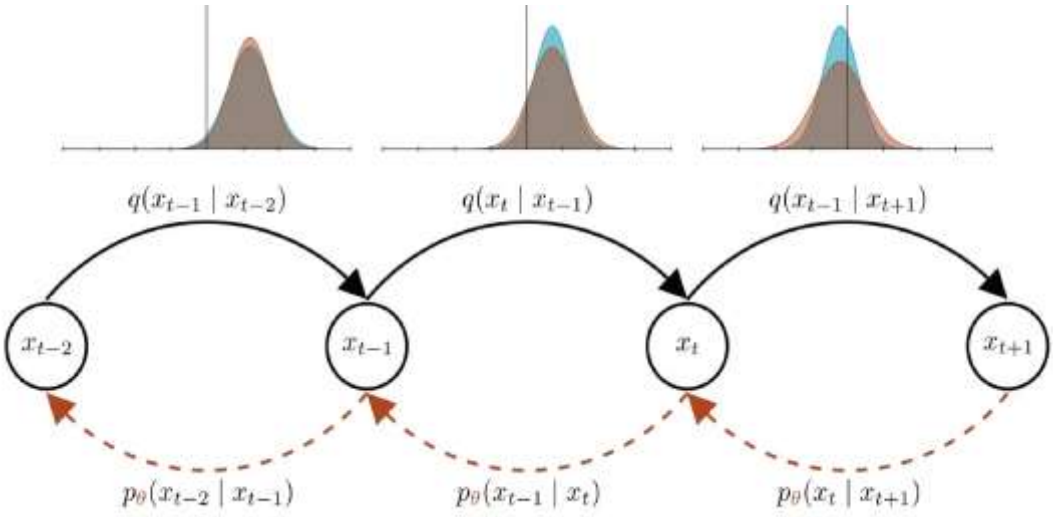


Рис.1.5.13. Візуалізація середнього значення апостеріорного значення

Тепер коли визначено ELBO можна сформулювати кінцеву мету тренування спростивши його ще більше. мінімізації суми ДКЛ було спрощено велику суму ДКЛ:

$$-\mathbb{E}_q \left[\sum_{t>1} D_{\mathrm{KL}}(q(x_{t-1} \mid x_t, x_0) \mid\mid p_\theta(x_{t-1} \mid x_t)) \right]$$

до квадратів:

$$\mathbb{E}_q \left[\sum_{t>1} \frac{1}{2\sigma_t^2} \|\tilde{\mu}_t - \mu_\theta(x_t, t)\|^2 \right]$$

Кожне з цих рівнянь вимірює як далеко передбачене середнє значення $\mu_\theta(x_t, t)$ находится від справжнього апостеріорного $\tilde{\mu}t$ і відстань зважується коефіцієнтом, який залежить від сігма-t. Цю втрату слід спростити ще більше,

що дійти до остаточного виразу.

Замкнута форма для справжньої апостеріорної середньої мютильди :

$$\tilde{\mu}_t = \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t$$

спрощується за допомогою параметризації, яка вже використовувалась раніше

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\epsilon$$

завдяки цьому xt записується як сума між x_0 та ϵ, чистий гаусівський шум, тож x_0 можна записати як залежність лише від x_t та ϵ.

$$x_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \sqrt{1-\alpha_t}\epsilon)$$

Якщо підставити це у вираз мю-тильди, то вийде вираз, який залежить лише від x_t та ϵ.

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$$

Далі виникає ключова ідея : оскільки деноізінг також має доступ до xt, то можна застосувати той самий трюк до передбачуваного середнього μt мережі та записати його як функцію від xt та передбачуваного складового шуму ϵ_θ

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

Тепер якщо підставити обидва ці вирази у втрату, складові в x_t зменшуються, що відкриває кінцеву цільову функцію Final loss function :

$$\mathcal{L} = \mathbb{E}_q\left[\sum_{t>1} \frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

Отже ϵ_θ стає оцінкою мережі щодо шуму, який було додано до x_0 для отримання x_t, а мінімізація втрат означає зробити цей прогноз масимально точним наскільки це можливо. Замість підсумовування по всіх часових кроках, слід просто вибирати випадковий часовий крок t для кожного зразка і для великої кількості зразків це сходиться до початкової мети. Це і є простою і зрозумілою метою навчання мережі.

2.2.2. Принцип навчання CLIP

На прикладі дифузійних моделей було розібрано, як відбувається пряма і зворотня дифузія. Це те що допомагає навчати нейроні мережі, але щоб

користувач міг з цим працювати, йому треба інтерфейс взаємодії. Для звичайного користувача потрібна зрозуміла можливість без додаткового складного навчання і навичок реалізувати власне бажання, ідею і потреби. Цим інструментом і інтерфейсом є текстове вікно з чат ботом. Але виникає поняття, як такі моделі як DALL-E розуміє, як текстове поняття “пухнастий песик”, проявляється у світі? Тонка лінія - зв’язок між словами та її візуалізацією у DALL-E реалізується іншою моделлю від OpenAI, яка називається CLIP (Contrastive Language-Image Pre-training — контрастне попереднє навчання на основі мови та зображень).

Процес навчання CLIP будується на аналізі сотні мільйонів зображень і пов'язаних з ними підписів, кожен сегмент тексту вивчається на відповідність до зображення, щоб побудувати зв’язки і асоціативний ряд з майбутніми промпт запитами. Тож CLIP не намагається передбачити, що в майбутнього в нього запитає користувач, CLIP вивчає наскільки певні промпти пов’язані з зображеннями. Ця контрастна мета дозволяє вивчити зв’язок між текстом і картинкою, а не намагатись передбачати те що модель не може знати і розуміти.

Основні принципи навчання CLIP починаються з пропускання усіх зображень і пов’язаних з ними підписів через енкодери, що розміщені в m-вимірному просторі.

Потім обчислюється косинусна подібність кожної пари зображення=текст.

Мета навчання полягає в тому, щоб одночасно максимізувати косинуса подібність між N правильними парами закодованих зображень/підписів і мінімізувати косинуса подібність між N2 - N неправильними парами закодованих зображень/підписів.

2.2.3. Принцип GLIDE моделі

До створення GLIDE дифузійні моделі генерували зображення за “мітками класу”, але тепер з’явилася можливість пов’язувати зображення з текстовою інформацією і створювати масивні по наповненню композиції. Це досяглося шляхом додавання на кожному етапі дифузіїї текстових токенів. Тепер при кодуванні тексту трансформер звертає увагу на кожен текстовий токен. Спочатку GIDE використовує шум, зумовлений текстом, але розробники GLIDE додатково

керували процесом, додаючи градієнт відповідності зображення до зображення. Фактично модель бере початкове зображення з шумом і переміщує його в передбачуваному напрямку де це зображення матиме максимальну відповідність до тексту. Також основна відмінність моделі GLIDE є Classifier-Free Guidance, коли модель вчиться на частині навчальних прикладів без тексту і з текстовим описом. Це дозволяє моделі генерувати зображення як з описом так і без, щоб результат був більш точним.

Тож якщо візуально об’єднати всі процеси, то виглядатиме це наступним чином (рис.1.5.14) :

Процес починається з перетворення текстового промпту енкодером CLIP у простір представлення.

Наступний етап це Diffusion prior map перетворюють текстове кодування CLIP у відповідне зображення.

Фінал, модифікована модель генерації GLIDE за допомогою зворотної дифузії перетворює представлення зі свого простору у простір зображення, створюючи одне з багатьох можливих зображень, яке співпадає з описом.

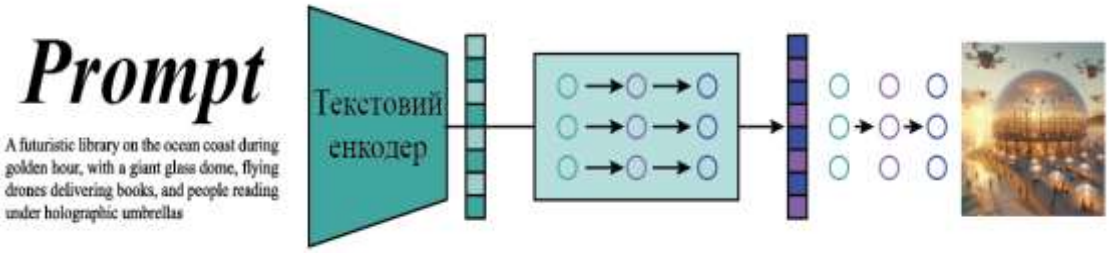


рис.1.5.14. Повний процес генерації зображення текст-в-зображення

2.3. Популярні графічні дифузійні моделі

2.3.1. Stable Diffusion

Stable Diffusion - це відкрита модель глибокого навчання, яка працює за принципом “текст-в-зображення”. Випущена в 2022 році компанією Stability AI на основі відкритий латентних дифузійних технологіях (LDM). Основна сфера застосування - це генерація детальних зображень на базі текстових підказок (prompt запитів), хоча застосовувати її генерації “зображення-зображення”, заповнення або відновлення пустих чи пошкоджених елементів в зображенні, або

доповнення зображення базуючись на тексті чи на оточуючому навколо бажаного розширення мультимедійному контенті.

Завдяки доступності кода і не великому об’єму моделі її можна використовувати на більшості обладнання. Це дозволяє досліджувати системи з метою візуалізації творчих ідей. І хоча з 2022 року моделі вже настільки покращились, що створені ними зображення і мультимедіа все частіше викликають страх, що зовсім скоро неможливо буде відрізнити згенероване від реального, люди все ще використовують цей інструмент для розваг і продовжують вивчати цей інструмент. Саме про вивчення цього інструменту і буде в цьому розділі.

Штучний інтелект Stable Diffusion використовує LDM розроблену CompVis, ця архітектура навчалась додавати шум до зображення і видаляти його, цей процес є послідовністю шумо понижувальних автокодувальників. Автокодувальник VAE фіксує семантичне значення стискаючи зображення в латентний простір. Процесом прямої дифузії застосовується Гаусівський шум до латентного представлення. Щоб повернути процес дифузії назад декодер U-Net видаляє шум з латентних векторів. Заключним етапом VAE декодер реконструє з очищеного від шуму латентного представлення унікальне зображення. Результат видаленого шуму залежить від текстового запиту за допомогою шарів поверхневою уваги і за допомогою текстового кодувальника Clip ViT-L/14.

Оскільки принцип навчання дифузійних моделей детально був описаний в попередньому розділі, то перейду одразу до текстового кодувальника, про який згадано не було. Процес зашумлення і декодування той же. А вже текстовий кодувальник приймає підказки (промпти) від користувачів та виводять латентні описи які об’єднуються та проектуються у правильну розмірність і зливаються з вхідними даними декодера. Це і забезпечує генерацію зображень текст-в-зображення. Вектори точками відтворення для декодера, який знижує шум відповідно до вхідних даних на кожному часовому кроці, орієнтуючись на те що прописав користувач чату і як це передав текстовий кодувальник.

Модель навчана працювати в малій роздільній здатності 512x512 пікселів, через це перші результати на початку роботи моделі були розмитими або

поганою якості, для багатьох це незадовільний результат, оскільки сучасні монітори і камери виходять далеко за межі цих параметрів і привчили користувачів бути більш вимогливими до того, що вони бачать очима. Через що згодом було придумано ШІ основна спеціалізація яких це додавання додаткових суміжних пікселів до вже існуючих, щоб розширити роздільну здатність і покращити візуально якість. Іноді надмірне збільшення розміру зображення веде до появи артефактів.

Рекомендується запускати модель із 10 ГБ або більше VRAM, однак користувачі з меншою кількістю VRAM можуть вибрати точність float16 замість стандартної float32, щоб забезпечити продуктивність моделі з меншим використанням відеопам'яті.

Також одним з мінусів є недостатні та низькоякісні навчальні дані щодо кінцівок людини, які призводять до анатомічних аномалій при генерації людей. Згенеровані кінцівки, руки та обличчя часто мають нереалістичний вигляд, дивні пропорці чи неправильну кількість об’єктів. На старті моделі відрізнити згенероване зображення можна було в 99% просто подивившись на кінцівки будь якого живого організма.

Найвідомішими сервісами є :

DreamStudio: Сервіс від Stability AI, розробника Stable Diffusion, що надає простий інтерфейс для генерації.

Hugging Face: Надає різні версії та варіації моделі для розгортання і експериментів.

Automatic1111/ComfyUI: Популярні локальні інтерфейси, які дають повний контроль над моделлю.

SD активно використовують художники для генерації ескізів чи надають фотографію вже намальованого макета, щоб модель доповнила його на свій розсуд. Хоча є і інший табір творців, які бойкотують використання штучного інтелекта і займаються охотою на штучний інтелект і тих хто його розповсюджує. Оскільки багато митців помічали явний вплив своїх робіт на стиль деяких сервісів, які надавали послуги генерації зображень на базі моделі Stable Diffusion. Деякі митці можуть вести цілі сторінки і монетизувати їх

виключно за рахунок ШІ генерації. Люди створюють не справжніх особистостей і ведуть від їх імені блоги про життя.

Дуже часто малий бізнес формує собі маркетингові стратегії чи рекламу за допомогою текст-в-зображення генерацій. Вони можуть спочатку піти до чату з ШІ, витратити час на підготовку декількох варіантів реклами і вже після звернутися до дизайнера з метою відмалювати макет відповідно до потреб поліграфії. Оскільки моделі неспроможні створювати складні векторні зображення, які потрібні для якісного друку в типографіях. Прості згенеровані зображення, можуть підійти тільки для особистого використання і друку хіба що постерів чи малих карток.

2.2.2. DALL-E

Ця модель просунулась далі у створенні високоякісних реалістичних зображень з prompt-запитів. DALL-E це один чергова мовна генеративна модель яка використовує процес дифузії архітектури кодера-декодера для навчання. Для обробки в DALL-E використовується варіаційний автокодер з векторним квантуванням (VQ-VAE) щоб обробляти інформацію високої вимірності про зображення порівняно з текстом. DALL-E 2 ж має значне покращення у вигляді кодування текстового опису в мовній моделі OpenAI під назвою Contrastive Language-Image Pre-training (Clip). Контрастне попереднє навчання мови і зображень, якщо перекладати дослівно, є багатовимірним вектором для текстового та візуального вмісту. Потім модель це декодує назад у зображення за допомогою дифузійної моделі.

На відміну ж від попередніх двох версій DALL-E 3 використовує синтетичне підписування (Synthetic Captioning), це дозволяє детально описувати основні об’єкти, беручи до уваги також оточуюче середовище і фон, контекст сцени і зв’язки між об’єктами. Це дозволяє працювати зі складними, більш детальними і більш заплутаними промптами. Під час створення зображення DALL-E 3 використовує ці детальні описи для відтворення візуальних ефектів, які відповідають опису. Для прикладу я реалізую однаковий промпт на двох різних моделях. Обрано промпт “A futuristic library on the ocean coast during

golden hour, with a giant glass dome, flying drones delivering books, and people reading under holographic umbrellas", DALL-E 3 (Додаток А) демонструє хорошу деталізацію, добре видно деталі і елементи, видно людей і дрібні деталі, такі як елементи одягу і аксесуари, видно інтер'єр і можна навіть прослідкувати патерни поведінки відвідувачів. DALL-E 2 (Додаток Б) у свою чергу демонструє повну відсутність деталей, людей майже не розібрати, це більше схоже на розводи, океан схожий на ледве синю пляму на фоні сміття і багнюки, бібліотеки згенеровано не було, хоча це наймасштабніший об'єкт, але модель просто не змогла з'єднати всі елементи в одну картину.

2.2.3. Midjourney

Це закрита модель яка працює по такому самому принципу як дифузійні Stable Diffusion або DELL-E з багатьма відмінностями, але у відкритому доступі інформації по цій моделі немає, через приватність. Тому вся інформація про цю модель буде наводитись виключно по порівнянню процесів, переваг, недолік і результатів які можна отримати працюючи з цією моделлю.

Доступ до Midjorney на відміну від інших моделей надається через Discord сервер, коли як для DALL-E це безліч сервісів : OpenAI (ChatGPT Plus) і Microsoft Designer (Bing Image Creator). Stable Diffusion це відкритий код, який можна локально запустити на власному комп'ютері, або використовувати через безліч сервісів з різною моделлю підписки. В Discord сервері працює система підписок, тож безкоштовне використання досить обмежене.

Інтерфейс діскорда може здатися незрозумілим тим хто не звик користуватись цією платформою, оскільки Discord це програма, яка не була розроблена для подобного типу послуг, а хоча інструментарій для модифікацій і налаштування серверу хоч і є обширний, все ж сильно обмежується інтерфейсом системи чатів і каналів Discord. В свою ж чергу інтерфейс інших двох моделей існує в конкурентному середовищі, де власники платформ і сервісів зацікавленні в розробці унікальної і зручної моделі взаємодії з їх ресурсом. Кожен з них може налаштувати власний додаток, сайт, чат бот чи веб інтерфейс найбільш вдалим для роботи з цим типом завдань чином. Якщо ж розглядати варіанти коли Stable Diffusion запускається локально власними силами, то тут вже все обмежується

знаннями і навичками людини яка налаштовує цей код під себе. Це абсолютно унікальний досвід який для кожного унікальний і розглядати його як інтерфейс неможливо.

Візуальний стиль Midjorney відомий високоякісними, ефектними та художніми зображеннями. Чудово підходить для концепт-арту та ілюстрацій. В випадку з DALL-E 3, все залежить від окремих сервісів і того що вони пропонують, якщо говорити наприклад за ChatGPT, то в роботі з ним досить часто прослідковується досить однакова і впізнавана палітра цього сервісу, дуже часто зображення згенеровані саме в ньому легко ідентифікувати за трошки коричневою гамою.

Якість це однозначно найсильніша сторона MidJorney. Створити зображення яке по якості буде виглядати “4к” звична справа для роботи з цією моделлю, особливо якщо мова йде про платну підписку. DELL-E звісно також пропонує досить високої роздільної здатності зображення, але навіть якщо генерувати зображення в fullhd розширені, все одно помітно, що навіть для 1920x1080 зображення застосовувався upscale, який не зміг дотягнути різкість і якість пікселів до потрібного рівня.

В можливості редагувати і гнучкості, на жаль, похвалити так само як в якості цю модель не вийде. Вона однозначно уступає іншим двом. Модель хоч і дає можливість обрати з декількох варіантів ті що сподобались найбільше і продовжити працювати з ними і пропонує нові варіанти сході на ті. все ж можливість внести зміни в промтп чи дописати новий доповнюючий промпт дає відчути контроль на процес в будь який момент часу роботи з моделлю.

Як висновок модель добре підійде для створення високоякісних зображень, людям які часто співпрацюють з створенням якісного мультимедійного контенту, але для новачків які хочуть тут і зараз зробити одну дві картинки краще підійдуть безкоштовні моделі DALL-E, які бувають як і частково безкоштовні так і обмежено безкоштовні. А для ентузіастів які працюють з цим типом завдань постійно, мають безліч часу і знання по роботі з кодом, і звісно не обмежені апаратним забезпеченням, підійде Stable Diffusion, адже є можливість навчити нейронну мережу так як треба, під ті задачі які треба і мати можливість

створювати стільки контенту скільки потребуватиме користувач.

3. DIFFUSION ВІДЕО ТА АУДІО ГЕНЕРАЦІЯ

3.1. Модель SORA та принципи роботи відео генерації

Sora – модель генерації відео за допомогою дифузії. За велику кількість кроків результат поступово змінюється з видаленням шуму, як це раніше описувалось в зображеннях, процес подібний до зворотної дифузії. OpenAi змогли вирішити одразу кілька складних проблем, одна з яких це збереження послідовності об’єкта, коли той зникає з кадру. Ще одна проблема яку Sora подолала це генерації сцен з кількома персонажами, кожен з яких має свої руки і не трансформується в одне ціле, а ще й взаємодіють з навколишнім середовищем. Для цього модель ШІ має не тільки розуміти з якими об’єктами вона працює, що описує користувач в промпті, а й те як працює фізичний світ і закони фізики цього світу. Подібно до ChatGPT використовується архітектура трансформера, тож ця архітектура продемонструвала свою неабияку ефективність вже не тільки в роботі з генерацією зображень. а й в роботі з генерацією відео. Зображення та відео представлені у вигляді сукупності менших одиниць даних “патчів”. Ще одна проблема яку OpenAI змогли вирішити це навчити дифузійні трансформери на широкому спектрі даних з різною тривалістю, роздільною здатністю і співвідношенням сторін.

Рис. 3.1. Процес видалення шуму для генерації графіки

Sora послуговується тим самим текстовим енкодером, що і DALL-E 3, принцип його роботи описувався в принципах роботи цієї моделі. Цікавим є й ще одне запозичення з DALL-E - це описування (re-captioning) Ця архітектура покращує точність відповідності тексту та загальну якість відео. Натхненна також використанням токенів у великих мовних моделях (LLM), Sora використовує представлення візуальних даних на основі "патчів" (patch-based

representation). Такий підхід ефективно уніфікує різноманітні модальності візуальних даних, сприяючи масштабованому та ефективному навчанню генеративних моделей. Патчі дозволяють моделі з легкістю обробляти різні типи відео та зображень.

Щоб перетворити відео на патчі, Sora спочатку стискає вхідні дані у латентний простір меншої вимірності, зберігаючи просторову і часову інформацію. Це стиснення відбувається за допомогою нейронної мережі стиснення відео, яка зменшує розмірність візуальних даних, зберігаючи їхні основні характеристики. Потім стиснене представлення розкладається на просторово-часові патчі, які слугують токенами-трансформерами для архітектури дифузійного трансформера Sora.

Основна перевага тренування моделі заключається в тренуванні на даних в оригінальному розмірі, а не в обрізаному. Цей гарантує гнучкість вибірки, покращену композицію кадру та краще розуміння мови. Тренуючись на відео з їхнім вихідним співвідношенням сторін, Sora досягає кращої композиції та кадрування, що призводить до генерації високоякісного відео.

Одним з цікавих інструментів Sora це створення цілих відео зі статичних зображень і можливість “оживляти” будь які знімки, як за допомогою промптів так і просто довірившись моделі самій обирати сценарій і доповнити зображення тією кількістю рухів якою вона була заздалегідь навчена. Це можуть бути і старі фотографії і сімейні дитячі фотографії, можна надати рухів намальованим персонажам, можливостей для застосування у подібної технології безліч. Сучасні техніки анімації зображень використовують методи візуалізації на основі нейронних мереж для створення реалістичної анімації. Однак досягнення точної та керованої анімації зображення за допомогою тексту залишається складним завданням, особливо для зображень із відкритих доменів, зроблених у різноманітних реальних умовах. Такі моделі як AnimateDiff, AnimateAnything тощо, також продемонстрували багатообіцяючі результати для анімації статичних зображень.

Детальне і точне моделювання складного простору другорядних об’єктів для Sora залишається слабким місцем, як і розуміння причинно-наслідкових

зв’язків і поведінки фізики об’єктів. Дуже часто можна зрозуміти, що перед тобою відео згенероване штучним інтелектом просто придивившись до текстури об’єктів, оскільки часто вона мерехтить. Часто живі організми такі як собаки чи коти ведуть себе нетипово, або їх тіла вигинаються в ту сторону, тіні можуть реагувати на світло не типово, і об’єкти будуть давати тінь не під тим кутом, людина може писати щось на аркуші але текст чи написані цифри будуть з’являтись не від ручки, а просто з'являтись цілими словами, вода яка виливається з бутилки ллється під дивним кутом і безлічі подібних прикладів, які все ще залишаються для моделі непідйомною задачею. Іноді модель може переплутати частини запитів промпта надати властивості різним об’єктам. І остання проблема яку варто зауважити це сценарії і чіткі описи подій по часу, дуже часто можна витратити години спроб в тому щоб отримати від ШІ дотримуватись сценарію промпта, щоб кожна подія відбулася чітко по запланованому графіку подій.

3.2. Генерація відео та звуку з моделлю Veo 3 від Google

Google Veo це один з передових відео та аудіо генераторів. Це можливо як і за допомогою промптів (текст-у-відео) так і за допомогою зображень (зображення-у-відео). Як і у випадку з Sora це вже не просто робота однієї дифузійної моделі для виконання конкретних завдань. Цей сервіс послуговується одразу декількома нейронними мережами і моделями, як командою для вирішення потреб користувачів. Представлена ця модель була у 2024 році і одразу набула популярності через якість і реалістичність генерованого контенту.

Архітектура, заснована на гібридному дифузійно-трансформерному підході, обробляє відеопотік, включаючи звукові дані, як послідовність просторово-часових «патчів» в єдиному латентному просторі. На фундаментальному рівні Veo 3 спирається на принципи латентної дифузії, додаючи сюди ще час. Що і логічно, оскільки в дифузійних моделях які працюють тільки з зображеннями час не потрібен, це просто картинка яка існує тут зараз і в статиці. Тому перемінна “час” дозволяє моделі спільно обробляти патерни в часі, просторі та звуці, забезпечуючи когерентність і безперервність

сцен упродовж усього відео. Під час навчання на великих наборах даних, що містять відео, аудіо та текстові метадані, трансформерна частина моделі вчиться передбачати фізичні взаємодії та зберігати візуальну й наративну єдність, що призводить до створення більш реалістичного контенту.

Одним із ключових архітектурних нововведень Veo 3 є спільне застосування дифузійного процесу до латентних уявлень як відео, так і аудіо. На кожному кроці процесу шумозаглушення механізм уваги трансформера працює з уніфікованою послідовністю токенів, що представляють як просторово-часові відеопатчі, так і часову аудіо інформацію. Це дозволяє моделі вивчати фундаментальні статистичні кореляції між звуком та зображенням як основну функцію навчання, а не як вторинне завдання зіставлення. Таким чином, Veo 3 від початку вчиться синхронізувати рухи губ з мовленням і співвідносити звукові ефекти з подіями, що відбуваються на екрані. Завдяки такому підходу, Veo 3 пропонує високий рівень контролю над процесом генерації, дозволяючи користувачам не лише задавати текстові описи, а й використовувати опорні зображення чи відеокліпи для забезпечення стилістичної відповідності. Модель здатна генерувати відео високої роздільної здатності, до 4K, з плавним рухом камери (включно з панорамуванням, масштабуванням і рухами візка), що значно розширює творчі можливості. Здатність Veo 3 забезпечувати високу деталізацію, реалістичне освітлення та фізично достовірний рух робить його важливим інструментом для професійної відеопродукції. Veo 3 є значним кроком уперед у галузі генеративного ШІ, долаючи обмеження ранніх моделей, які часто стикаються з проблемами часової когерентності та відсутності звуку. Його мультимодальна архітектура, що об'єднує дифузійні процеси з трансформерними механізмами, дозволяє створювати не просто послідовність кадрів, а цілісні аудіовізуальні твори з високим рівнем реалізму та художньої виразності.

3.3. Генерація аудіо за допомогою дифузійних моделей

3.3.1. Способи генерації аудіо

Суть генерації аудіо полягає в процесі автоматичного синтезу нових хвильових форм за допомогою моделей глибокого навчання. Це зазвичай виконується за двома абсолютно різними способами : символічний і на рівні

звукової хвилі.

За символічним генеруванням мається на увазі створення представлення аудіо з використанням символів які перетворюються в звукову хвилю. Це найпростіший в використанні метод, але за спрощенням криється складність передати усі тонкості звуків, оскільки символи це далеко не ідеальний варіант передачі звуку.

Генерація на рівні звукової, є більш складним методом, оскільки полягає у безпосередньому створенні звукової хвилі моделлю, а це велика кількість значень за короткий час. Проте основною перевагою цього метода є точність та детальність генерованих звуків.

Також можна розділити аудіо генеративні ШІ ще за двумя признаками. Одні навчаються лише на аудіоданих і можуть генерувати нові зразки без будь-якого додаткового вводу, їх називають “безумовними”. Інша модель – умовна, навчається на парах аудіоданих та вказівках (промптах) або додатковій інформації жанр, текст пісні, інструмент, тощо. Під час виведення ця інформація для керування може бути використана для спрямування генерації нових зразків аудіо, що відповідають бажаним характеристикам.

Необхідно враховувати численні компроміси під час генерації аудіо на рівні звукової хвилі (waveform level). Щоб згенерувати одну секунду високоякісного стерео аудіо з частотою 48 кГц, потрібно створити 96 000 значень, що можна порівняти за розміром із зображенням середньої роздільної здатності. Якщо мета полягає в генерації цілої пісні (сотні секунд) із підтримкою високої якості та прийнятної швидкості генерації, це завдання стає набагато складнішим.

3.3.2. Принцип роботи і навчання

За принципом роботи аудіо дифузійних моделей стоїть вже звичний з графічних DDIM чи DDPM і підсумкова цільова функція для навчання моделі розраховується з випадковими рівнями шуму t, які вибираються з рівномірного розподілу. Після навчання генерація нових даних у DDPM відбувається прямолінійно, спочатку додається випадковий шуму, який відповідає

конкретному, заздалегідь визначеному нормальному розподілу. Потім модель використовується потрібну кількість разів. На кожному кроці вона використовує оцінені середні шуми з нормальних розподілів. Це дозволяє крок за кроком отримати послідовність, що веде до початкових даних, які і будуть в нашому випадку згенерованим результатом. Це все відбувається не випадковим чином а з конкретною послідовністю.

Що вже слід виділити як нове і те що відноситься виключно до аудіо дифузійних моделей. Це V-дифузія, або точніше кажучи дифузія з V-цільовою функцією. Це дифузійний метод який виник під впливом DDIM, який тренується з використанням неперервного значення σ_t, що знаходиться в діапазоні від 0 до 1 включно. Цей метод виявився найкращим для різноманітних завдань, пов'язаних з аудіо. У V-дифузії, якщо параметр σ_tдорівнює нулю, то вихідне значення x_{σ_t} являє собою точку даних x взяту з розподілу даних. Якщо параметри σ_t дорівнює одиниці, то це значення перетворюється на гаусівський шум. У методі DDIM слід обрати, чи використовувати модель для передбачення вихідних даних x_0, чи для передбачення шуму ϵ_t. Однак у випадку V-дифузії оцінюється значення "швидкості" v_{σ_t}, на основі якого можна відновити як вихідні дані x_0, так і шум ϵ_{σ_t}.

Якщо в дифузійних моделях для роботи з генерацією зображень використовується бібліотека PyTorch, то в випадку з аудіо використовується бібліотека audio-encoders-pytorch. Компонент автокодувальника має структуру, подібну до U-Net, з кількома змінами:

Щоб зробити його автокодувальником, не використовуються пропускні з'єднання (skip connections), блоки умовного форматування (conditioning blocks), і не застосовуються блоки уваги (attention blocks), щоб забезпечити його універсальність для будь-якої довжини вхідної послідовності.

AEP містить як кодувальники (encoders), так і декодувальники (decoders), а також набір "вузьких місць" (bottlenecks), які можна використовувати для нормалізації латентного простору, а саме:

Варіаційне вузьке місце (variational bottleneck) у стилі VAE.

Просте tanh-вузьке місце (simple tanh bottleneck).

Квантизуюче вузьке місце (quantizer bottleneck), подібне до того, що було запропоновано VQ-VAE.

Крім того, використовується два кодувальники, які кодують спектрограми поканально в одновимірний латентний простір, а саме: ME1d (кодувальник лише спектрограми амплітуд) або MelE1d (кодувальник мел-спектрограми). Обидва сумісні з різними варіантами "вузьких місць".

Генератор дифузії бере як вхідний матеріал високоякісне стерео аудіо з наборів даних, яке потім спотворюється до випадкового рівня шуму на основі обраного методу дифузії. За допомогою U-Net генератор передбачає результат, який може бути як вхідними даними, очищеними від шуму, так і значенням, яке використовується для їх розрахунку, залежно від типу застосовуваного методу дифузії. Рівень шуму зазвичай називають часом або σ, він подається мережі як умова у вигляді закодованого вектора ознак. Це допомагає мережі зрозуміти, скільки шуму потрібно видалити з вхідних даних. Для цього дифузійного генератора не використовуються ані блокування вбудовування, ані блоки перехресної уваги.

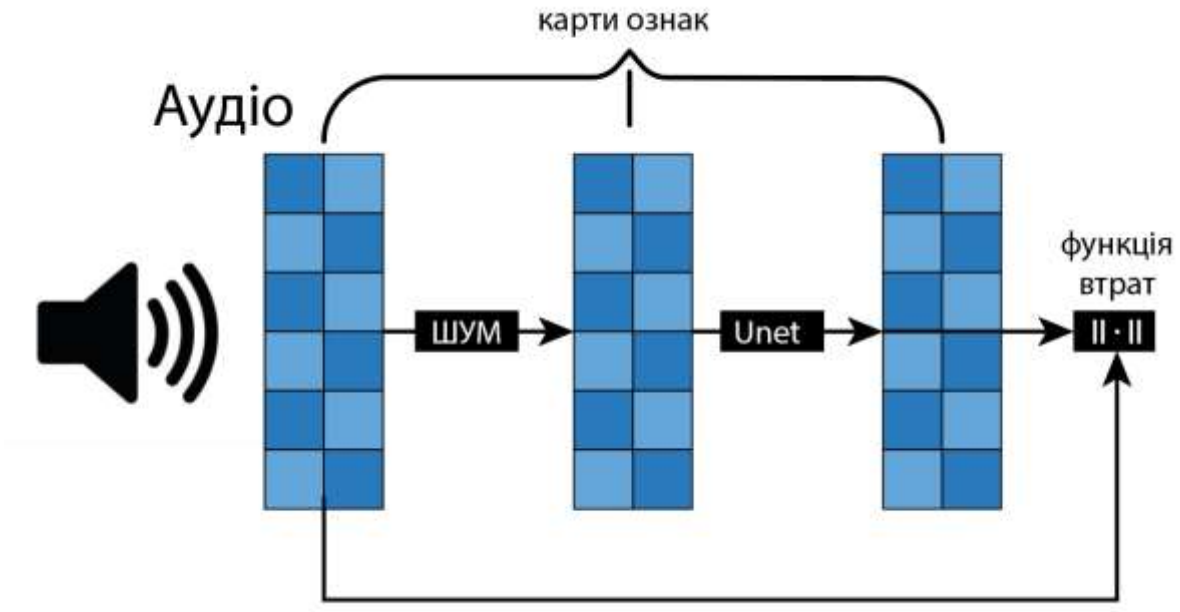


Рис. 3.2. Схема тренування аудіо дифузійної моделі

Під час генерації береться вибірка випадкового вектора, що має таку ж форму, як і навчальний зразок аудіо. Потім U-net ітеративно викликається зі змінним рівнем шуму, щоб згенерувати новий, правдоподібний зразок із

розподілу даних.

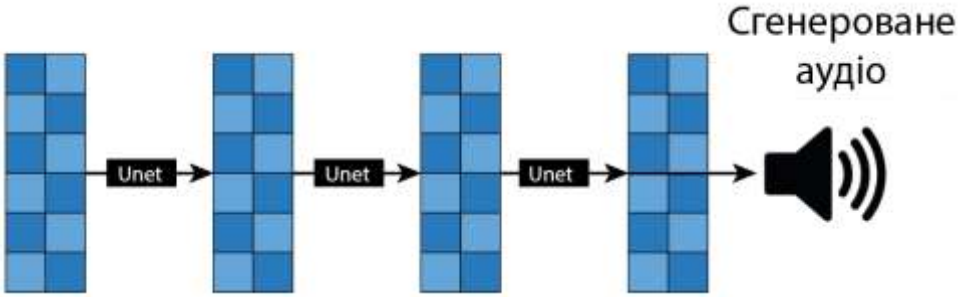


Рис. 3.3. Схема зворотної дифузії

Модель одразу продемонструвала хороший результат із DDPM дифузією, але потребує приблизно 200 кроків вибірки з базовим семплером під час інференсу, щоб згенерувати прийнятні результати. Якщо використовувати k-diffusion та відповідну конфігурацію гіпер параметрів, модель намагається згенерувати високоякісне аудіо з високим динамічним діапазоном навіть за допомогою вдосконалених методів семплювання. При належній нормалізації гучності можна отримати хороші результати, використовуючи приблизно в 10 разів менше кроків. Дифузія Zen DDPM, але втратить здатність створювати зразки, які відрізняються рівнем гучності. Виявлено, що v-дифузія є найбільш сприятливим методом, стійким до не нормалізованого аудіо та все ще досить швидким для семплювання: близько 50 кроків семплювання дають хороші результати за допомогою семплювання DDIM. Крім того, v- дифузія вимагає налаштування меншої кількості гіпер параметрів, забезпечуючи хороший баланс між простотою, швидкістю та якістю зразка.

Незалежно від використаного методу дифузії, ця модель без будь-яких доповнень намагається згенерувати лише кілька секунд звуку. Якщо на вхід мережі подається сирий сигнал (waveform), початкові згорткові блоки U-Net будуть змушені обробляти величезні зразки. Наприклад, навіть одна секунда високоякісного аудіо з частотою 48 кГц вимагає обробки 48 000 значень першим згортковим блоком. Це може стати проблемою швидкодії, якщо аудіо не буде достатньо швидко зменшено (downsampled) в U-Net, оскільки ця неефективність накопичуватиметься протягом кількості кроків семплювання процесу дифузії. Окрім того, якщо використовуються блоки уваги (attention blocks), нам

доведеться достатньо зменшити вибірку (downsample), щоб переконатися, що кількість часових кроків (timesteps) знаходиться в діапазоні 1024 або 2048 значень. Перевищення цього значно сповільнить самостійну увагу (self-attention) через обчислювальну складність для довжини послідовності. Отже, для задоволення цих критеріїв потрібне значне зменшення вибірки (downsampling) для довгих аудіо семплів. Щоб пом'якшити згадані раніше проблеми, розробники моделей досліджують використання різних методів та аудіо перетворень для конвертації вихідного сирого аудіосигналу в представлення, яке зменшує часовий вимір в обмін на додаткові канали.

Перше перетворення — це патчинг (patching), спочатку запропонований для доменної області зображень у. Слід адаптувати патчинг до 1D домену, де ідея полягає в групуванні послідовних часових кроків у фрагменти (chunks), які потім будуть транспоновані в канали. Враховуючи розмір патча p, довжина t зменшується до $\frac{t}{p}$, тоді як кількість каналів збільшується до $c \cdot p$. Наприкінці обробки U-Net канали розгруповуються (unchunked) назад до повної довжини. Було виявлено, що патчинг забезпечує значне прискорення, майже в p разів для p=2,4,8,16,32,..., що дозволяє тренувати моделі з набагато довшими аудіо джерелами. Однак, навіть якщо якість генерації аудіо майже відповідає версії без патчингу, чутний аліасинг (aliasing) присутній при всіх факторах. Цей недолік, ймовірно, пов'язаний із повторюваним процесом розгрупування, який матиме повторювану структуру, створюючи високочастотну синусоїду в сигналі. Крім того, було виявили, що патчинг із p>64)починає погіршувати якість, ймовірно, через певні обмеження пропускної здатності в розмірності каналів. Слід можемо розглядати патчинг як детермінований процес автокодування з коефіцієнтом зменшення вибірки (downsampling factor) p.

Другим перетворенням є раніше представлена STFT (Short-Time Fourier Transform — Короткочасне перетворення Фур'є). Використовуються загальні налаштування: 1024 для кількості FFT (num fft) та довжини вікна (window length) з кроком (hop size) 256.

Обгортаючи U-Net функціями STFT та iSTFT (обернене STFT), перетворення зменшує довжину аудіо в 1024 рази, одночасно пропорційно

збільшуючи кількість каналів. STFT реалізовано за допомогою швидкого перетворення Фур'є (Fast-Fourier Transform), отже, його застосування є ефективним. Нормалізація на спектрограмі не потрібна, оскільки втрати дифузії все одно застосовуватимуться до реконструйованої хвилі.

Цей метод забезпечує значне прискорення завдяки великому зменшенню вибірки (downsampling), але, подібно до патчингу, страждає від погіршення якості порівняно з представленням сирої хвилі. Відчутний шум присутній у згенерованих даних як при перетворенні в формат "величина+фаза" (magnitude+phase), так і при використанні "дійсна+комплексна" частини (real+complex).

Цікаво роглянути роботу так званих навченого перетворення (learned transformation) з єдиним згортковим (convolutional) та оберненим згортковим (transposed-convolutional) блоком на початку та, відповідно, в кінці U-Net. Суть роботи перетворення полягає у використанні великого розміру ядра та кроку, який дорівнює 64. Це зменшить вибірку вихідного сигналу за один крок, жертвуючи невеликою кількістю швидкості порівняно з детермінованим патчингом або STFT, реалізованим через FFT.

Однак, оскільки це згортковий метод, є можливість вибрати кількість каналів і збільшити її до більшого значення, наприклад, 128, що вдвічі перевищує розмір ядра та крок, ніж використовувалося під час патчингу, дозволяє бути більш стійкимм до артефактів. Водночас є можливість використовувати ідеї з STFT і мати великі вікна з перекриттям та навченими ядрами замість фіксованих синусоїдальних/косинусоїдальних хвиль (наприклад, розмір ядра 128, крок 64, 64 канали, з доповненням (padding) для збереження розмірності), що може допомогти подолати аліасинг. Цей метод забезпечує найкращий компроміс між якістю та швидкістю серед методів попереднього перетворення аудіо.

3.3.3. Відомі представники та сервіси для дифузійних моделей для генерації аудіо

Оскільки неймовірний вибух дифузійних моделей вже подарував всплеск мультимедійних генеративних моделей. То абсолютно закономірно, що і інші ланки мультимедійного мистецтва висловили бажання долучитись до генеративних ШІ, тому попит невпинно зростав останнє десятиріччя.

Моделей для роботи зі звуком безліч, оскільки звукові хвилі це не просто набори пікселів, чи генерування тих ж картинок в патчі з подальшим перетворення це у відео, хоча звісно це також не прості процеси. Але навчання на картинках все ж легше ніж пояснити усю багатогранність звуків для штучного інтелекта і нейронних мереж. Спробую охопити основні моделі:

Long: латентна дифузійна модель для генерації музики за текстовою умовою, яка здатна генерувати аудіо з розширеним контекстом у кілька хвилин на частоті 48 кГц, з акцентом на довжину контексту та структуру (~857 мільйонів параметрів).

Crisp: дифузійна модель для генерації аудіо за текстовою умовою з контекстом у десятки секунд на частоті 48 кГц, з акцентом на простоту та високоякісні звукові хвилі (~419 мільйонів параметрів).

Upsampler: дифузійна модель для збільшення частоти дискретизації музики з 3 кГц до 48 кГц (~238 мільйонів параметрів).

Vocoder: дифузійна модель для реконструкції звукових хвиль на 48 кГц з 80-канальних мел-спектрограм, зі змінною довжиною вхідних даних (~178 мільйонів параметрів).

Однією з піонерських моделей генерації на рівні звукової хвилі є WaveNet (2016) — повністю згорткова архітектура, що використовує розширені (dilated) згортки з різними коефіцієнтами розширення для захоплення широкого контексту. Вона здатна синтезувати кілька секунд як мовлення, так і класичної фортепіанної музики з частотою 16 кГц.

Jukebox (2020) використовує кілька квантованих автокодерів для дискретизації звуків у 3 різних роздільностях, після чого каскад трансформерних моделей-апсемплерів (upsampler) генерує квантовані представлення авторегресійно. Jukebox може генерувати музику з частотою 44 кГц, обумовлену текстом, виконавцями та жанрами. Каскад трансформерів жертвує швидкістю генерації заради структури та якості.

AudioLM (2022) використовує (залишковий) квантований автокодер для стиснення звукової хвилі в дискретні токени та семантичний кодер. Потім каскад трансформерних декодерів (семантичний, грубий, тонкий) використовується для

генерації 16 кГц аудіопродовжень зверху вниз на основі семантичного представлення.

Musika (2022) навчає набір 1D згорткових авто кодерів для стиснення спектрограм логарифмічної амплітуди та вокодер для реконструкції як фази, так і амплітуди зі стисненого представлення. Використовуючи 2D GAN-дискримінатор, навчений на послідовних фрагментах аудіо, цей процес використовується авторегресійно для генерації довших послідовностей аудіо з частотою 44 кГц. Цей метод має обмежену довжину контексту, але є дуже ефективним завдяки 1D структурі згорток.

Riffusion (2022) донавчає модель Stable Diffusion на фрагментах мел-спектрограм по 5 секунд з частотою 44 кГц, і використовує перенесення стилю для генерації кількох когерентних, склеєних зображень, обумовлюючи це текстовим описом пісні. Цей метод має обмежену довжину контексту (5 секунд) і жертвує швидкістю через велику 2D-архітектуру, але працює напрочуд добре, враховуючи, що оригінальна модель навчена на зображеннях, а не на аудіо.

Dance diffusion це родина аудіо-генеративних моделей машинного навчання створені організацією, яка назвала своєю місією розробити відкриту для використання аудіо-генеративну модель для продюсерів, музикантів і простих користувачів які цікавляться створенням музики. Вони є частиною Stability AI і назва цій організації Harmonai.

Наразі існує 6 загальнодоступних моделей Dance Diffusion, кожна з яких навчена на різних наборах аудіофайлів:

1. glitch-440k

   Навчена на кліпах, наданих glitch.cool
2. jmann-small-190k

   Навчена на невеликій підбірці кліпів із проекту Джонатана Манна «Пісня щодня» (Song A Day)
3. jmam-large-580k

   Навчена на великій підбірці кліпів із проекту Джонатана Манна «Пісня щодня» (Song A Day)
4. maestro-150k

Навчена на підбірці фортепіанних кліпів із набору даних MAESTRO

5. unlocked-250k

   Навчена на кліпах із набору даних Unlocked Recordings

6. honk-140k

   Навчена на записах канадського гусака, отриманих через xeno-canto

Оскільки дані, на яких навчається дифузійна модель — і тому вчиться їх відновлювати — впливають на тип даних, які вона згодом генерує, аудіо приклади, створені, наприклад, моделлю maestro-150k, завжди звучатимуть як фортепіанна музика, а не як музика гітари чи труби.

Зак Еванс, творець Dance Diffusion, випустив блокнот Google Colab для відкритого бета-доступу до вищезазначених моделей Dance Diffusion. Еванс також написав ще один блокнот Colab, де ви можете додатково налаштувати або кастомізувати модель Dance Diffusion на власному наборі даних для більшого контролю над згенерованими аудіокліпами.

3.3.4. Покращення записаного голосу, або мікрофону за допомогою Diffusion моделей

На сьогодні існує велика кількість методів по зменшенню затримки в моделях текст-в-голос (TTS), які основані на дифузії, це “метод без тренування” і “метод з тренуванням”.

Один з них, метод який не передбачає навчання мережі звичним шляхом - через зворотню дифузію. Зосереджуючись лише на оптимізації багатоетапного процесу дифузії. Процес дифузії зазвичай розглядають як вирішення звичайних та стохастичних диференціальних рівнянь, тобто прямого і реверсивного процесу, тому зазвичай їх покращують шляхом пошуку кращого розвязувача, подібні покращені розв’язувачі були описані вище, це DDIM, DDPM та DPM, вони зменшують кількість кроків дифузії і оптимізують процес навчання. Але існують інший метод - метод без тренування, які базуються на ітераціях Пікара або нормалізуючих потоках.

в Ітерації Пікара пропонують, замість обчислень кожного кроку послідовно, робити початкові припущення щодо всього шляху дифузії, а потім

паралельно уточнювати ці припущення в декілька ітерацій. Це дозволяє відійти від повільної послідовності на користь швидкого паралельного за рахунок обробки з GPU.

В роботі з нормалізуючими потоками, використовується ефективне перетворення простого розподілу шуму у складний розподіл реальних даних. З метою забезпечити ефективне відображення відбувається стиснення і ущільнення шляху від шуму до даних. Цей метод дозволяє швидше навчити модель залежності між різними часовими кроками дифузії, що також сприяє паралелізації та зменшенню загальнох затримки генерації.

3.4. Актуальне застосування звукових генеративних дифузійних моделей

Розквіт генеративного ШІ не обійшов і звукову царину мультимедійних продуктів. Генерація аудіо текст-в-звук, звук-в-звук чи покращення вже існуючих звукових доріжок це все лише мала частина того з чим сьогодні допомагають працювати дифузійні моделі в аудіосинтезі. Музична індустрія активно трансформується і підлаштовується під сучасний світ і не упускав можливості економити на живій робочій силі на користь більш дешевої машинної. Тому моделі які поступово перетворюють шум на повноцінний повний звуковий сигнал тісно сплітаються в роботу.

Перша найбільш очевидна сфера застосування дифузійних моделей є музична. Композитори, лейбли, аматори та інді гурти з усього світу створюють унікальні семпли для натхнення. Замість того, щоб годинами сидіти шукати потрібні звуки серед сотні музичних бібліотек.

Існують великі моделі, розроблені гігантами комп’ютерної індустрії. Наприклад, модель Google MusicLM використовує дифузійний підхід для створення високоякісної музики з текстових описів, пропонуючи швидкий і доступний рівень контролю над жанром, інструментами та настроєм композиції. Інший приклад – Riffusion, що дозволяє генерувати музику на основі візуальних спектрограм, поєднуючи світ зображень та аудіо. Ці інструменти не лише прискорюють творчий процес, але й роблять створення музики доступнішим для ширшого кола людей.

Комерційне застосування також процвітає. Рекламні агентства

використовують генеративні аудіомоделі для швидкого створення фонової музики для відеороликів, озвучка 3д маскотів чи інших анімованих персонажів, також згенерованих за допомогою генерації відео. Розробники ігор – для генерації динамічної фонової музики чи звукових ефектів інтерфейсів чи бойової системи. Це значно скорочує витрати часу та ресурсів на традиційні студійні записи. Люди які працюють з промпт запитами і надають свої послуги беруть значно менше ніж схожа послуга коштує на ринку праці.

Генерація людського голосу це ще одна галузь, де дифузійні моделі демонструють використовуються на до покращення мультимедійного контенту для стрімерів, блогерів чи людей які створюють різний контент. Якість синтезованого мовлення стала настільки високою, що його важко відрізнити від справжнього людського голосу. Це має величезне значення для людей які бажають залишитися анонімними але мають бажання підтримувати кар'єру яка вимагає взаємодіяти з інтернет користувачами.

У сфері розваг дифузійні моделі використовуються для дубляжу фільмів та серіалів на різні мови, зберігаючи оригінальну інтонацію та емоції акторів. Також відеоблогери різних галузей все частіше самі налаштовують на своїх відео можливість дивись його різними мовами і все що для цього треба це один промпт і платна підписка на сервіс який ці послуги надає. Важко навіть уявити скільки б це вимагало ресурсу, як людського так і фінансового, щоб перекласти відео довжиною 30 хвилин двадцятьма мовами. Треба знайти акторів дубляжу, знайти перекладачів, домовитись з студією для запису і всі ці процедури мають відбутись окремо для кожної мови. Сьогодні це декілька рядків промпту.

Також однією з популярних ніш це зміна тональності голосу, що дозволяє людям озвучувати відео чи вести стріми та подкасти не своїм справжнім голосом, а застосовувати цікаві незвичні фільтри, або навіть змінювати стать. Це дозволяє не тільки уникнути непотрібної уваги, а й дозволяє урізноманітнити мультимедійний контент, щоб ним зацікавилась ширша аудиторія.

Персоналізовані голосові асистенти, або озвучка аудіокниг чи наукових робіт також отримали свою увагу в ніші дифузійних голосових моделей. Тепер це дозволяє непросто витрачати час і зосереджено читати різного роду матеріали

а й слухати, звільнивши людей від необхідності обмежувати себе в діях. Окрім цього цикавою виглядає можливість для студентів навіть обирати голос лектора, оскільки велика кількість професорів вже прийняли участь чи отримали запрошення на те щоб доєднатись і поділитись своїми голосами для сервісів які надають послуги озвучки тексту. Таким чином, якщо є лектор голос якого допомагає зосередитись на матеріалі найбільше, то подібні сервіси дають можливість не втрачати цю концентрацію навіть з тим контентом до якого цей професор не має жодного відношення. Якість мовлення яка генерується подібними дифузійними моделями вражає відсутністю будь яких машинних звуків чи підробленої інтонації. Моделі навчились вправно імітувати емоції, жартівливий тон, обурення чи страх, це все для моделі набір даних яким вони вже вправно володіють і використовують орієнтуючись на наповнення сторінок які озвучують.

Найкращою сферою використання є дифузійні моделі які допомагають людям з обмеженими можливостям. Вони дозволяють створювати персональних асистентів, або голоси для альтернативної та доповненої комунікації, що значно покращує якість життя користувачів.

Перше що спадає на думку при словах “генерація аудіо” це створення нових унікальних екземплярів аудіо, але ще однією ланкою цих моделей є застосування ШІ для обробки та покращення існуючих аудіозаписів. Дифузійні генераційні моделі показали неабияку ефективність в тому щоб покращувати вже існуючі аудіодоріжки. Допомагати користувачам з поганим мікрофоном, або рятувати в моменти, коли під час роботи професійного обладнання сталась накладка. Саме в такі критичні моменти на допомогу приходить штучний інтелект. Моделі сьогодні можуть виконувати завдання з видалення шуму, ревербації та покращення чіткості мовлення, витягуючи потрібні частоти.

Загалом, дифузійні моделі стали фундаментальною технологією, яка стала невід’ємною частиною різних аспектів роботи зі звуком у всьому світі. Вони не просто автоматизують процеси, а відкривають нові творчі горизонти та значно покращують якість кінцевого продукту.

Актуальні дослідження зосереджені на створенні мультимодальних

систем, де дифузійні моделі працюють у синхронно з іншими видами ШІ, наприклад, щоб згенерована аудіо доріжка ідеально лягала на відеоряд або переклад іноземних мов відбувався за долі секунди, щоб затримка була настільки мінімальна, щоб у людини перед якою співрозмовник не було навіть думки про те що він чує машинний переклад, а чистота перекладу і тональність голосу ідеально повторювала взірець голосу. Це наближає людство до ери повністю автоматизованого створення високоякісного медіаконтенту, де штучний інтелект виступає як потужний співтворець та інструмент оптимізації. Застосування дифузійних моделей в аудіосфері – це динамічний процес, що постійно розвивається, і ми тільки починаємо бачити його справжній вплив на глобальну індустрію.

4. СТВОРЕННЯ КОДА ДЛЯ ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ АБО АНІМАЦІЙ

У цьому розділі висвітлено процес створення та застосування генеративної дифузійної моделі використовуючи бібліотеки PyTorch та deepinv. В першій частині буде продемонстровано навчання моделі на датасеті MNIST. Друга частина буде повністю присвячена застосуванню навченої моделі і генерації нових зображень.

4.1. Тренування дифузійної моделі

Для реалізації цієї задачі було обрано Python, а також дві бібліотеки PyTorch і DeepInverse. Бібліотека Pytorch неодноразово згадувалась у цій магістерській роботі, оскільки є незамінним інструментом для роботи з навчання нейронних мереж і роботи з штучним інтелектом, як і при роботі з візуалізацією так і з генерацією аудіо. Друга бібліотека має назву DeepInverse, це є відкрита бібліотека на основі PyTorch, яка призначена для вирішення обернених задач обробки зображень за допомогою методів глибокого навчання. Обидві бібліотеки додаються через командну строку, хоча для PyTorch треба ознайомитись з інструментом (рис.4.1), який допоможе підібрати спеціальну команду. Всі параметри обираються відповідно до системи, система яку я використовую має відеокарту 3090Ti яка працює на базі CUD ядер 13.0. Бібліотека DeepInv додається простою командою через -pip.

Рис. 4.1. Інструмент підготовки команди для імпорту бібліотеки PyTorch

Після того як обидві бібліотеки було встановлено переходимо до налаштування набору даних. Буде використано MNIST (Modified National

Institute of Standards and Technology dataset) (рис. 4.2) - це набір даних невеликих, який складається з 60 000 тренувальних і 10 000 тестових зображень рукописних цифр від 0 до 9, всі зображення розміром 28x28 пікселів і у градаціях сірих відтінків.

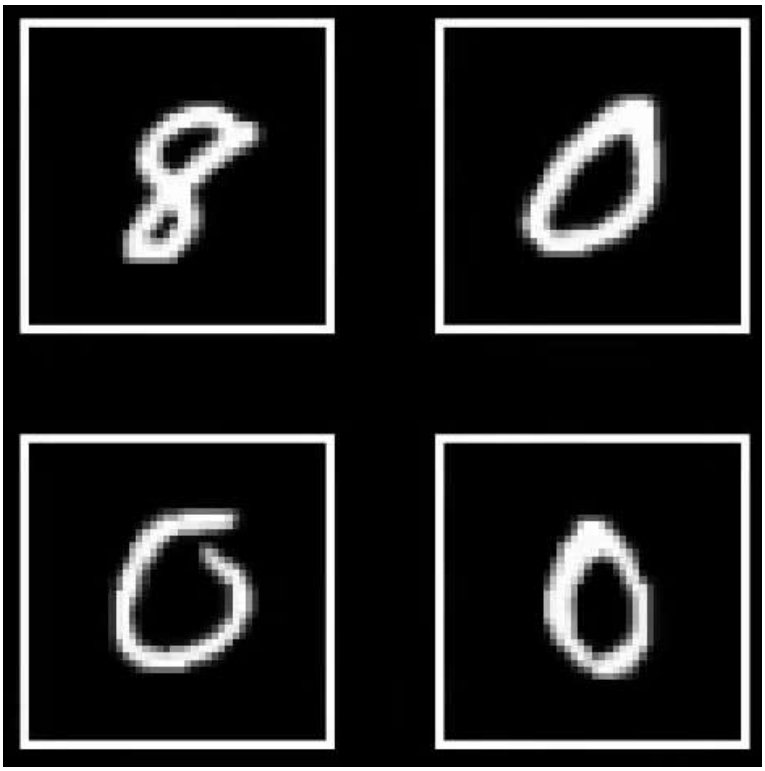


Рис. 4.2. приклад набору даних під назвою MNIST

Перші рядки це імпорт вище доданих до системи бібліотек. Переходимо до налаштування датасету. Визначаємо пристрій за допомогою рядка device = "cuda" і встановлюємо розміріи 28 пікселів batch_size та image_size. Далі встановлюємо послідовність трансформації, яка буде застосована до кожного зображення перед його подачею в модель (transform = transforms.Compose). І змінюємо усі зображення (transforms.Resize) до вказаного раніше до вказаного раніше (image_size). Далі відбувається конвертація зображення з формату PIL image у багатовимірний масив PyTorch, який масштабує значення пікселів із діапазону [0, 255] до [0.0, 1.0]. В свою чергу transforms.Normalize((0.0,), (1.0,)): Нормалізує тензор зображення, віднімаючи середнє значення (0.0) і ділячи на стандартне відхилення (1.0), що допомагає моделі краще навчатися.

З допомогою рядка train_loader = torch.utils.data.DataLoader(...): створюємо завантажувач даних (DataLoader), який спрощує ітерацію по набору даних, забезпечує пакетну обробку, перемішування та паралельне завантаження.

datasets.MNIST(root="./data", train=True, download=True, transform=transform): Завантажує набір даних MNIST (зображення рукописних цифр).

root="./data": Вказує локальну теку для зберігання даних.

train=True: Завантажує тренувальний набір даних.

download=True: Якщо даних немає локально, завантажує їх з інтернету.

transform=transform: Застосовує раніше визначений ланцюжок трансформацій до кожного зображення.

batch_size=batch_size: Встановлює розмір пакетів.

shuffle=True: для перемішування даних на початку кожної епохи (проходу по всьому набору даних), що важливо для кращого узагальнення моделі.

```
import torch
import deepinv
from torchvision import datasets, transforms

device = "cuda"
batch_size = 28
image_size = 28

transform = transforms.Compose(
    [
        transforms.Resize(image_size),
        transforms.ToTensor(),
        transforms.Normalize((0.0,), (1.0,)),
    ]
)
train_loader = torch.utils.data.DataLoader(
    datasets.MNIST(root="./data", train=True, download=True, transform=transform),
    batch_size=batch_size,
    shuffle=True,
)
```

Рис. 4.1. Установка датасета

Далі слід визначити все, що пов’язано з нейронною мережею та на налаштуванням навчання. Завдяки бібліотеці DeepInverse це досить просто зробити. Слід просто створити екземпляр класу Diffusion UNet та вказати кількість вхідних та вихідних каналів. Цю модель було обрано навчати з нуля, тож не слід завантажувати жодних попередньо навчених вагових коефіцієнтів. Для оптимізації використовується оптимізатор Adam зі стандартною швидкістю читання. А для функції втрат використовуватиметься проста середня квадратична похибка, про яку було сказано вище в розділі [2.2].

```
lr = 1e-4
epochs = 100

model = deepinv.models.DiffUNet(in_channels=1, out_channels=1, pretrained=None).to(
    device
)
optimizer = torch.optim.Adam(model.parameters(), lr=lr)
mse = deepinv.loss.MSE()
```

Рис. 4.2. Налаштування моделі

Наступним кроком буде визначити всі константи, які використовуватимуться для дифузії. Першою буде бета-графік, який описує рівень шуму, що додається на кожному етапі процесу дифузії (рис. 4.3).

```
beta_start = 1e-4
beta_end = 0.02
timesteps = 1000

betas = torch.linspace(beta_start, beta_end, timesteps, device=device)
```

Рис. 4.3. Бета-графік

Далі обчислюються alphas, кумулятивні добутки альфа-бару та квадратний корінь з альфа-бару та його доповнення (рис.4.4)

```
beta_start = 1e-4
beta_end = 0.02
timesteps = 1000

betas = torch.linspace(beta_start, beta_end, timesteps, device=device)
alphas = 1.0 - betas
alphas_cumprod = torch.cumprod(alphas, dim=0)
sqrt_alphas_cumprod = torch.sqrt(alphas_cumprod)
sqrt_one_minus_alphas_cumprod = torch.sqrt(1.0 - alphas_cumprod)
```

Рис. 4.4. Дифузійні константи

Тепер коли константи встановлено, можна переходити до циклу навчання. На кожному кроці вибирається випадковий часовий крок t між 0 і 1000, та формується вибірка шуму ϵ зі стандартного нормального розподілу. Потім зашумлення зображення обчислюються, використовуючи формулювання зі збереженням дисперсії, яке було отримано раніше. Це зображення з шумом та часовий крок потім передаються в мережу. Модель оцінює шум, присутній на зображенні і обчислює втрати між прогнозом та ϵ. Цей процес шумозаглушення зображення повторюється, а потім обчислення MSE, доки модель не буде навчена, що займає близько 100 епох. В кінці навчання рядок зі збереженням моделі, щоб мати змогу використовувати їх пізніше.

```python
for epoch in range(epochs):
    model.train()
    for data, _ in train_loader:
        imgs = data.to(device)
        noise = torch.randn_like(imgs)
        t = torch.randint(0, timesteps, (imgs.size(0),), device=device)

        noised_imgs = (
            sqrt_alphas_cumprod[t, None, None, None] * imgs
            + sqrt_one_minus_alphas_cumprod[t, None, None, None] * noise
        )

        optimizer.zero_grad()
        estimated_noise = model(noised_imgs, t, type_t="timestep")
        loss = mse(estimated_noise, noise)
        loss.backward()
        optimizer.step()

```

Рис. 4.5. Петля для тренування моделі

Під час навчання можна спостерігати прогрес моделі, для прикладу я візьму випадкові елементи на різних етапах (рис. 4.5). Зліва підписане x_0 це оригінальне зображення, потім зашумлені версії на різних часових кроках та оцінку з усуненням шуму, обчислену за допомогою нейронної мережі. На ранніх етапах навчання результати погані на високих рівнях шуму, але не викликають питань за низького рівня шуму, оскільки це вже не такі важкі задачі для моделі.

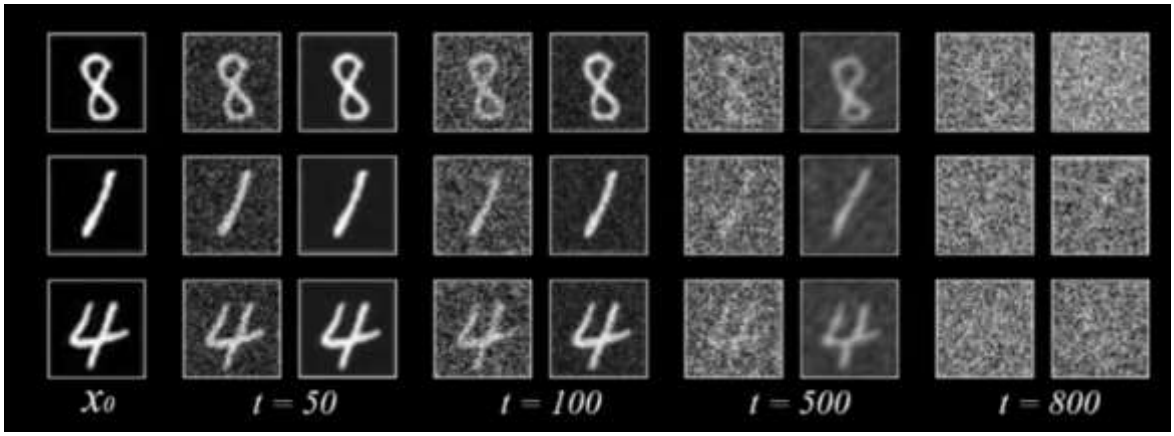


Рис. 4.5. Прогрес моделі в перші 10 циклів

Але якщо продовжувати навчання. то з кожним циклом результати стають краще з кожним циклом, можна спостерігати незначне покращення з кожним кроком епохи, незначні покращення моделі на 30 циклі. Ідеальні цифри ни низьких рівнях шуму і вже краще видно на високих рівнях шуму на шестидесяти епохах. І вже значний прогрес на фінальному етапі (рис. 4.6).

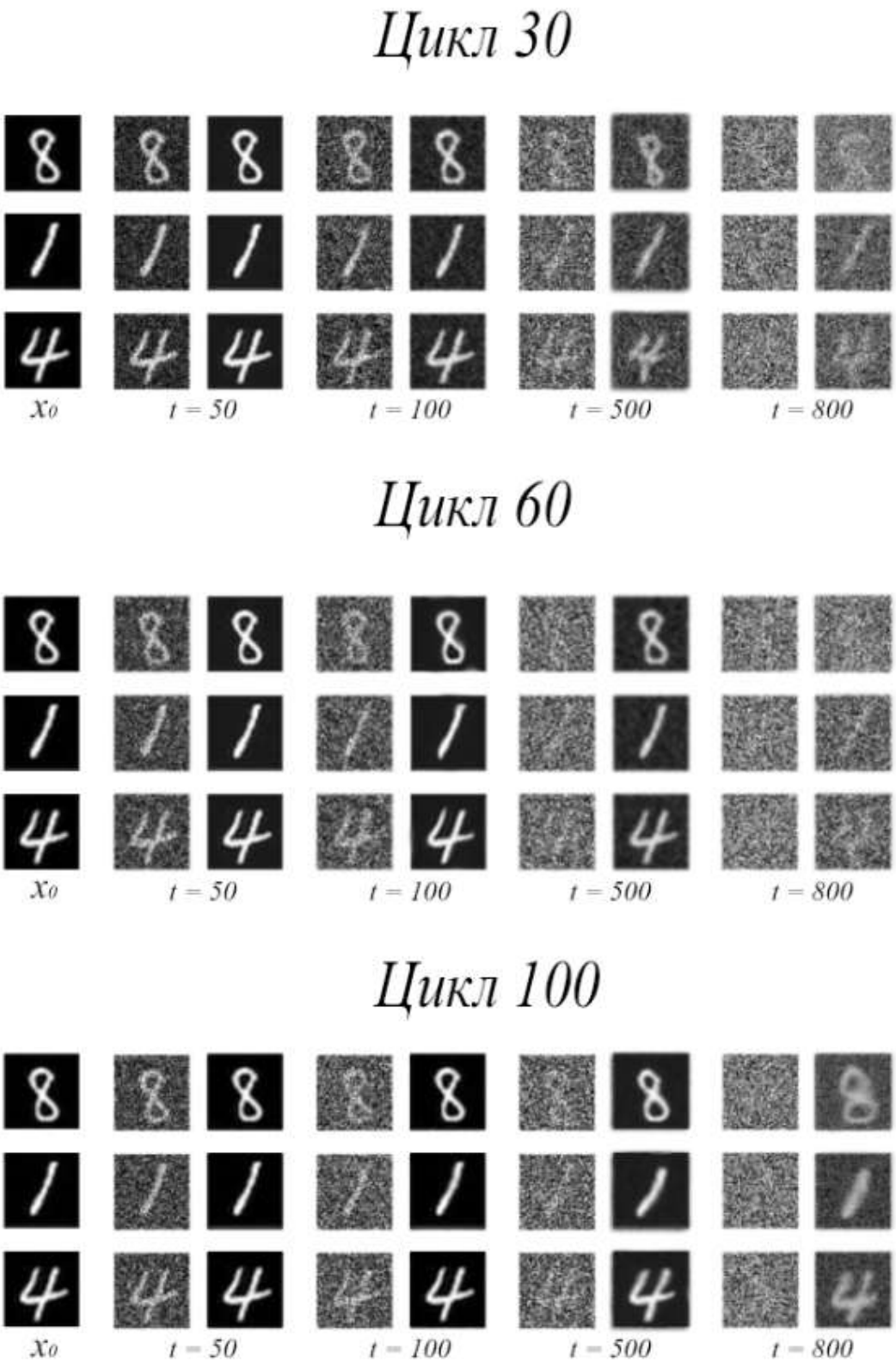


Рис. 4.6. Прогрес моделі на циклі 30, 60 і 100

4.2. Генерація зображень за допомогою моделі

Тепер коли етап тренування пройдено наступним етапом буде прописати код інференсу та згенерувати нові зразки із чистого гаусового шуму. Код

виведення буде реалізовано окремим файлом. Початок ідентичний до попереднього файлу, імпорт бібліотек та створення екземпляра нейронної мережі, використовуючи попередній файл під ідентичною назвою “Magisterska_diffusion_model_training.pth” (рис. 4.7)

```
import torch
import deepinv
from pathlib import Path

device = "cuda"
image_size = 28
checkpoint_path = "./checkpoints/Magisterska_diffusion_model_training.pth"
model = deepinv.models.DiffUNet(
    in_channels=1, out_channels=1, pretrained=Path(checkpoint_path)
).to(device)
```

Рис. 4.7 Імпорт бібліотек та попередньо навченої моделі

Далі відбувається перевизначення усіх констант, які були використані під час навчання (рис. 4.8). Ці параметри мають ідеально збігатись між навчанням та логічним висновком, інакше кількість шуму на заданому кроці часу під час логічного висновку не відповідатиме тій самій кількості шуму, що спостерігається під час навчання.

```
beta_start = 1e-4
beta_end = 0.02
timesteps = 1000
betas = torch.linspace(beta_start, beta_end, timesteps, device=device)
alphas = 1.0 - betas
alphas_cumprod = torch.cumprod(alphas, dim=0)
sqrt_alphas_cumprod = torch.sqrt(alphas_cumprod)
sqrt_one_minus_alphas_cumprod = torch.sqrt(1.0 - alphas_cumprod)
```

Рис. 4.8. Перевизначення усіх констант

Тепер слід перейти до реалізація циклу выборки. Перший етап це вибірка чистого гаусового шуму, який відповідає початковій змінній x_T. Потім цей шум разом із кроком часу t подається в нейронну мережу для прогнозування ϵ – оцінки шуму всередині цього зображення. Ця оцінка шуму тепер використовуватиметься для отримання x_{t-1}, що є вибіркою з приблизної апостеріорної функції. Для цього оцінка шуму підставляється у формулу апостеріорного середнього і додається трохи шуму для фактичної вибірки з приблизного апостеріорного значення. $\mu_0 x_t, t) = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_{t-1}}}\epsilon_0(x_t, t)\right)$.

Потім цей процес повторюється, оновлення зображення передається далі в мережу, апостеріорне середнє обчислюється та вибірково аналізуємо наступне зображення, цей процес відбувається тисячу разів.

```
model.eval()
n_samples = 28
with torch.no_grad():
    x = torch.randn(n_samples, 1, image_size, image_size).to(device)
    for t in reversed(range(timesteps)):
        t_tensor = torch.ones(n_samples, device=device).long() * t
        predicted_noise = model(x, t_tensor, type_t="timestep")
        alpha = alphas[t]
        alpha_cumprod = alphas_cumprod[t]
        beta = betas[t]
        if t > 0:
            noise = torch.randn_like(x)
        else:
            noise = 0
        x = (1 / torch.sqrt(alpha)) * (
            x - (beta / torch.sqrt(1 - alpha_cumprod)) * predicted_noise
        ) + torch.sqrt(beta) * noise
```

Рис. 4.9. Семплювання

Перші сто разів дифузії сильно нічого не відбувається, але результат поступово покращується з кожнєю сотнею циклів (рис. 4.10).

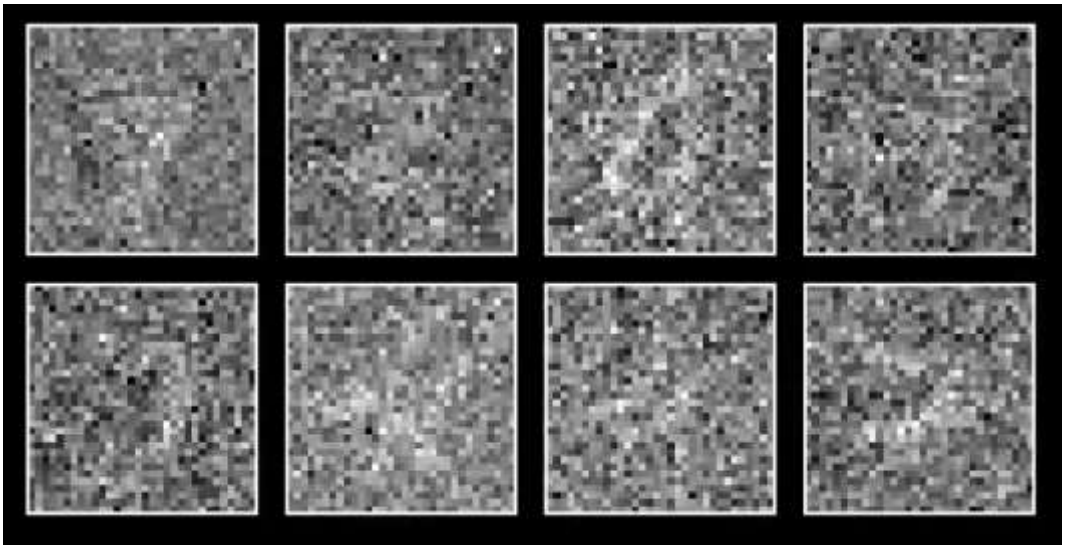


Рис. 4.10. Двохсотий цикл дифузії

На половині шляху вже стає видно очерк цифр (рис. 4.11)

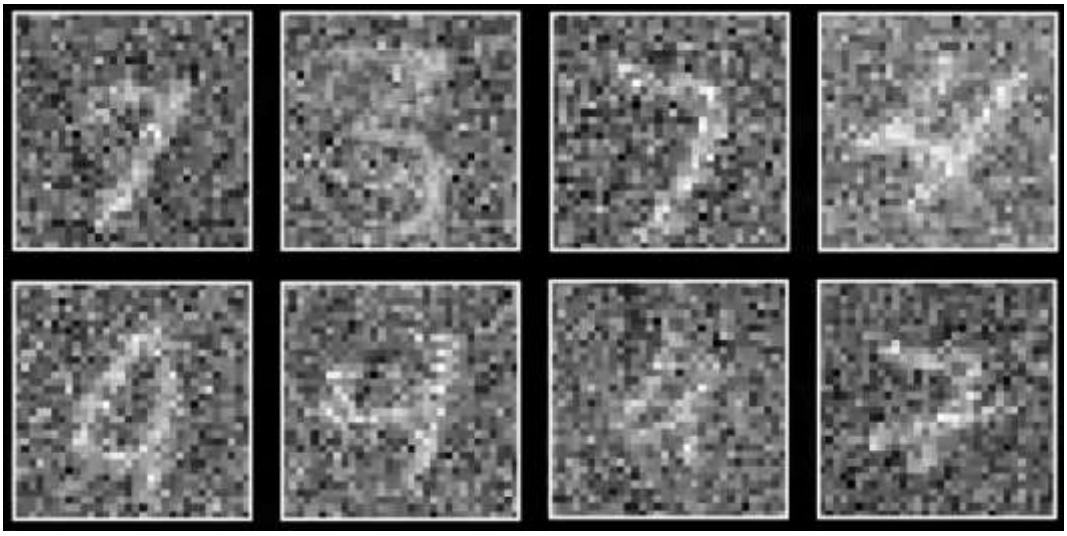


Рис. 4.11. Пятисотий цикл дифузії

Коли підійшла черга останній кроків дифузії деталі шум повністю покидає чистий рівний контур фігур і деталі вимальовуються в зрозумілу картину. Як фінал модель видає дуже чисті зразки.

Рис. 4.12. Фінальний цикл дифузії

Як результат повністю з нуля було створено високоякісні рукописні цифри, використовуючи лише шумозаглушувач та кілька рівнянь дифузії. Завдяки бібліотеці DeepInverse можна покращити цей результат ще далі і навчити модель на інших моделях навчання і навчити модель генерувати інший набір даних, це можуть бути тварини, портрети облич або будь який предмет, в залежності від того яка модель тренування моделі буде обрана. А для того щоб поміняти модель кодування треба лише поміняти два рядки коду. Якщо для цього коду було використано MNIST датасет, то для портретів використовують FFHQ. Fashion-MNIST використовують для одягу, цей датасет складається з 70 000 чорно-білих

зображень різного одягу. Датасет CelebA містить понад 200 000 зображень знаменитостей.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи на тему “Diffusion-моделі у генерації мультимедійного контенту” були досягнуті значні результати, що підтверджують ефективність дифузійних моделей і роботи зі штучним інтелектом.

Був проаналізований розвиток штучного інтелекту штучного інтелекту і детально розглянута зміна ролі, яку відігравав штучний інтелект протягом свого розвитку. Були розглянуті основні види генеративних моделей їх переваги і недоліки, їх характеристики і можливості при їх використанні. Визначено ризики та етичні недоліки надмірного і необдуманного використання генеративних моделей як інструмент створення мультимедійного продукту, вплив на громадську думку або маніпулювання інформацією. Розглянуто випадки судових позовів і мітингів через використання штучного інтелекту. Визначено, що штучний інтелект сьогодні є невід’ємною частиною суспільства, без нього важко уявити сферу обслуговування, сферу розваг і створення мультимедійного контенту, медичну сферу і комп’ютерні технології.

Основним принципом навчання дифузійних моделей є принцип прямої і зворотної дифузії. Цей метод дозволяє навчатись на будь якому контенті, за рахунок можливостей накладання Гаусового шуму і його очищення. Було розглянуто різні принципи навчання і математичну частину дифузійної архітектури. Було детально розглянуто принцип дифузії і тренування моделі для зображень. Потім доповненно принципами навчання для відео, як продовження зображення. І як останній елемент представлено елементи які принципово відрізняють тренування і навчання для аудіо. Виявлено, що основи тренування і генерації у всіх трьох різних видів контенту схожий, але в певний момент додаються певні відмінності для кожного виду мультимедійного продукту.

Було визначено, що Diffusion-моделі стали проривом у генерації мультимедійного контенту випередивши усі інші альтернативи при цьому охоплюючи широкий спектр сфер використання. Дифузійні моделі можуть використовуватись для генерації зображень, відео, 3Д моделей, аудіо і інших сфер.

Проаналізовано можливості різних моделей та сервісів які використовують Diffusion-моделі. Визначено переваги і недоліки кожної і перспективи використання. Перспективною моделлю для генерації зображень наразі є DALL-E 3 та MidJourney, саме ці дві моделі наразі пропонують найбільший інструментарій і функціонал, а також якість створених зображень. Визначено, що для генерації відео найкращими залишаються Sora AI та Veo 3 від Google, ці два сервіси пропонують свої унікальні можливості для використання кожному в залежності від потреб. Конкуруючи одна з одною ці моделі розвиваються, щоб захопити увагу користувача. Виявлено, що великі студії по створенню мультимедійного контента використовують штучний інтелект для полегшення виробництва і зменшення бюджетів, на що негативно реагує глядач. Також розглянуто можливості в генерації аудіо

Мова програмування Python, а робота відбувалась з бібліотеками PyTorch та DeepInverse. Було детально описано переваги цих бібліотек і можливості які вони пропонують. Python залишається ідеальним вибором для створення власної моделі і її навчання на датасетах будь якої ємності. Обрані бібліотеки є лише одними з безлічі, було розглянуто додаткові бібліотеки які можна було б використати, як і безліч інших датасетів для тренування моделей.

Розроблено та впроваджено код для тренування дифузійної моделі на датасеті MNIST, який складається з 60 000 зображень для навчання і 10 000 зображень для тестування. Також підготовлено код для генерації зображень за допомогою навченої моделі. Навчена модель може генерувати унікальні зображення цифр. Також продемонстровано, що заміна лише декількох рядків дозволить змінити результат генерації з цифр на інші зображення. В залежності від обраного датасета це можуть бути : одяг, обличчя людей, зображення знаменитостей, тощо.

Отже результати досліджень зробили значний внесок у розвиток штучного інтелекту для генерації за допомогою Diffusion-моделей та продемонстрували перспективи використання дифузійних генеративних моделей у соціальних мережах, мультимедійних ресурсах і при професійній діяльності. Використання генеративних ШІ дозволяє пришвидшити роботу, зменшити витрати при роботі

з мультимедіа, підвищити якість мультимедійного продукту. Нові методи створення контенту за допомогою генеративних ШІ дозволяють отримати кращі і швидші унікальні результати з кожним днем, що дозволяє ширшому колу людей почати створювати контент і урівнює умови.

Таким чином, результати дослідження підтвердили ефективність та перспективність використання Diffusion-моделей у генерації мультимедійного контенту, що робить значний внесок у розвиток цієї галузі та відкриває нові можливості для подальших досліджень.

ПЕРЕЛІК ПОСИЛАНЬ

1. Denoising Diffusion Probabilistic Models [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2006.11239

2. Deep Unsupervised Learning using Nonequilibrium Thermodynamics [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1503.03585

3. A Logical Calculus of the Ideas Immanent in Nervous Activity [Електронний ресурс]. – Режим доступу: https://www.cs.cmu.edu/~./epxing/Class/10701-08s/recitation/mcculloch.pitts.pdf

4. Learning representations by back-propagating errors [Електронний ресурс]. – Режим доступу: https://www.nature.com/articles/323533a0

5. ImageNet Classification with Deep Convolutional Neural Networks [Електронний ресурс]. – Режим доступу: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

6. WaveNet: A Generative Model for Raw Audio [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1609.03499

7. Jukebox: A Generative Model for Music [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2005.00341

8. Sora: Video generation models as world simulators [Електронний ресурс]. – Режим доступу: https://openai.com/sora

9. PyTorch Documentation [Електронний ресурс]. – Режим доступу: https://pytorch.org/docs/stable/index.html

10. DeepInverse Library Documentation [Електронний ресурс]. – Режим доступу: https://deepinv.github.io/deepinv/

11. MNIST Dataset (Modified National Institute of Standards and Technology dataset) [Електронний ресурс]. – Режим доступу: http://yann.lecun.com/exdb/mnist/

12. Stability AI. Stable Diffusion Public Release [Електронний ресурс]. – Режим доступу: https://stability.ai/blog/stable-diffusion-public-release

13. OpenAI. DALL·E 3 [Електронний ресурс]. – Режим доступу: https://openai.com/index/dall-e-3/

14. Midjourney Documentation [Електронний ресурс]. – Режим доступу:

https://docs.midjourney.com/

15. Google Research. Veo: Our most capable generative video model [Електронний ресурс]. – Режим доступу: https://deepmind.google/technologies/veo/

16. J. Ho, A. Jain and P. Abbeel, "Denoising Diffusion Probabilistic Models," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 6840-6851, 2020.

17. J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan and S. Ganguli, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics," in Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 2015, pp. 2256-2265.

18. R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 10684-10695.

19. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh et al., "Learning Transferable Visual Models from Natural Language Supervision," in Proceedings of the 38th International Conference on Machine Learning (ICML), vol. 139, pp. 8748-8763, 2021.

20. A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Communications of the ACM, vol. 60, no. 6, pp. 84-90, May 2017, doi: 10.1145/3065386.

21. A. van den Oord et al., "WaveNet: A Generative Model for Raw Audio," in arXiv preprint arXiv:1609.03499, 2016.

22. P. Dhariwal et al., "Jukebox: A Generative Model for Music," in arXiv preprint arXiv:2005.00341, 2020.

23. D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," in Nature, vol. 323, pp. 533-536, Oct. 1986, doi: 10.1038/323533a0.

24. A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 2017, pp. 5998-6008.

25. A. Nichol, P. Dhariwal, A. Ramesh et al., "GLIDE: Towards Photorealistic

Image Generation and Editing with Text-Guided Diffusion Models," in Proceedings of the 39th International Conference on Machine Learning (ICML), Baltimore, MD, USA, 2022, pp. 16784-16804

Додаток А. Згенероване моделлю DALLE-3 зображення

Додаток Б. Згенероване моделлю DALLE-2 зображення

Додаток В. Демонстраційний матеріал

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних систем та технологій

«DIFFUSION-МОДЕЛІ У ГЕНЕРАЦІЇ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ»

Виконав студент групи САДМ-61 Гліб КОШКАРЬОВ
Науковий керівник : Олександр МАСТАКОВ

Мета роботи та актуальність

Мета роботи - розробка моделі для дифузійної генерації контенту.

Об'єкт дослідження - Системи штучного інтелекту для створення мультимедійного контенту за допомогою генеративних моделей

Предмет дослідження : Архітектура та алгоритми дифузійних моделей (наприклад, DDPM, Stable Diffusion) для формування візуального контенту.

Актуальність - обумовлення широким використанням генеративних моделей і їх роль у житті людства. Стрімкий розвиток, який вимагає щоденно приділяти увагу цій ніші.

Актуальність дослідження зумовлена широким використанням генеративних ШІ в сучасній культурі

Рис. 1. Приклад використання ШІ генерації в культурі

Завдання проекту

1. Дослідити розвиток штучного інтелекту та виклики з якими зіштовхнулась ця технологія.
2. Визначити основні архітектури дифузійних моделей, проаналізувати їх та виділити ряд відмінностей, переваг та недоліків.
3. Створити мультимедійний контент за допомогою дифузійних генеративних ШІ.
4. Реалізувати власну дифузійну модель для генерації зображень. Провести її тренування та проаналізувати можливості і перспективи навчання.
5. Підтвердити ефективність дифузійних моделей та можливості які надає дифузійна генерація мультимедійного контенту.

Дослідження принципу тренування дифузійних моделей

Шум як спосіб тренування та генерації

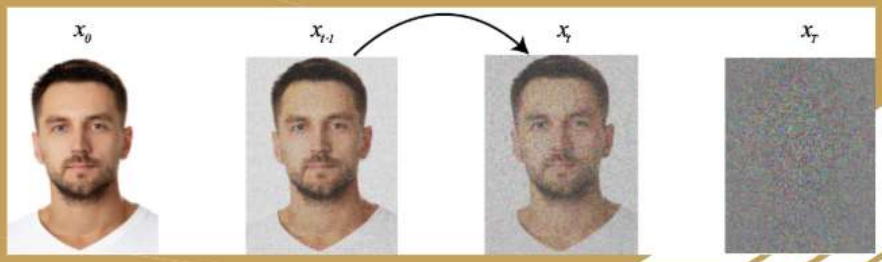


Рис. 2. Процес тренування моделі

Процес видалення шуму для генерації графіки

Рис. 3. Процес зворотної дифузії

Процес генерації текст-в-зображення

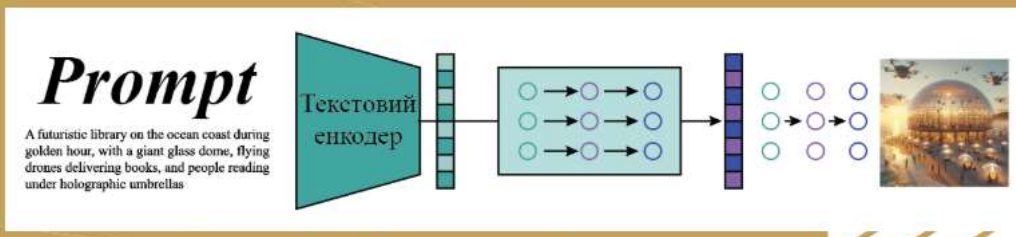


Рис. 4. Процес генерації через промпт

Дослідження аудіо генерації

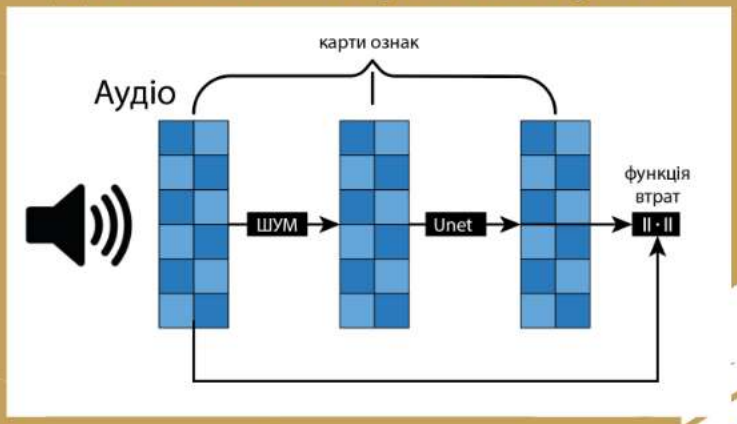
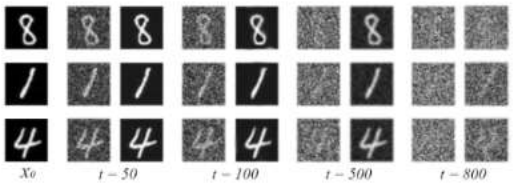


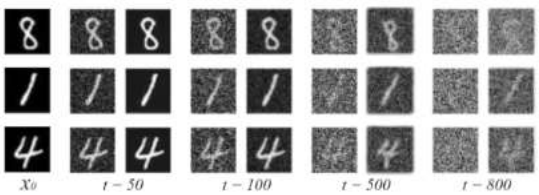
Рис. 5. Процес аудіогенерації

Процес тренування власної моделі

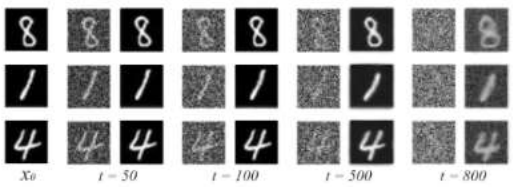
Цикл 60



Цикл 30



Цикл 100



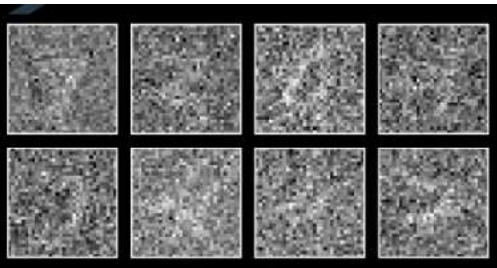


Рис. 6 Перші 500 циклів дифузії і останні 500 циклів

Висновки

1. Було досліджено розвиток штучного інтелекту та його логічний шлях до сьогоднішніх можливостей.
2. Було досліджено та розглянуто основні види генеративних моделей, їх методи тренування
3. Було сгенеровано мультимедійний контент за допомогою дифузійних генеративних ШІ.
4. Було реалізовано власну дифузійну модель для генерації зображень. Проведено її тренування та проаналізовно її можливості і перспективи навчання.

Апробація :

1. Кошкарьов Г.О. ЗАСТОСУВАННЯ ДИФУЗІЙНИХ МОДЕЛЕЙ У ГЕНЕРАЦІЇ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ III ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ. «ТЕХНОЛОГІЧНІ ГОРИЗОНТИ: ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ТЕХНОЛОГІЧНОГО ПРОГРЕСУ УКРАЇНИ І СВІТУ» Тези доповідей 18 листопада 2025 року ст.99 https://duikt.edu.ua/uploads/p_2779_90287546.pdf
2. Кашкарьов Г.О. ОСНОВНІ ДИФУЗІЙНІ АРХІТЕКТУРИ У ГЕНЕРАЦІЇ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ. І МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ. «ПРИКЛАДНІ СИСТЕМИ УПРАВЛІННЯ ТА РОБОТОТЕХНІКА». Доповідь 12-13 Листопада 2025 року. ст 84.