Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра інформатики

Кваліфікаційна робота

освітній ступінь — бакалавр

на тему: «**ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ СИНТЕЗУ ЗВУКУ ТА НАЛАШТУВАННЯ ПАРАМЕТРІВ ЗВУКОВОЇ ДОРІЖКИ НА ОСНОВІ ТЕКСТОВИХ ОПИСІВ**»

Виконав: студент 4-го року навчання
Спеціальності
122 Комп’ютерні науки
Письменний Антон Костянтинович
Керівник Медвідь С.О.
старший викладач
Рецензент _____
(прізвище та ініціали)

Кваліфікаційна робота захищена
з оцінкою _____
Секретар ЕК _____

«___»_____2025 р.

Київ 2025

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики

доцент, к.н

_____ Гороховський С.С

«___»_____2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на кваліфікаційну роботу

студенту 4 курсу факультету інформатики

Письменному Антону Костянтиновичу

ТЕМА: Використання машинного навчання для синтезу звуку та налаштування параметрів звукової доріжки на основі текстових описів

Зміст ТЧ до курсової роботи:

Зміст
Анотація
Вступ
1 Аналіз предметної області
2 Використання моделей text-to-sound у цифровій звуковій робочій станції
3 Інтеграція великої мовної моделі в віртуальний інструмент-синтезатор
4 Перспективи подальших досліджень

Дата видачі «___»_____ 2024 р. Керівник _____

Завдання отримав _____

Тема: <u>Використання машинного навчання для синтезу звуку та налаштування параметрів звукової доріжки на основі текстових описів</u>

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання завдання	Примітка
1	Ознайомлення з темою й завданням	04.11.2024	
2	Пошук та структурування загальної теоретичної інформації на тему	05.12.2024	
3	Формулювання оптимальних методів вирішення задачі й побудова відповідних програмних застосунків	10.02.2025	
4	Систематизація напрацювань, завершення написання тестової частини роботи	01.05.2025	

Письменний А.К. _____

Медвідь С.О. _____

«___» _____ 2025 р

Зміст

Анотація

Метою даної кваліфікаційної роботи було дослідження можливостей застосування алгоритмів машинного навчання для синтезу звуку та коригування параметрів звукової доріжки на основі текстового опису користувача з метою використання у сучасних музичних творах. Завданнями роботи були аналіз наявних засобів синтезу звуку з використанням машинного навчання та ступінь їхньої інтеграції в цифрові звукові робочі станції, розробка способів покращення теперішнього стану галузі, дослідження перспективності цих покращень шляхом створення демонстраційних програмних проєктів.

У роботі розглянуто існуючі моделі text-to-sound (зокрема – моделі латентної дифузії), перспективи використання великих мовних моделей (LLM) для керування синтезаторами, досліджено можливості створення віртуальних інструментів для цифрових звукових робочих станцій на базі зазначених підходів, проблеми та обмеження, з якими доводиться стикатися в ході розробки й використання таких засобів кінцевими користувачами.

Ключові слова: штучний інтелект, велика мовна модель, модель латентної дифузії, синтез звуку, цифрова звукова робоча станція

Перелік прийнятих скорочень

ЦЗРС	– цифрова звукова робоча станція
LLM	– велика мовна модель (large language model)
MM-LLM	– мультимодальна велика мовна модель (multimodal large language model)
ШІ	– штучний інтелект
VST	– формат програмних доповнень Virtual Studio Technology

Вступ

Дослідження в галузі штучного інтелекту останнім часом мають стійку тенденцію до кількісного зростання [1]. Постійне вдосконалення існуючих та поява нових технологій призводять до глибшої інтеграції засобів ШІ в наше повсякденне життя. Зокрема, така інтеграція відбувається у творчих напрямках, таких як живопис, фотографія, музика.

Крім спроб замінити митця цілком [2], розвивається також напрям створення засобів, що полегшували б його діяльність та відкривали б перед ним нові горизонти. Цей підхід ми спостерігаємо в продуктах, зокрема, компанії Adobe [3]. Завдяки контролю з боку користувача, безпосередньому залученні його креативності, така сумісна діяльність людини й машини є продуктивною й приносить якісні результати.

Стан моделей штучного інтелекту останніх років, спрямованих на генерацію зразків мультимедіа (зображень, аудіофрагментів тощо) відкриває широкий спектр можливостей. Як наслідок розвитку генеративних змагальних моделей [4] та появи моделей прихованої дифузії [5], можливо отримувати аудіозразки достатньої якості для використання в музичних творах.

Безперечно, використання штучного інтелекту під час написання музики можливе вже зараз, проте характер розвитку наявних технологій, що дозволяють це робити, суттєво відрізняється від інтеграції ШІ в процес обробки чи створення зображень, написання коду тощо. В першу чергу відмінності помітні в простоті та універсальності цього використання. У згаданих сферах ми бачимо такі продукти, як графічний редактор Adobe Photoshop чи редактор вихідного коду Cursor, що активно залучають штучний інтелект до синтезу контенту користувачем (мова йде про модифікацію обраних частин зображення за текстовим запитом [3], редагування коду за наданими вказівками і навіть запуск необхідних команд в

терміналі [6]). Лише обмежені прояви такого залучення наявні в провідних засобах для написання музики – цифрових звукових робочих станціях. Так, безперечно, існують такі програмні доповнення, як пропоновані Kits.ai, для генерації голосу чи написання мелодії на базі голосового вводу [7], проте в таких продуктах не реалізується весь потенціал синтезу аудіодоріжок та досягнення потрібного звучання синтезаторів з використанням засобів штучного інтелекту.

Тому метою даної роботи було визначено дослідження способів інтеграції моделей машинного навчання з сучасними інструментами для написання музики. Така інтеграція має полегшити синтез та вдосконалення звуку недосвідченим користувачам. Для демонстрації перспектив кожного напряму необхідно створити відповідні програмні продукти.

Текстовий опис, за рахунок придатності мови до побудови абстракцій, асоціацій та інших складних семантичних конструкцій, може допомогти спростити процес синтезу звуку. Це спонукає дослідити способи інтеграції моделей типу text-to-sound в програмні модулі-інструменти. Проте, враховуючи продемонстровану далі проблему обмеженої якості звукового сигналу, було б доцільним проаналізувати також перспективність іншого підходу, що дозволив би утилізувати існуючі засоби синтезу якісного цифрового сигналу (синтезатори) - використання універсальних великих мовних моделей для управління ними.

Робота складається з чотирьох розділів:

Перший розділ присвячений аналізу предметної області.

Другий розділ зосереджується на напрямку синтезу звуку з використанням моделей латентної дифузії (Latent diffusion models) та демонструє його перспективність на прикладі модуля-інструменту формату Virtual Studio Technology на базі моделі AudioLDM 2.

Третій розділ розглядає перспективи використання великих мовних моделей (LLM) для керування синтезаторами. В ньому продемонстровано створення програмного доповнення формату Virtual Studio Technology з використанням моделі Gemini як приклад застосування такого підходу.

Четвертий розділ описує потенційні шляхи продовження досліджень і покращення отриманих результатів.

1 Аналіз предметної області

1.1 Огляд наявних прикладів використання ШІ в синтезі звуку й музиці

Досліджуючи сучасні засоби штучного інтелекту, що пов’язані із синтезом звуку й мають застосування в галузі музики, а також безпосередньо створених для такого використання засобів, можна помітити три проблеми.

Проблема 1. Обмежена якість вихідного звукового сигналу рішень, що діють за принципом перетворення тексту в аудіозразки (text-to-sound). Помітні покращення в порівнянні з попередниками [8, с. 7] (як-от DiffSound) демонструє AudioLDM та AudioLDM 2, проте отримані в результаті звуки все одно містять характерні викривлення. Проведені порівняння амплітудно-частотних характеристик вихідних файлів, отриманих за допомогою AudioLDM 2, зі знайденими в онлайн-бібліотеках аудіозразками реальних, отриманих без використання ШІ звуків продемонстрували суттєві відмінності. Зокрема, отримані з AudioLDM 2 зразки не містять частот, вище за 7 кГц. Це демонструють рисунки 1.1 і 1.2, на яких представлені відмінності амплітудно-частотних характеристик аудіофайлу, отриманого за допомогою AudioLDM 2 за запитом “Birds singing calmly on cold winter morning” і файлу “city garden” з ресурсу Freesound відповідно.

Загальновідомим є факт, що людське вухо сприймає частоти від 20 до 20000 кГц [9]. Відповідно, отримані аудіозразки використовуватимуть менше половини доступного людському вуху діапазону. Для демонстрації важливості високих частот для сприйняття музики, було створено амплітудно-частотну характеристику звуку удару в хай-хет тарілки. Вона демонструє, що найбільша сила сигналу спостерігається якраз на частотах поза діапазоном генерації AudioLDM 2. Стає очевидною проблема якісних обмежень отриманих аудіозразків.

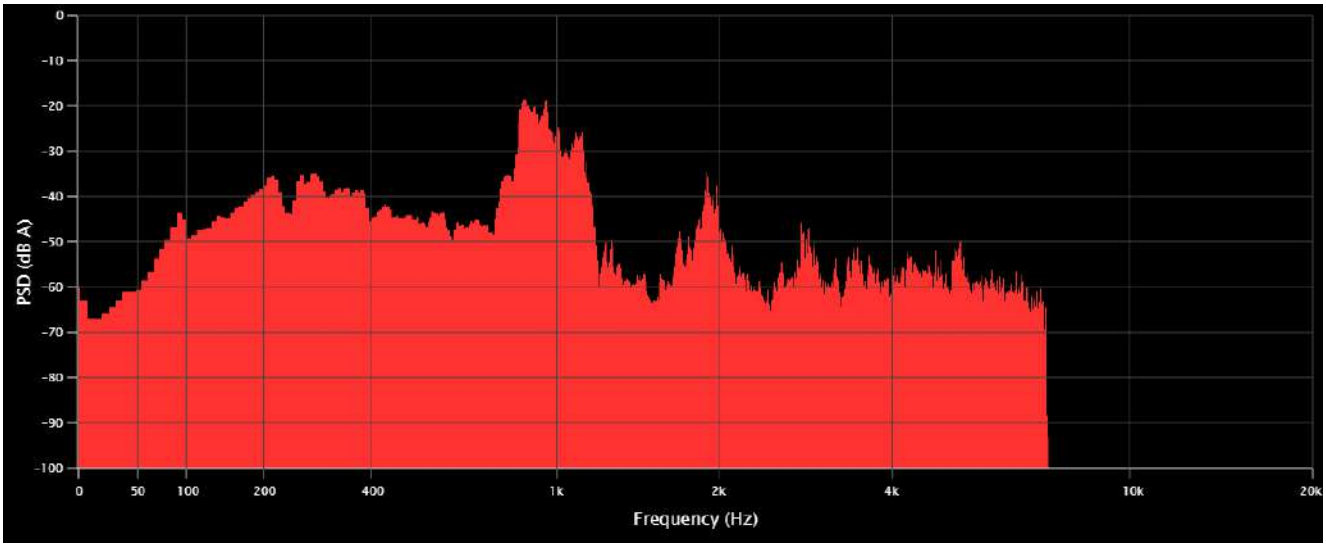

Рисунок 1.1 – АЧХ зразка, отриманого за допомогою AudioLDM 2

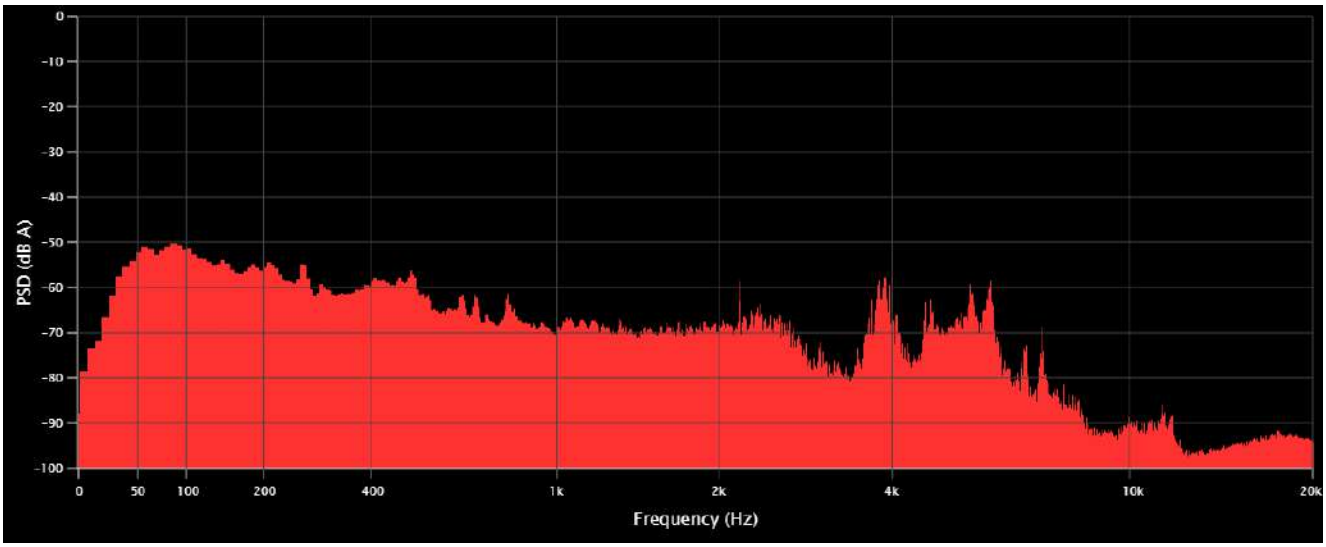

Рисунок 1.2 – АЧХ зразка, знайденого на платформі Freesound

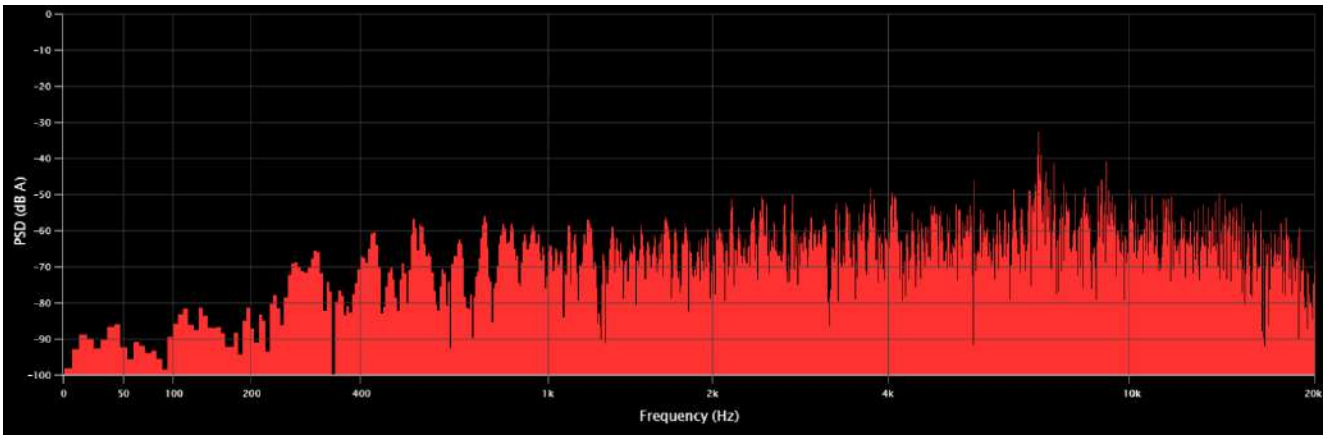

Рисунок 1.3 – АЧХ звуку удару в хай-хет тарілки

Проблема 2. Відсутність інтеграції безпосередньо в цифрову звукову робочу станцію (ЦЗРС) засобів, що могли б використовуватися початківцями. Так, наприклад, сервіс ElevenLabs дозволяє генерувати аудіозразки на базі текстового вводу користувача, проте використання результатів передбачає велику кількість накладних витрат (локальне збереження файлів, відкриття потрібним програмним доповненням), необхідність перемикатися між вікном браузера й цифровою звуковою робочою станцію, а також слабкий контроль за згенерованим результатом (відсутність регулювання довжини отриманої звукової доріжки та параметрів самої моделі).

Проблема 3. У разі наявності інтеграції в ЦЗРС – використання, що в першу чергу орієнтоване на досвідчених музикантів, а не на початківців. Наочно демонструє це Synplant2: синтезатор, що має функцію налаштування за існуючим аудіозразком, але вимагає розуміння принципів синтезу для покращення

отриманого звучання [10]. Складність інтерфейсу користувача демонструє знімок екрану нижче [11].

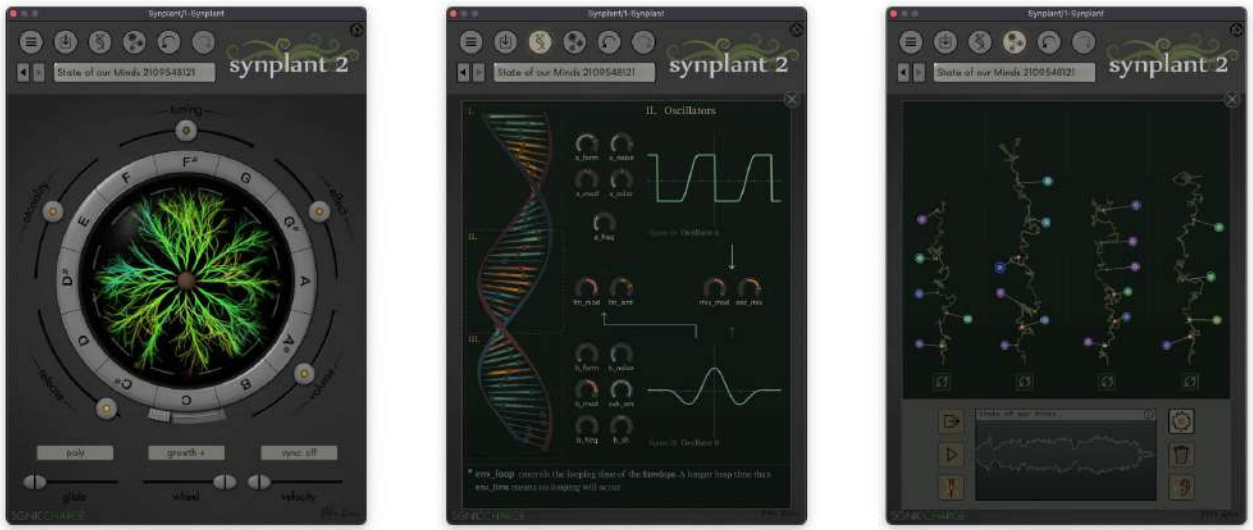

Рисунок 1.4 – Інтерфейс користувача програмного доповнення Synplant 2

DrumGAN також показує, що наявні рішення передбачають традиційні способи управління процесом синтезу з використанням таких засобів, як повзунки, прив’язані до конкретних параметрів [12].

Рисунок 1.5 – VST-модуль DrumGAN: інтерфейс користувача

Проведений аналіз демонструє, що наразі відсутні готові рішення, що дозволяли б синтезувати звук за текстовим запитом користувача та при цьому були б інтегровані в програмні продукти, з якими працює потенційна цільова аудиторія. Використання текстового опису для генерації потрібного аудіозразка, за рахунок продуктивності (здатності до створення нескінченної кількості змістовних повідомлень) та потенціалу до абстрактного опису людської мови [13], дозволило б істотно розширити можливості використання штучного інтелекту сучасними музикантами.

1.2 Практична цінність і новизна

Отже, під час аналізу предметної області прогалину, що полягає у відсутності досліджень потенційної інтеграції з цифровими звуковими робочими станціями сучасних text-to-sound моделей, а також інших засобів, що могли б допомогти музиканту з генерацією аудіозразків, використовуючи текстовий опис.

Відповідно бачимо можливість окреслити мету даної роботи, що буде заповнювати цю прогалину. Визначимо її як проведення дослідження способів такої інтеграції й створення дослідних зразків програмного забезпечення, що реалізували б ці способи й демонстрували їхню перспективність.

Логічним кроком була б спроба використання вузькоспеціалізованих моделей, безпосередньо орієнтованих на генерацію звуку за текстовими запитами. Проте, безперечно, це єдиний можливий варіант. Адже існують універсальні генеративні моделі, як-от великі мовні моделі, що у відповідь на текстовий запит, як демонструє практика [14], можуть створювати релевантні складні структури даних за шаблоном, описаним у цих запитах (наприклад, JSON-документи), без необхідності додаткових тренувань моделі. Такі документи, наприклад, могли б описувати поточний стан параметрів віртуального синтезатора, інтегрованого з моделлю в вигляді програмного доповнення.

Отже, зусилля було сконцентровано на двох згаданих напрямках роботи. Для демонстрації їхніх перспектив створено VST-модуль, в основі якого була обрана text-to-sound модель, а також програмне доповнення, що будується на базі універсальної великої мовної моделі. Останнє використовує переваги такої моделі, пов’язані з великим розміром контекстного вікна та якісним імітуванням розумових процесів. Вони дозволяють семантично поєднати побажання користувача з наявними в багатьох інструкціях до синтезаторів даних про те, як їх можна реалізувати на практиці.

Варто наголосити, що можливість створення ефективних і зручних програмних засобів у сучасному технологічному контексті для демонстрації поточного стану галузі була б найвищою оцінкою перспективності напряму. Проте очікується, що отримані в результаті досліджень рішення можуть виявитися не

достатньо практичними для реального використання через брак обчислювальних потужностей або певних технологій.

2 Використання моделей text-to-sound у цифровій звуковій робочій станції

2.1 Аргументація актуальності

Незважаючи на згадану проблему якості звуку сучасних text-to-sound моделей, не можна говорити про непридатність до використання аудіозразків, згенерованих такими моделями. Певні жанри музики та певні ситуації вимагають аудіо, для якого спектр частот не принциповий. Зокрема це демонструє існування Lo-Fi як музичного напрямку, що проявляє себе одразу в багатьох жанрах. Як можна зрозуміти з назви (low fidelity – низька точність, якість), його специфіка полягає в толеруванні й заохочувані деяких дефектів кінцевої аудіодоріжки. Одним із проявів є спроби в деяких композиціях повторити ефект, отриманий після багаторазового переписування звукових доріжок з однієї магнітної касети на іншу, що спричиняє в першу чергу втрату високих частот і заміну їх шумом [15]. Прояви феномену можемо бачити в електронній музиці різних періодів. Так, наприклад, роботи музичного виконавця Beck, в яких спостерігаються елементи Lo-Fi, датовані початком 2000-х років [15], тоді як поштовх таким жанрам, як фонк (Phonk), що характеризується зокрема спотвореними ударними, які можна розглядати як прояв Lo-Fi, дали події початку другого десятиріччя XXI сторіччя [16]. Сам же напрям було вперше охарактеризовано в 1950-х [15]. Вищесказане дає зробити висновок, що навіть враховуючи якісні обмеження, отримані з використанням AudioLDM 2 зразки можуть бути релевантними для сучасних музикантів.

До того ж, інтеграція такої моделі в VST-модуль спростила б створення в майбутньому подібних програмних продуктів із її альтернативами, якщо вони показуватимуть кращі результати.

Сучасні text-to-sound моделі використовують підходи, аналогічні таким, що дозволяють генерувати зображення на базі текстового вводу. Так, наприклад, сімейства моделей Diffsound і AudioLDM, спираються на принцип дифузії, що використовує, як можна зрозуміти з назви, сімейство text-to-image моделей Stable Diffusion [8], [17], [18]. Моделі, що використовують дані підходи, продемонстрували високе різноманіття вихідних результатів. Потреба в такій універсальності, безперечно, є в галузі написання музики: саме нею можна пояснити існування таких сервісів, як Freesound, де можна знайти бажаний аудіозразок і використати у власній композиції [19]. Проблеми, які породжують такі сервіси, аналогічні тим, які породжують сервіси-бази даних безкоштовних фотографій: не завжди реально знайти матеріал, потрібний користувачу. Як альтернатива, пропонуються веб-застосунки з можливістю створення бажаного зображення самостійно (як-от DALL-E, інтегрований в веб-портал ChatGPT). Безперечно, для генерації світлин за текстовим описом такий спосіб взаємодії з користувачем є найбільш універсальним, проте в випадку створення музики ситуація інша: існує загальноприйнятий формат доповнень до цифрових звукових робочих станцій VST (Virtual Studio Technology) [20]. Його наявність дозволяє інтегрувати такий сервіс у середовище, де кінцевий користувач хоче використовувати результуючий звук – тобто в саму ЦЗРС.

Вищесказане дозволяє говорити про можливість та актуальність створення генератора звукових зразків на базі технологій text-to-sound у вигляді VST-модуля.

2.2 Огляд підходів до комунікації з моделлю

2.2.1 Загальна характеристика

Для визначення інших технологій, які можуть знадобитися під час створення VST-модуля, необхідно обрати підхід інтеграції в нього обраної моделі. Тут наявні два варіанти:

а) вбудувати AudioLDM 2 безпосередньо в доповнення до ЦЗРС

б) розмістити модель на сервері й комунікувати з нею, наприклад, через програмний інтерфейс, побудований за принципами REST.

Розглянемо переваги та недоліки кожного підходу. У випадку інтеграції моделі в ЦЗРС кінцевий користувач буде незалежним від наявності доступу до мережі Інтернет під час використання програмного модуля. Проте в такому разі йому необхідні певні значні ресурси для запуску нейронної мережі на локальному комп'ютері.

2.2.2 Дослідження перспективності підходів

Під час тестів AudioLDM 2 з використанням наданої платформою HuggingFace Spaces віртуальної машини на базі Nvidia L40S було отримано середній час у 16.0 с, необхідний для генерації одного звукового зразку. Для моделювання середньостатистичної графічної карти користувача й порівняння її з Nvidia L40S було проаналізовано 57 найпопулярніших портативних відеокарт, розрахованих на встановлення в ноутбуки. Розглядалися саме портативні відеокарти з міркувань високоймовірної відсутності електромережі у місцях відсутності доступу до мережі Інтернет [21, с. 2], [22], де буде перевагою розташування моделі на користувацькому пристрої. Для них було розраховано середній бал у сервісі оцінки обчислювальної потужності відеокарт 3Dmark. Отриманий середній результат – 1183.2. Найближчим відповідником є Nvidia GeForce1660 TI (1184 бали). Для порівняння між собою цих двох графічних карт було використано сервіс оцінки, в базі даних якого наявні результати проходження тестувань обома пристроями.

В PassMark Nvidia L40S демонструє результат у 20022 бали, тоді як 1660 TI – лише 10160. На перший погляд, різниця не є критичною, якщо припустити

лінійну градацію продуктивності на поставленій задачі згідно з отриманими балами, проте потрібно наголосити на деяких нюансах:

а) даний результат було отримано в умовах надання пристроям всієї необхідної електричної потужності, що в вищезгаданих умовах недоступності електромережі можливо лише на короткий термін часу. Так, наприклад 1660 TI на піці навантаження вимагає 80 Вт [23], що безперечно є проблемою для ноутбуків з батареєю, наприклад, ємністю 50 Вт*год, Такі умови означали б можливість підтримання навантаження протягом 38 хв, навіть без урахування роботи інших компонентів пристрою.

б) не враховано той факт, що відсутність достатньої кількості відеопам’яті чи інших ресурсів відеокарти може спричинити суттєві часові затримки в порівнянні з роботою оптимальних для задачі пристроїв (ефект, відомий як bottleneck)

в) цифрові звукові робочі станції є ресурсовибагливими, особливо враховуючи розміри деяких музичних проєктів (кількість партій, накладених на них ефектів тощо). Зокрема, часто задіяна велика кількість ресурсів процесора [24] – пристрою, який також відповідає за завантаження відеокарти. Відповідно, ми не можемо гарантувати безперешкодну одночасну роботу моделі AudioLDM 2 і ЦЗРС.

г) 3Dmark враховує ефективність роботи відеокарти з точки зору результатів, продемонстрованих на графічних тестах (3Dmark Steel Nomad DX12) [25]. Хоч отримані бали і надають можливість робити певні припущення щодо ефективності відеокарти на задачах, пов’язаних із моделями штучного інтелекту, не слід забувати, що це саме припущення, а не факти.

Враховуючи вищесказане, було прийняте рішення на даному етапі розробки програмного продукту використовувати клієнт-серверний підхід для забезпечення

взаємодії програмного модуля для цифрової звукової робочої станції з обраною text-to-sound моделлю.

2.3 Огляд досліджених технологій

2.3.1 Text-to-sound модель

Враховуючи популярність проблеми генерації медіа (звуку, зображень тощо), а також можливість використання одних підходів одразу для багатьох типів медіа, є природнім той факт, що наразі існує багато рішень, що дозволяють створювати звук за текстовим чи іншими запитами. Серед них:

а) рішення на основі генеративних змагальних мереж (GAN). Особливістю змагальних моделей є той факт, що одночасно тренуються дві мережі: одна генерує деякий зразок, а інша – намагається визначити, чи цей зразок отриманий штучно, чи ні [26]. Хоч такі моделі в певних умовах можуть показувати задовільні результати, вони стикаються з труднощами, пов’язаними з нестабільністю наслідків тренування, повторюваністю звуків тощо [17]. Авторегресійні моделі та моделі на базі дифузії демонструють кращі результати [17], [27], через що цю категорію рішень було відкинуто.

б) рішення на основі авторегресії – використовують авторегресію для врахування попередніх результатів (наприклад – фрагментів аудіодоріжки, що відповідають деякому кадру відео) у поточному результаті [27, с. 4] одразу на декількох етапах перетворення вхідного сигналу у аудіо. Так, у [27] вона використана і на етапі створення шифру, і на етапі декодування в спектрограму. Цей підхід демонструє задовільні результати в таких сферах, як генерація звуку на базі відео [27], проте після аналізу можливостей його застосування в роботі над Diffsound, стало зрозуміло, що для задачі генерування звуку на базі тексту він показує себе гірше, ніж дискретна дифузія [17].

в) рішення на основі дифузії – будують процес тренування моделі на основі деформування вхідних даних шумом на базі деякого розподілу (як-от розподілу Ґауса) і спроб їх відновлення [17]. На момент вибору text-to-sound моделі, найкращі результати демонструє AudioLDM 2, що комбінує підхід дифузії з прихованими шарами. Такі шари використовуються для виділення вторинних ознак і зменшення обсягу інформації. Зокрема, прихований шар на етапі тренування будується кодувальником на базі трипросторового представлення звукового зразка (спектрограми) з урахуванням нелінійності сприйняття різниці в частотах людиною [8], яку можна зобразити на мел-шкалі (mel-scale) [28]. Такий підхід дозволяє покращити швидкодію моделі без втрати якості в порівнянні, наприклад, з Diffsound, що не використовує приховані шари і працює безпосередньо з мел-спектрограмами. Також це дозволяє уникнути іншу проблему: Diffsound використовує ланцюги Маркова замість розподілу Ґауса через роботу з категоріальними токенами замість неперервних величин [17]. За рахунок роботи кодувальника, який переводить мел-спектрограму й текстові дані в матрицю прихованого шару з неперервною областю визначення елементів, і декодувальника, який проводить зворотню роботу, у випадку моделей сімейства AudioLDM є можливим безпосереднє використання нормального розподілу [8].

Показовими є результати порівняння роботи Diffsound і AudioLDM з використанням відстані Фреше, що вираховувалася між взятим з набору даних AudioClap аудіозразком і згенерованим за його текстовим описом звуком. В таких умовах Diffsound продемонстрував у 2 рази гірший результат, ніж AudioLDM (50.40 проти 24.26) [8]. Беручи до уваги також аналогічну перевагу Diffsound над авторегресійними моделями [17] і наочні результати суб’єктивних експертних оцінок [8], можемо прийти до висновку про якісну перевагу сімейства AudioLDM

над альтернативами. Дослідним шляхом була також продемонстрована перевага AudioLDM в обсягах необхідних для роботи обчислювальних потужностей [5].

У свою чергу, AudioLDM 2, базуючись на архітектурі AudioLDM, але демонструючи суттєві нововведення, показує пропорційні ним покращення результатів попередника за багатьма метриками [29].

Враховуючи найкращі результати серед досліджених text-to-sound моделей, а також чітку орієнтованість на поставлену задачу, було прийняте рішення використати AudioLDM 2 як модель генерації звукових зразків за текстовим запитом.

2.3.2 Особливості архітектури моделей сімейства AudioLDM

Під час роботи модель AudioLDM використовує кодувальник, отриманий за допомогою механізму контрастивного переднавчання аудіо та мови (“contrastive language-audio pretraining”) на базі мовної моделі BERT для кодування текстового запиту користувача в багатовимірний вектор. Його простір є спільним для кодувальників звуку й тексту, що дозволяє використовувати для тренування AudioLDM не пари текстового опису й відповідного звукового зразка, а лише звукові зразки. Такий підхід продемонстрував кращі результати [8].

Модуль прихованої дифузії побудовано на базі U-Net – згорткової мережі, розробленої для сегментації зображень [30]. Архітектуру U-Net також використовує модель генерації зображень на базі текстового вводу StableDiffusion [5]. Модуль прихованої дифузії ініціює вектор багатовимірного простору, використовуючи розподіл Ґауса й заповнюючи його шумом. Цей багатовимірний простір після декодування представлятиме мел-спектрограму кінцевого аудіозразка. Сам же простір суттєво менший за простір мел-спектрограми [8]. На

кожному кроці модуль прихованої дифузії поступово виділяє з-поміж шуму звук, що співставляється з закодованим текстовим запитом користувача.

На кожному етапі очищення шуму генерується два багатовимірних вектори прихованого шару: один з використанням текстового запиту користувача, другий – без нього. З них, у вигляді суми, зваженої ступенем керування, формується кінцевий для цього етапу вектор.

Після завершення ітеративного процесу очистки шуму, з вектору прихованого шару згорткова модель-декодувальник вибудовує мел-спектрограму, яка передається моделі-вокодеру Hi-Fi GAN для конвертації в аудіофайл [8].

AudioLDM 2 доповнює отриманий конвеєр обробки новими елементами, а також видозмінює старі для отримання кращих результатів.

По-перше, модель здатна працювати з багатьма форматами вхідних даних на етапі виводу (inference). У багатовимірний вектор фіксованих розмірів текстовий, звуковий і голосовий кодувальники кодують відповідні вхідні дані [29].

По-друге, додається ще одна операція над вхідними даними перед наданні їх блоку дифузії. Отриманий вектор обробляє модель на базі GPT-2, що трансформує його в послідовність 768-вимірних векторів, яку дослідники називають «мовою аудіо» [29].

По-третє, в AudioLDM 2 вносить зміни в архітектуру блоку дифузії. Між кожними двома блоками U-Net знаходиться така кількість блоків трансформера, яка відповідає порядковому номеру останнього з двох блоків U-Net [29].

2.3.3 Архітектура кінцевого продукту

2.3.4 Інструменти для створення програмного доповнення формату VST

Задача забезпечення клієнт-серверної взаємодії в доповненні до ЦЗРС формату VST не є тривіальною для розробника програмного забезпечення, пов'язаного з аудіо. Відповідно, підтримка мережевої комунікації стала б перевагою обраного для розробки фреймворку. До того ж, враховуючи специфіку завдання, можна очікувати необхідність вирішення інших нетипових задач, таких, як, наприклад, проблема асинхронності запитів: синтезатори здебільшого передбачають однопотокове синхронне виконання інструкцій з синтезу й обробки звуку. У цій ситуації могла б допомогти популярність обраного інструменту, яка б вказувала на активність форумів та приділення уваги підтримці документації й функціоналу (з урахуванням запитів користувачів).

Під цей опис підпадає JUCE. Будучи фреймворком на базі об'єктно-орієнтованої мови C++, він пропонує великий набір вбудованих функцій (як-от інструментарій для роботи з HTTP-запитами, аудіофайлами, а також їхніми фрагментами), достатню гнучкість завдяки контролю за роботою програми на низькому рівні та популярність. Зокрема, фреймворк використовується такими компаніями, як Adobe, Google, Arturia, Sennheiser, Steinberg (власники стандарту VST) тощо [31]. Саме за допомогою нього було в результаті створено демонстраційний програмний продукт.

2.3.5 Розгортання моделі

Враховуючи обрання клієнт-серверного принципу взаємодії моделі та модуля, є необхідність обрати розміщення AudioLDM 2 під час розробки та під час потенційного функціонування кінцевого програмного модуля. Постає згадана проблема нестачі обчислювальних ресурсів, що не дозволяє використовувати

модель локально навіть на етапі розробки демонстраційного зразка. Вона спонукає залучити засоби хмарного розгортання (cloud hosting), такі, як HuggingFace Spaces. Ця платформа орієнтована на надання ресурсів для демонстраційних прикладів моделей штучного інтелекту [32], що однозначно спрощує задачу, адже такий напрям діяльності збігається з нашими потребами. До того ж, на платформі вже розміщений демонстраційний приклад AudioLDM 2, що дозволяє почати розробку простого серверу з фундаменту попередніх зусиль.

При необхідності розгортання моделі для реальних, недемостраційних задач можливе використання такого сервісу, як Runpod – хмарного рішення, що пропонує обчислювальні потужності (зокрема – відеокарти) для тренування й подальшого функціонування моделей штучного інтелекту. Runpod, зокрема, пропонує швидке автомасштабування [33], необхідне в випадку потенційного масового використання продукту.

Незважаючи на можливість комунікації напряму зі Spaces, використовуючи прості HTTP-запити, було прийняте рішення про створення додаткового сервера-посередника, який буде брати на себе відповідальність за утримання актуальних засобів авторизації зі Spaces, а пізніше – з Runpod або іншими аналогічними платформами, а також аутентифікації користувачів. Враховуючи ресурсоємність і як наслідок – значні часові витрати завдання синтезу звукового зразка (близько 20.65 с, що отримано дослідним шляхом з використанням віртуальної машини на базі Nvidia A10G), накладні витрати на відправку додаткових HTTP-запитів не змінюють загальної ситуації кардинально.

2.3.6 Використання великих мовних моделей для покращення результатів

На вхід AudioLDM 2 рекомендується подавати запит з якомога більш детально описаними характеристиками кожного об’єкту, тобто якомога більшою кількістю уточнюючих епітетів [34]. Задачу максимальної конкретизації запиту,

безперечно, можна покласти на кінцевого користувача, проте з наявністю великих мовних моделей ми отримуємо можливість її автоматизувати.

Враховуючи перевагу клієнт-серверної архітектури, аргументовану апаратними вимогами AudioLDM 2, можемо доступатися до обраної LLM за таким самим принципом. Цей спосіб взаємодії дозволить нам обрати велику мовну модель, беручи до уваги в першу чергу відповідність результату поставленій задачі, а не кількість необхідних ресурсів. Через тривіальність задачі, простоту підключення, попередній досвід з великими мовними моделями сімейства Gemini [14] і кращі результати в тестах порівнянні з попередниками, для автоматичного покращення запитів користувачів було обрано модель Gemini 1.5 Flash.

Аналогічно з AudioLDM 2, підключення великої мовної моделі здійснюється через сервер-посередник, що дозволяє надійно приховати ключі доступу від кінцевого користувача.

2.3.7 Способи оцінки результатів

[8], [29] надають комплексні оцінки роботи моделей сімейства AudioLDM. Об’єктивна складова оцінки надана з використанням відстані Фреше та інших способів вимірювання похибки. Суб’єктивна ж складова спирається на результати опитування. Інтеграція моделі в програмний продукт за клієнт-серверним принципом не впливає на роботу моделі text-to-sound, тобто ця комплексна оцінка є актуальною для демонстраційного зразка. Тим не менш, з деякими аудіозразками пропонується ознайомитися в репозиторії для формування власної суб’єктивної оцінки.

Важливою новою об’єктивною оцінкою, яку можна надати отриманій системі, є часова. Це було зроблено з урахуванням потенційного покращення

обчислювальних потужностей у випадку розгортання продукту для масового використання.

2.4 Особливості реалізації

2.4.1 Модернізація демонстраційного зразка на платформі HuggingFace

Наявність демонстраційного робочого середовища AudioLDM 2 на HuggingFace суттєво допомогла на ранніх етапах розробки. Цей зразок неможливо було б використати в роботі, якби не той факт, що в його основі лежить Gradio - фреймворк для побудови програм, що перш за все мають модель штучного інтелекту як центральний елемент [20]. Однією з важливих його особливостей є той факт, що будь-який програмний продукт передбачає комунікацію через автоматично згенерований RESTful API [35]. Саме це й було використано під час розробки.

В подальшому програму було модифіковано під задачі створюваного модуля: за замовчуванням генерувався не просто аудіозразок, а відео з простою амплітудно-частотною характеристикою. Було внесено відповідні зміни для уникнення подвійної конвертації з формату .wav до формату .mp4 в середовищі з моделлю і назад з формату .mp4 до формату .wav на сервері-посереднику.

До того ж, було додано можливість регулювати кількість кроків зменшення шуму [8]. Цей параметр не був доступним для редагування у демонстраційному зразку, де він зафіксований на значенні 200. Проте, враховуючи часову складність завдання генерації аудіозразка, а також використання результуючого аудіо в творчості, де за певних згаданих вище причин, таких як стиль композиції, штучно знижена якість може стати в нагоді, було прийняте рішення надати користувачу контроль за кількістю кроків зменшення шуму.

2.4.2 Можливості налаштування програмного доповнення

Описуваний програмний продукт поєднує в собі функціонал інтерфейсу моделі штучного інтелекту й програмного доповнення, спеціалізованого на роботі з готовими аудіозразками (семплера). Відповідно, для забезпечення його актуальності, він має надавати гнучкість кожного з цих інструментів.

У зв'язку з вищесказаним, було прийняте рішення розділити інтерфейс на два логічні блоки налаштувань:

а) блок налаштувань кінцевого звучання надає можливість вирізати бажаний фрагмент з отриманого аудіозразка, а також набір параметрів обвідної ADSR ("attack, decay, sustain, release"), необхідних для опису тембру віртуального інструменту [36]. Інші параметри звуку вважаємо додатковими й необов'язковими в реалізації. Для обґрунтування було проведено аналіз можливостей Avid Pro Tools – найпопулярнішої на даний момент цифрової звукової робочої станції [37], з якого можна зробити висновок, що всі інші ефекти, як-от exo (reverb, delay), гучність тощо, можна застосувати до звукової доріжки шляхом побудови ланцюга ефектів [38].

б) блок налаштувань моделі надає можливість редагувати роботу AudioLDM 2 залежно від потреб користувача. Випадкове початкове значення (random seed) дає впливати на результат псевдовипадкових операцій моделі. Зокрема, його зміна може покращити якість вихідного аудіозразка на конкретному пристрої, на якому запущена модель [39]. Роль ступеня керування ("guidance scale") подібна до ролі температури великих мовних моделей [14]: більше значення спонукає модель генерувати зразок, який точніше описує заданий текст, проте з потенційними втратами якості. В разі, якщо параметр "number of candidates" має значення більше за 1, генерується вказана кількість аудіозразків, після чого з них обирається той, який демонструє меншу оцінку втрат у спільному текстово-

звуковому векторному просторі, використовуваному кодувальником і декодувальником CLAP. До того ж, наявна можливість редагувати негативний текстовий запит, що вказує моделі, які поняття на цьому векторному просторі потрібно оминати [40].

2.4.3 Особливості збереження аудіозразків

Одним із очевидних викликів, який стоїть перед будь-яким доповненням до цифрової звукової робочої станції, є збереження стану для забезпечення можливості переривати роботу над композицією.

Завдяки тому факту, що аудіозразки зазвичай завантажуються з файлової системи користувацького пристрою, у програмних модулів, що дозволяють відтворювати аудіофайли (так званих семплерів) є можливість зберігати лише шлях до них. В нашому ж випадку гарантоване збереження файлу на пристрої користувача за межами програмного модуля не передбачається, адже в такому разі на ньому залишатиметься велика кількість не потрібних музиканту аудіозразків. Отже, через необхідність зберігати весь файл в документі формату XML, було прийняте рішення зробити це з використанням кодування Base64. Після імплементації такого підходу, втрати часової ефективності роботи програми не спостерігалось.

2.4.4 Допомога початківцям

Враховуючи потенційну цільову аудиторію програмного модуля, серед якої можуть опинитися люди, не знайомі з базовими принципами роботи з аудіозразками з використанням стандартних інструментів ЦЗРС, а також такі, які не мали досвіду з налаштування нейронних мереж, було надано необхідні короткі пояснення щодо кожного параметру, на який користувач може вплинути.

Беручи до уваги наявність комунікації модуля з великою мовною моделлю, було розглянуто також можливість її використання для допомоги користувачу, проте враховуючи невелику кількість інструментів впливу користувача на результат, цю можливість було відкинуто для цієї конкретної задачі.

2.5 Огляд результатів. Потенційні покращення

Рисунок 2.1 – Інтерфейс користувача програмного продукту

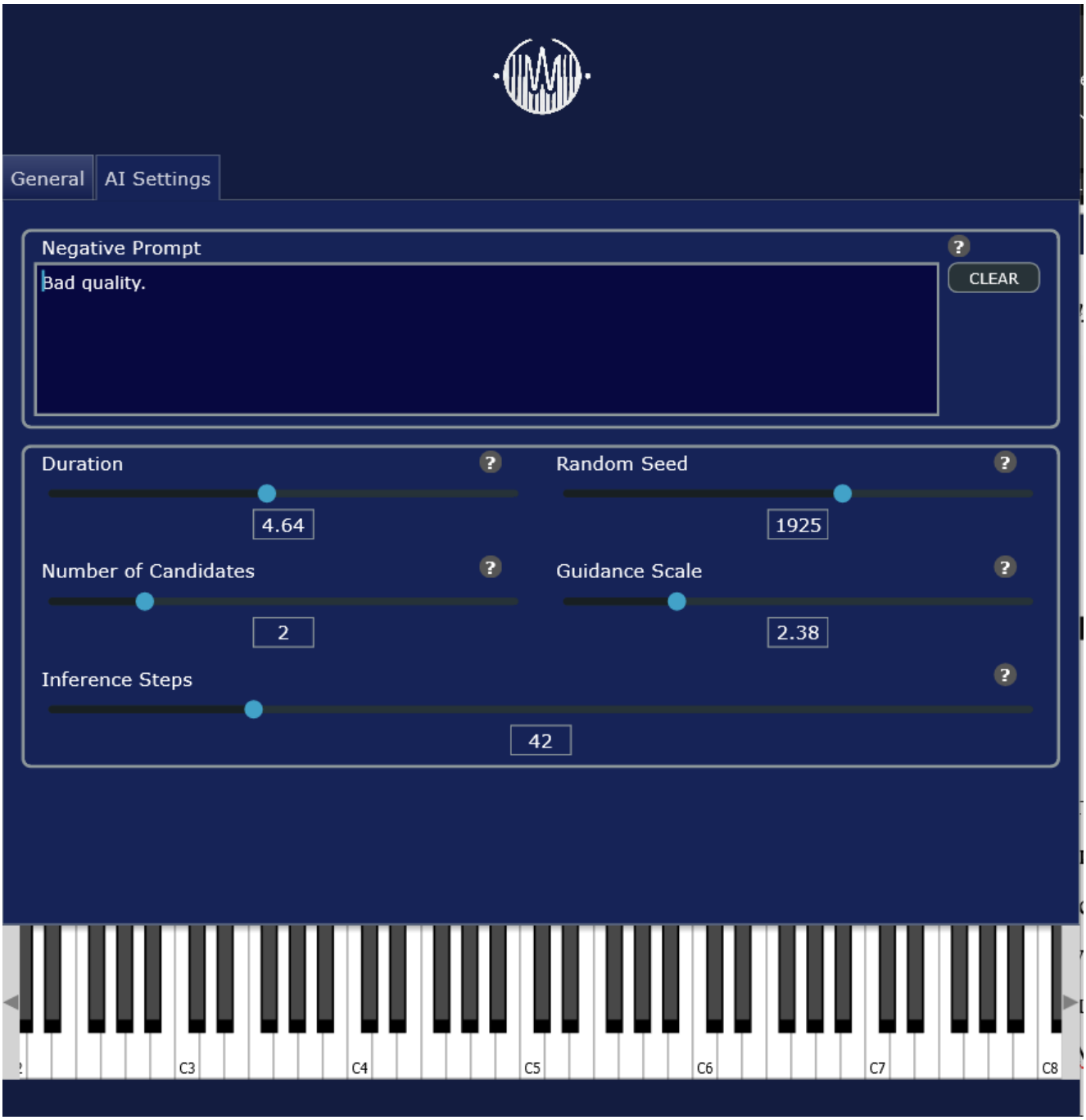

Рисунок 2.2 – Додаткові налаштування text-to-sound моделі, доступні користувачу

Отриманий в результаті розробки демонстраційний програмний модуль представляє собою програму в форматі VST3, що підтримується більшістю цифрових звукових робочих станцій та дозволяє синтезувати необхідні звукові фрагменти безпосередньо в середовищі для написання музики. Зокрема, його підтримує Cubase – друга за популярністю на даний момент ЦЗРС [37], [41]. У випадку ж Avid Pro Tools, на допомогу приходить використання відповідних програмних модулів-адаптерів, таких, як SigMod [42].

В ході тестування було виявлено, що часові витрати кінцевого програмного продукту на генерацію звукового зразка є задовільними на відносно невеликих обчислювальних потужностях. Так, наприклад, на віртуальній машині HuggingFace Spaces з Nvidia A10G на стандартних для демонстраційного зразка налаштуваннях моделі середня часова витрата на генерацію аудіофрагмента склала 20.65 с, що, враховуючи релевантність отриманого звукового зразка [8], однозначно є швидшим за пошук необхідного звуку в онлайн-бібліотеках. Порівнюючи кількість можливих операцій на секунду на тензорних ядрах A10G і RTX4090 (відеокарти, створеної для користувацького сегменту), можна побачити перевагу останньої (1000 TOPS проти 1321 TOPS) [43], [44]. Безперечно, ми не можемо говорити про вищу продуктивність лише на базі цього факту, проте активне залучення до генерації графічного контенту штучного інтелекту відеокартами Nvidia лінійки RTX [45], [46], а також той факт, що самі варіації моделі AudioLDM були натреновані на Nvidia RTX 3090 [8], вказує на можливість успішного використання відеокарт, доступних пересічному громадянину.

Отже, вже зараз є певна категорія користувачів, які потенційно можуть використовувати даний модуль без потреби в хмарних обчисленнях. Це спонукає, незважаючи на невелику чисельність представників цієї категорії, дослідити

можливість локального розміщення моделі на обчислювальних пристроях верхнього сегменту ринку.

Як і очікувалося, якість кінцевих аудіозразків можна покращити. Мова йде про контекстуальні прогалини, як із певними звуками автомобілів, наприклад (див у додатках), а також згадану проблему відсутності сигналу на високих частотах. Відповідно, у розробленому каркасі демонстраційного проєкту в майбутньому можлива заміна AudioLDM 2 на іншу модель, що б забезпечила кращі результати. Можливе також дотренування AudioLDM 2 та повторне використання деяких її елементів.

Однією з переваг аналогічних застосунків для генерації зображень на базі штучного інтелекту є можливість ведення діалогу з моделлю й відповідно збереження сесії. Напрямок надання такої можливості однозначно варто дослідити, адже він суттєво розширив би можливості користувача щодо отримання бажаного звукового зразка.

Загалом, незважаючи на згадані потенційні покращення, отриманий результат демонструє перспективність напрямку. Уже на даному етапі програмний зразок є корисним продуктом, що може пришвидшити й зробити більш ефективним пошук бажаного звукового зразка для використання в композиції.

3 Інтеграція великої мовної моделі в віртуальний інструмент-синтезатор

3.1 Аргументація актуальності

Як було продемонстровано в попередньому розділі, сучасні наробки в сфері синтезу звуку на базі текстового вводу користувача, такі як AudioLDM 2, мають ряд недоліків. Серед них – неможливість вдосконалення звуку, непостійна якість отриманих результатів й обмежені можливості щодо опису бажаного. Всі ці проблеми спонукають до дослідження перспектив використання засобів штучного інтелекту для отримання очікуваного користувачем звучання шляхом налаштування існуючих інструментів-синтезаторів.

Великі мовні моделі є підвидом фундаментальних моделей. Завдяки навчанню на максимально різноманітних неструктурованих даних, вони демонструють виняткову універсальність [14]. Ця універсальність, наявність в мережі великої кількості інформації з приводу теорії синтезу звуку (зокрема – 291 млн результатів за запитом “sound design tutorial” знайдених пошуковим рушієм Google [47]), а також здатність великих мовних моделей генерувати й сприймати великі обсяги тексту чи документів певних форматів, може стати в нагоді при роботі з синтезаторами. Такі моделі могли б отримати інформацію про поточний стан інструмента, текстове пояснення бажаної зміни звучання від користувача, і скоригувати параметри так, щоб очікуваний ефект був досягнутий.

Типовий сучасний синтезатор є складним механізмом, що вимагає часу й зусиль для опанування. Навіть враховуючи велику кількість інструкцій та документації в мережі (11 млн результатів за запитом “synthesizer user manual”, знайдених пошуковим рушієм Google [48]), шаблони налаштувань до синтезаторів залишаються популярними й потрібними. Зокрема, 70 відсотків респондентів опитування, проведеного KVR Audio, використовують передустановки під час написання музики [49]. Наявність можливості пояснити очікуване результуюче

звучання в вигляді текстового повідомлення та власне отримати таке звучання фактично можна порівняти з наявністю необмеженої кількості шаблонів налаштувань. Це б суттєво пришвидшило роботу з синтезаторами для досвідчених користувачів, дозволивши пропустити рутинне налаштування інструменту до звуку, на базі якого вони бажають експериментувати, і покращило б результати початківців, які мали б можливість знаходити нові звучання й поступово опановувати основи синтезу звуку, щоб надалі отримувати бажані результати швидше.

З іншого боку, не можна сказати, що такий підхід буде кращим в усіх аспектах за попередній, описаний у цій роботі. Призначення синтезаторів не полягає в імітації органічно отриманих звуків, хоч їх і можна використовувати для цього до певної міри: навпаки, вони були створені як неконвенційні інструменти, здатні створювати унікальний аудіосигнал [50]. Відповідно, наявність модуля формату VST з інтегрованою великою мовною моделлю покривала б певну нішу: пропонуючи часто вищу якість та модифікованість за готові аудіозразки, він би міг використовуватися для написання партій, що не мали б на меті відтворення звуків реального життя.

3.2 Огляд досліджених технологій

3.2.1 Великі мовні моделі

Архітектура сучасних LLM відрізняється від класичного трансформеру. Зокрема, тепер їх складно назвати мовними, адже вони підтримують не тільки такий формат введення даних [51]. У роботі нас не цікавлять можливості MM-LLM (мультимодальних великих мовних моделей) щодо технологій вводу даних користувачем (як-от звукових доріжок, зображень, точок тривимірної площини тощо) [51], проте такі можливості та їхнє використання в синтезі звуку можуть бути досліджені в майбутньому.

Зміни відбулися не тільки на етапі введення даних, а й на глибшому рівні. В основі досі лежить трансформерна архітектура – ланцюг блоків обрахунку уваги та мереж прямого поширення, з нормалізацією значень між ними. Увага в цьому контексті є коефіцієнтом значущості кожної пари вхідного токена та попередньо згенерованого вихідного токена в наступному згенерованому токені [52]. Відмінності полягають у залученні параметрів отриманої мережі до генерації відповіді на запит. Так, наприклад, сімейство моделей Gemini 1.5 використовує принцип суміші експертів [53]. Він полягає в тому, щоб розбити велику трансформерну модель на підмножини параметрів (експертів), після чого натренувати функцію-перемикача, що буде підбирати ланцюг експертів для вирішення конкретних задач від користувача [54]. Це дозволяє збільшити загальну кількість параметрів моделі, не збільшуючи кількість параметрів, активованих деяким запитом.

Залучення лише частини параметрів до генерації результату, а також використання трансформерної архітектури, яка через відсутність згорткових шарів сприяє простоті паралелізації [52], сприяє тому, що сучасні великі мовні моделі, за умови наявності достатніх обчислювальних потужностей, працюють винятково ефективно.

Вибір великої мовної моделі напряму залежить від кінцевої комунікації між нею та синтезатором: чи буде вона відбуватися безпосередньо в програмному модулі, чи модель буде розміщена на сервері. Безперечно, локальне розміщення моделі передбачає наявність у користувача можливості запустити її паралельно зі своєю цифровою звуковою робочою станцією. Проте така можливість буде лише в невеликої категорії користувачів, враховуючи ресурсовибагливість процесу написання музики. Зокрема, саме про неї говорять рекомендації щодо обчислювальних потужностей, наданих ImageLine до ЦЗРС FLStudio [24].

Враховуючи вищесказане, згаданий рівень продуктивності середньостатистичного пристрою, що є у власності користувача, а також системні вимоги для локального запуску великих мовних моделей (на прикладі Deepseek [55]), для експериментальної реалізації було використано клієнт-серверний механізм взаємодії. Він дозволив обрати велику мовну модель, що безпосередньо виконує поставлену задачу, а не компромісний варіант, що буде здатний працювати на обмежених ресурсах кінцевого користувача.

В ході дослідження було визначено модель Gemini 1.5 Flash як найлегшу для інтеграції в кінцевий продукт через доступний прикладний програмний веб-інтерфейс, а також таку, що демонструє задовільні результати в тестах.

Зокрема, це обґрунтоване позитивним досвідом використання Gemini 1.0 Pro для генерації тестів з англійської мови початкового рівня [14]. Розв’язувана моделлю задача подібна до тої, яка була поставлена Gemini 1.0 Pro: необхідно згенерувати документ певного формату за шаблоном, врахувавши деякий текстовий опис користувача.

Gemini 1.5 Flash має показову різницю в результатах тестувань моделей на свою користь [56], а також задовільну для генерації тестів ефективність [14], більше контекстне вікно в 1 млн токенів [53], що може суттєво вплинути на результативність моделі, враховуючи необхідність передавати в деякому форматі інформацію про будову синтезатора.

3.2.2 Початковий синтезатор

Після перегляду наявних у вільному доступі синтезаторів з відкритим кодом, написаних за допомогою JUCE, було виявлено, що роль основи програмного модуля здатний виконати TapSynth, створений спільнотою The Audio Programmer в рамках відео-курсу з роботи з фреймворком. За рахунок останньої

деталі, його код не використовує нестандартних підходів до створення синтезаторів, є чистим, чітким і зрозумілим, а набір налаштувань віртуального інструмента – подібним за масштабом до деяких популярних програмних модулів, як-от Tyrell N6 [57], [58], що дозволяє оцінити реальні перспективи використання великої мовної моделі для роботи з параметрами синтезатора.

3.2.3 Інші використані технології

Аналогічно з синтезатором на базі AudioLDM 2, клієнтська частина застосунку написана за допомогою фреймворку JUCE, для серверної ж використано FastAPI. JUCE зарекомендував себе як універсальний інструмент, що підходить для розв'язання задачі комунікації з використанням HTTP запитів, а його популярність дозволяє знайти достатню кількість модулів-синтезаторів, написаних із його допомогою, що можуть бути використані в демонстраційному зразку. Створення ж сервера-посередника між прикладним програмним інтерфейсом MM-LLM і модулем аргументовано міркуваннями безпеки: інакше довелося б зберігати екземпляр ключа доступу до сервісу великої мовної моделі безпосередньо в файлах програмного модуля, що створює певні ризики з точки зору кібербезпеки.

3.2.4 Способи оцінки результатів

Аналогічно з програмою на базі AudioLDM 2, оцінити об'єктивно результативність роботи отриманого програмного модуля можливо лише з точки зору швидкодії. Натомість, суб'єктивну оцінку пропонується дати, ознайомившись із ним в репозиторії даної роботи.

3.3 Особливості реалізації

3.3.1 Робота з поточним станом синтезатора

Враховуючи специфіку взаємодії синтезатора з ЦЗРС, він має підтримувати збереження та відновлення свого поточного стану, щоб у свою чергу збереження музичних проєктів у форматі, підтримуваному звуковою робочою станцією, не призводило до втрат інформації. Застосовуваний для цього механізм можна використати, щоб передавати стан синтезатора LLM засобами REST API.

В JUCE за підтримку стану відповідає клас AudioProcessor, а точніше його нащадки, реалізовані користувачем [59]. Доступ до актуального стану може здійснюватися в будь-який момент роботи програми, що спрощує задачу його передачі моделі в зручному форматі.

Незважаючи на той факт, що JUCE передбачає збереження поточного стану в форматі XML [59], використання JSON для поставленої задачі є оптимальнішим. Причинами є граматична простота JavaScript Object Notation в порівнянні з XML, а також його більша популярність [60], що може покращити результати Gemini 1.5 Flash і зменшити кількість некоректно відформатованих відповідей за рахунок спрощення задачі й більшої кількості даних, на яких модель була натренована сприймати JSON і генерувати документи в цьому форматі. До того ж, для ще більшого спрощення задачі для великої мовної моделі, документ формується самостійно, у ньому зберігаються лише необхідні дані. Це також дозволило убезпечити програму у випадках, якщо отриманий від моделі JSON-документ не містить всіх ключів, що були в вхідному файлі. Вважаємо, що в такому разі параметри синтезатора, інформація про які не була вказана, мають лишитися незмінними.

3.3.2 Регулювання параметрів великої мовної моделі

Однією з можливостей покращити результати великої мовної моделі є налаштування її температури, а також параметрів topP і topK. Всі вони впливають на передбачуваність результату роботи моделі. Так, зміна температури спричиняє зміну розподілу ймовірності серед обраних токенів-кандидатів на наступну позицію в вихідному тексті, topP – обмежує суму імовірностей обрання токенів, що розглядається, деяким числом від 0 до 1, а topK – обмежує кількість таких токенів-кандидатів [61]. Можливість таких додаткових налаштувань надає використана бібліотека google.generativeai [62].

Враховуючи чіткі обмеження, що стоять перед моделлю в рамках даної задачі, можна говорити про потенційне покращення зменшенням температури й параметрів topP і topK для стабільнішого й більш передбачуваного результату.

В ході роботи програмного модуля не було виявлено дефектів вихідного результату, через що дані параметри не були модифіковані. Відповідно, використовувалися значення за замовчуванням: (T дорівнює 1, topP дорівнює 0.95, topK дорівнює 40). В разі продовження роботи над програмним модулем і розширення функцій синтезатора, слід брати до уваги вищесказане.

3.3.3 Допомога початківцям

Однією з задач створюваного програмного модуля є поступове навчання користувача синтезу звуку на реальних прикладах. Необхідною його функцією є наочна демонстрація змін, виконаних моделлю. Логічним кроком є залучення Gemini до двох згаданих процесів. В отриманому програмному доповненні шляхом коригування запиту до великої мовної моделі досягнуто вкладення короткого текстового пояснення до виконаних змін у вихідний JSON-документ.

3.4 Огляд результатів. Потенційні покращення

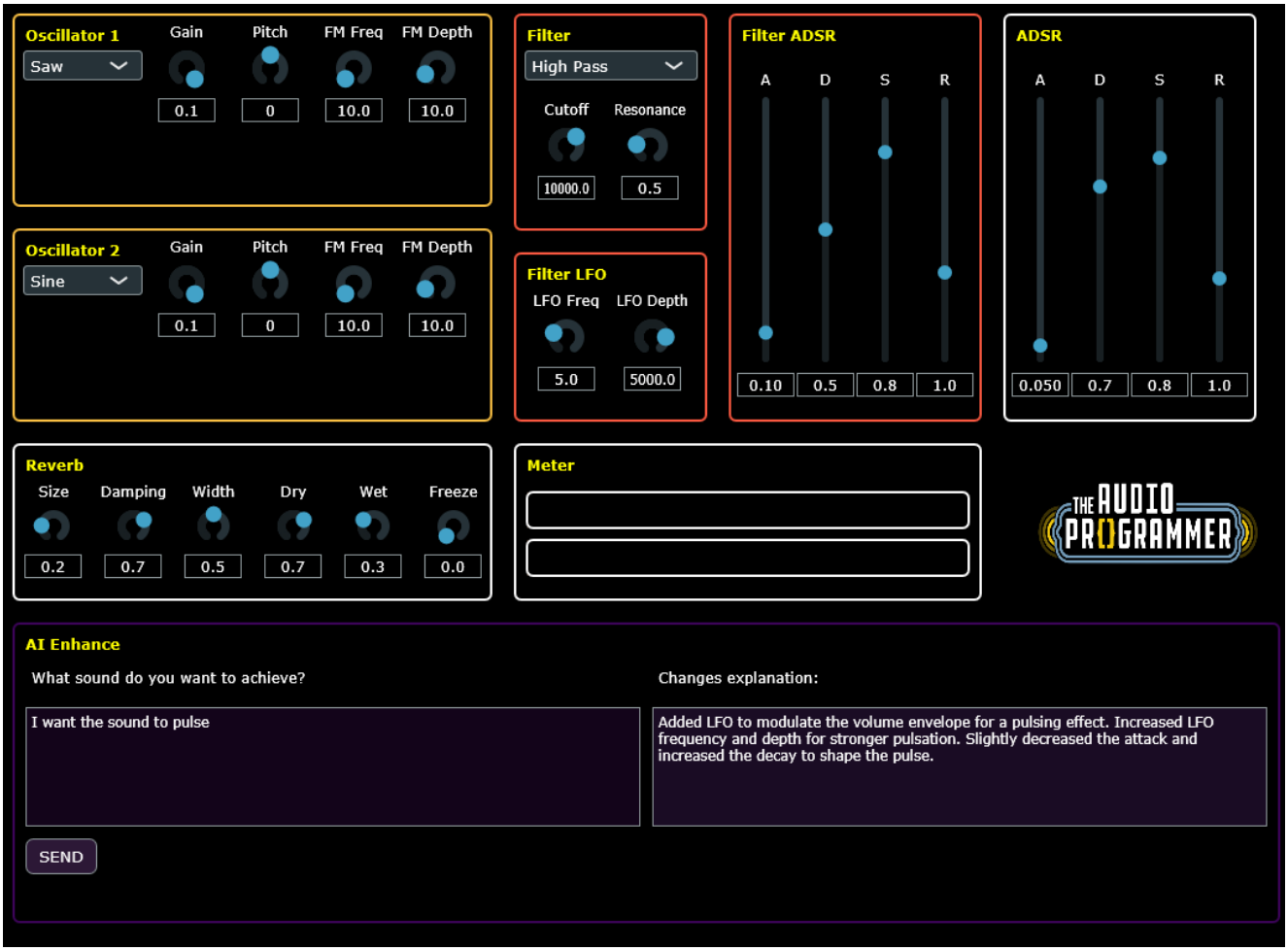

Рисунок 3.4 Інтерфейс користувача програмного продукту

Незважаючи на те, що використана Gemini 1.5 Flash не пропонує ефективність рівня “state of the art” галузі мультимодальних моделей [63], технологічна схема показала задовільні результати в роботі з різними текстовими запитами. Модель продемонструвала здатність виконувати як прості вказівки, як-от модифікувати конкретні вказані параметри ефектів, так і більш складні запити, як-от «відтворити звук, характерний для аркадних ігор 1980-х років».

Показовою також є швидкодія отриманої системи. Проведений з використанням 10 різних запитів експеримент показав, що середній час відповіді від Gemini 1.5 Flash, розгорнутої на серверах компанії Google, складає 7.48 с – суттєво менший за, наприклад, середній час, що необхідний для синтезу аудіозразку AudioLDM 2 на доступних обчислювальних потужностях.

Додаткові тести були проведені з використанням Gemini 2.5 Flash, що продемонструвала суб'єктивно порівнювані результати. Поведінка моделей та їхня реакція на конкретні запити частково відрізняється, проте ці відмінності в реакції не демонструють суттєвої переваги однієї над іншою на значеннях температури та параметрів topP і topK за замовчуванням. Натомість показовою була різниця в часовій ефективності, що може бути пов'язана з пріоритетами компанії Google або з безпосереднім розміром і складністю моделей. Так, на розглянутих прикладах Gemini 1.5 Flash генерує відповідь у 2 рази швидше за Gemini 2.5 Flash (8 с проти 16 с). З отриманими аудіозразками можна ознайомитися в репозиторії.

Отже, можна говорити про перспективність даного напрямку, зокрема — можливості збільшення набору налаштувань синтезатора та потужностей моделі для досягнення більшої варіативності й розширення спектру результуючих звукових сигналів. Проте окрім такого покращення отриманого результату, можна також дослідити перспективи його екстраполяції.

У використаному запиті Gemini не отримує жодного сталого, закріпленого в коді або контексті опису синтезатора, спираючись виключно на його поточний стан, отримуваний автоматично у вигляді JSON документу з кожним таким запитом. Це говорить про слабку зв'язаність компонентів кінцевої системи, а отже – їхню взаємозамінність, спонукає дослідити можливості створення програмного доповнення, в якому модель не просто інтегрована в деякий обраний заздалегідь на етапі розробки синтезатор, а під'єднується до існуючого програмного модуля, обраного користувачем.

З точки зору навчального потенціалу програмного продукту, покращенням стала б можливість у тому чи іншому вигляді порівняти стан синтезатора до і після змін, виконаних моделлю. Наразі користувач отримує короткі текстові пояснення таких змін, проте можливість візуально бачити редаговані моделлю елементи контролю суттєво покращила б його розуміння кореляції між параметрами та звучанням синтезатора. До того ж, в разі реалізації вільної

інтеграції моделі з обраними користувачем програмними модулями, доцільною була б опція розтлумачити теоретичну роль кожного їхнього параметра.

Отриманий в результаті програмний модуль показує, незважаючи на демонстраційний характер, високі результати й може уже на цьому етапі свого розвитку стати в нагоді музикантам.

4 Перспективи подальших досліджень

Таким чином, перспективними є декілька напрямків покращення розроблених програмних застосунків.

Проаналізовані в ході роботи text-to-sound моделі не показали достатньо високих результатів, щоб покрити весь спектр задач із синтезу аудіозразків, який постає перед сучасним музикантом. Це спонукає до покращення цих моделей та/або пошуку альтернатив. Аналогічно, більші великі мовні моделі (або їхні альтернативи) потенційно здатні керувати складнішими програмними доповненнями та робити це вправніше, особливо в разі дотренування (fine-tuning), наприклад, на великому об’ємі інструкцій до різного роду синтезаторів. До напрямку покращень, пов’язаних зі штучним інтелектом, також можна віднести надання можливості подальшого уточнення бажаного запиту користувача до text-to-sound моделей.

Іншим потенційним напрямком покращення є вдосконалення з точки зору навчального аспекту. Це стосується програмного доповнення на базі великої мовної моделі. Мова йде про наочну демонстрацію змін, зроблених MM-LLM, а також про використання її можливостей з мовного синтезу для генерації пояснень до ролі конкретних параметрів синтезатора у вихідному сигналі.

Найбільш перспективним є напрямок інтеграції великої мовної моделі із будь-яким обраним користувачем програмним модулем ЦЗРС. В разі успіху можливе застосування результатів даної роботи не тільки для синтезу звуку на базі всіх наявних синтезаторів, а й для його обробки (в разі інтеграції за аналогічним до продемонстрованого в роботі принципом LLM з програмними модулями ефектів).

Простішими в реалізації та водночас важливими для забезпечення користі продукту є останні два згадані напрями. Всі технології для забезпечення описаних покращень вже наявні на даний момент. Відповідно, в подальшій роботі будуть досліджуватися в першу чергу можливості інтеграції програмних доповнень до цифрових звукових робочих станцій із великими мовними моделями, а також їхнє використання в цілях спрощення опанування звукового синтезу музикантами.

Висновки

В ході даної роботи було досліджено способи інтеграції засобів штучного інтелекту з сучасним інструментарієм для написання музики з метою покращення або заміни існуючих засобів синтезу звуку. Для демонстрації досліджених підходів було побудовано програмне доповнення на базі text-to-sound моделі, а також – синтезатор з інтегрованою великою мовною моделлю.

Створені у рамках даної роботи демонстраційні програмні продукти показали перспективність включення засобів штучного інтелекту до сучасного інструментарію музиканта. Можливі покращення передбачають підвищення універсальності використаних підходів, зручності використання кінцевих продуктів шляхом перенесення моделей на пристрій користувача тощо.

Технічні обмеження, пов’язані як із можливостями сучасних text-to-sound моделей, так і з ефективністю роботи обчислювальних пристроїв середньостатистичного користувача, спонукають провести аналогічне дослідження повторно через певний період часу. Сучасна тенденція до стрімкого розвитку технологій, пов’язаних із штучним інтелектом [1], не може не сприяти змінам пріоритетів під час розробки нових обчислювальних пристроїв. Вона спричиняє суттєві зміни в галузі вже зараз [46]. Відповідно, ці зміни в перспективі відкриють нові можливості щодо інтеграції засобів штучного інтелекту із засобами синтезу звуку, або цілковитої їхньої заміни в цифрових звукових робочих станціях у майбутньому.

Список використаних джерел

[1] «(PDF) Forecasting the future of artificial intelligence with machine learning-based link prediction in an exponentially growing knowledge network», *ResearchGate*, Груд 2024, doi: 10.1038/s42256-023-00735-0.

[2] J. Copet *et al.*, «Simple and Controllable Music Generation», 30, Січень 2024, *arXiv*: arXiv:2306.05284. doi: 10.48550/arXiv.2306.05284.

[3] «Use AI in Photoshop to Streamline Your Workflow.» Дата звернення: 27, Квітень 2025. [Online]. Доступний у: https://www.adobe.com/products/photoshop/ai.html

[4] J. Nistal, S. Lattner, і G. Richard, «DrumGAN: Synthesis of Drum Sounds With Timbral Feature Conditioning Using Generative Adversarial Networks», 28, Червень 2022, *arXiv*: arXiv:2008.12073. doi: 10.48550/arXiv.2008.12073.

[5] R. Rombach, «High-Resolution Image Synthesis with Latent Diffusion Models», 2112.10752. [Online]. Доступний у: https://arxiv.org/pdf/2112.10752

[6] «Features | Cursor - The AI Code Editor». Дата звернення: 27, Квітень 2025. [Online]. Доступний у: https://www.cursor.com/features

[7] «How to Use Kits AI with Pro Tools: Innovating Your Production Process - Kits.AI». Дата звернення: 27, Квітень 2025. [Online]. Доступний у: https://www.kits.ai/blog/pro-tools-ai-voice-generators

[8] H. Liu *et al.*, «AudioLDM: Text-to-Audio Generation with Latent Diffusion Models», 09, Вересень 2023, *arXiv*: arXiv:2301.12503. doi: 10.48550/arXiv.2301.12503.

[9] M. Wereski, «The Threshold of Hearing», *steam*, вип. 2, вип. 1, с. 1–4, Вер 2015, doi: 10.5642/steam.20150201.20.

[10] «Sonic Charge - Synplant». Дата звернення: 03, Травень 2025. [Online]. Доступний у: https://soniccharge.com/synplant

[11] «The road to Synplant 2: the story behind this mind-bending plug-in sequel - CDM Create Digital Music». Дата звернення: 03, Травень 2025. [Online]. Доступний у: https://cdm.link/the-road-to-synplant-2/

[12] « ［PRESS RELEASE］ Sony CSL unveils “DrumGAN,” An AI-powered Drum Sound Generation Technology – Sony CSL». Дата звернення: 03, Травень 2025. [Online]. Доступний у: https://www.sonycsl.co.jp/en/news-articles/2022/06/22/18682/

[13] C. F. Hockett і C. D. Hockett, «The Origin of Speech», *Scientific American*, вип. 203, вип. 3, с. 88–97, 1960.

[14] А. Письменний, «Створення віртуального репетитора для підготовки до мовних іспитів», 2024, Дата звернення: 09, Березень 2025. [Online]. Доступний у: https://ekmair.ukma.edu.ua/handle/123456789/32088

[15] A. C. Harper, *Lo-Fi Aesthetics in Popular Music Discourse*. University of Oxford, 2014.
[16] Ward, Andy and Luttrell, Briony and Goold, Lachlan, «How a global crisis, drift racing and Memphis hip-hop gave us phonk–the music of the TikTok generation», *The Conversation*, вип. 16, 2024.
[17] D. Yang *et al.*, «Diffsound: Discrete Diffusion Model for Text-to-sound Generation», 28, Квітень 2023, *arXiv*: arXiv:2207.09983. doi: 10.48550/arXiv.2207.09983.
[18] A. Borji, «Generated Faces in the Wild: Quantitative Comparison of Stable Diffusion, Midjourney and DALL-E 2», 05, Червень 2023, *arXiv*: arXiv:2210.00586. doi: 10.48550/arXiv.2210.00586.
[19] «Freesound». Дата звернення: 03, Травень 2025. [Online]. Доступний у: https://freesound.org/
[20] «Our Technologies | Steinberg». Дата звернення: 16, Березень 2025. [Online]. Доступний у: https://www.steinberg.net/technology/
[21] Y. Mekonnen і A. I. Sarwat, «Renewable energy supported microgrid in rural electrification of Sub-Saharan Africa», в *2017 IEEE PES PowerAfrica*, Accra, Ghana: IEEE, Чер 2017, с. 595–599. doi: 10.1109/PowerAfrica.2017.7991293.
[22] «Individuals using the Internet (% of population) | Data». Дата звернення: 22, Березень 2025. [Online]. Доступний у: https://data.worldbank.org/indicator/IT.NET.USER.ZS
[23] «GeForce GTX 1660 Ti Gaming Laptops | NVIDIA». Дата звернення: 22, Березень 2025. [Online]. Доступний у: https://www.nvidia.com/en-me/geforce/gaming-laptops/gtx-1660-ti/
[24] «What computer should I get for music creation?» Дата звернення: 15, Березень 2025. [Online]. Доступний у: https://support.image-line.com/action/knowledgebase/?ans=214
[25] «Best Graphics Cards - March 2025». Дата звернення: 09, Березень 2025. [Online]. Доступний у: https://benchmarks.ul.com/compare/best-gpus?amount=0&sortBy=SCORE&reverseOrder=true&types=MOBILE&minRating=0
[26] I. J. Goodfellow *et al.*, «Generative Adversarial Networks», 10, Червень 2014, *arXiv*: arXiv:1406.2661. doi: 10.48550/arXiv.1406.2661.
[27] V. Iashin і E. Rahtu, «Taming Visually Guided Sound Generation», 17, Жовтень 2021, *arXiv*: arXiv:2110.08791. doi: 10.48550/arXiv.2110.08791.
[28] «Mel». Дата звернення: 22, Березень 2025. [Online]. Доступний у: https://www.sfu.ca/sonic-studio-webdav/handbook/Mel.html
[29] H. Liu *et al.*, «AudioLDM 2: Learning Holistic Audio Generation with Self-supervised Pretraining», 11, Травень 2024, *arXiv*: arXiv:2308.05734. doi: 10.48550/arXiv.2308.05734.

[30] O. Ronneberger, P. Fischer, і T. Brox, «U-Net: Convolutional Networks for Biomedical Image Segmentation», 2015, *arXiv*. doi: 10.48550/ARXIV.1505.04597.
[31] «Made With JUCE», JUCE. Дата звернення: 23, Березень 2025. [Online]. Доступний у: https://juce.com/made-with-juce/
[32] «Spaces Launch – Hugging Face». Дата звернення: 23, Березень 2025. [Online]. Доступний у: https://huggingface.co/spaces/launch
[33] «RunPod - The Cloud Built for AI». Дата звернення: 23, Березень 2025. [Online]. Доступний у: https://www.runpod.io/
[34] «Audioldm Text To Audio Generation - a Hugging Face Space by haoheliu». Дата звернення: 30, Березень 2025. [Online]. Доступний у: https://huggingface.co/spaces/haoheliu/audioldm-text-to-audio-generation
[35] G. Team, «Getting Started With The Python Client». Дата звернення: 06, Квітень 2025. [Online]. Доступний у: https://www.gradio.app/guides/getting-started-with-the-python-client
[36] Фадєєва К.В., *Музичні комп'ютерні технології XX століття : монографія*. 2006.
[37] «2024 DAW User Survey - The Results | Production Expert». Дата звернення: 20, Квітень 2025. [Online]. Доступний у: https://www.production-expert.com/production-expert-1/2024-daw-user-survey-the-results
[38] «Pro Tools Documentation». Дата звернення: 20, Квітень 2025. [Online]. Доступний у: https://avidtech.my.salesforce-sites.com/pkb/articles/en_US/user_guide/Pro-Tools-Documentation
[39] «haoheliu/AudioLDM2: Text-to-Audio/Music Generation». Дата звернення: 20, Квітень 2025. [Online]. Доступний у: https://github.com/haoheliu/audioldm2
[40] «AudioLDM 2». Дата звернення: 20, Квітень 2025. [Online]. Доступний у: https://huggingface.co/docs/diffusers/main/en/api/pipelines/audioldm2#tips
[41] «Cubase Pro 14.0.20 Operation Manual • Viewer • Steinberg». Дата звернення: 04, Травень 2025. [Online]. Доступний у: https://www.steinberg.help/v/u/cubase_pro_14_0_operation_manual_en.pdf
[42] «SigMod | NUGEN Audio». Дата звернення: 04, Травень 2025. [Online]. Доступний у: https://nugenaudio.com/sigmod/
[43] «GeForce RTX 4090 Graphics Cards for Gaming | NVIDIA». Дата звернення: 20, Квітень 2025. [Online]. Доступний у: https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4090/
[44] «A10 Tensor Core GPU | NVIDIA». Дата звернення: 20, Квітень 2025. [Online]. Доступний у: https://www.nvidia.com/en-us/data-center/products/a10-gpu/
[45] M. M. Thomas, G. Liktor, C. Peters, S. Kim, K. Vaidyanathan, і A. G. Forbes, «Temporally Stable Real-Time Joint Neural Denoising and Supersampling», *Proc. ACM Comput. Graph. Interact. Tech.*, вип. 5, вип. 3, с. 1–22, Лип 2022, doi: 10.1145/3543870.

[46] «DLSS 4: Transforming Real-Time Graphics with AI». Дата звернення: 04, Травень 2025. [Online]. Доступний у: https://research.nvidia.com/labs/adlr/DLSS4/
[47] «sound design tutorial - Google Search». Дата звернення: 04, Травень 2025. [Online]. Доступний у: https://www.google.com/search?q=sound+design+tutorial&sca_esv=3f991894fcbd4882&rlz=1C1GCEU_enUA1160UA1160&sxsrf=AHTn8zoBl5daL1HIT46fchSFjGyeuIbGoA%3A1746339437758&ei=bQYXaJyALuDlxc8P_-qgOA&ved=0ahUKEwjc6uCulYmNAxXgcvEDHX81CAcQ4dUDCBA&uact=5&oq=sound+design+tutorial&gs_lp=Egxnd3Mtd2l6LXNlcnAiFXNvdW5kIGRlc2lnbiB0dXRvcmlhbDIFEAAYgAQyBhAAGBYYHjIGEAAYFhgeMgYQABgWGB4yBhAAGBYYHjIGEAAYFhgeMgYQABgWGB4yBhAAGBYYHjIGEAAYFhgeMgYQABgWGB5I_RNQ0AJY8RFwAXgBkAEAmAFpoAH3BaoBAzguMbgBA8gBAPgBAZgCCqACrgbCAgoQABiwAxjWBBhHwgINEAAYgAQYsAMYQxiKBcICChAAGIAEGEMYigXCAgoQABiABBgUGIcCwgILEAAYgAQYkQIYigXCAgUQLhiABMICCBAAGBYYChgemAMAiAYBkAYKkgcDOS4xoAfhO7IHAzguMbgHpgY&sclient=gws-wiz-serp
[48] «synthesizer user manual - Google Search». Дата звернення: 04, Травень 2025. [Online]. Доступний у: https://www.google.com/search?as_q=synthesizer+user+manual&as_epq=&as_oq=&as_eq=&as_nlo=&as_nhi=&lr=&cr=&as_qdr=all&as_sitesearch=&as_occt=any&as_filetype=&tbs=
[49] «How much do you use presets? [Poll] - Instruments Forum - KVR Audio». Дата звернення: 20, Квітень 2025. [Online]. Доступний у: https://www.kvraudio.com/forum/viewtopic.php?t=383650
[50] Bode i Harald, «Sound Synthesizer Creates New Musical Effects.», 1961.
[51] D. Zhang *et al.*, «MM-LLMs: Recent Advances in MultiModal Large Language Models», 28, Травень 2024, *arXiv*: arXiv:2401.13601. doi: 10.48550/arXiv.2401.13601.
[52] A. Vaswani *et al.*, «Attention Is All You Need», 02, Серпень 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.
[53] «Introducing Gemini 1.5, Google’s next-generation AI model». Дата звернення: 09, Березень 2025. [Online]. Доступний у: https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/#architecture
[54] G. Team *et al.*, «Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context», 16, Грудень 2024, *arXiv*: arXiv:2403.05530. doi: 10.48550/arXiv.2403.05530.

[55] «DeepSeek-R1 671B: Complete Hardware Requirements – AI By PY». Дата звернення: 15, Березень 2025. [Online]. Доступний у: https://aibypy.com/deepseek-r1-671b-complete-hardware-requirements/10/
[56] «Gemini 1.0 Pro vs Gemini 1.5 Flash». Дата звернення: 09, Березень 2025. [Online]. Доступний у: https://llm-stats.com/models/compare/gemini-1.0-pro-vs-gemini-1.5-flash
[57] «TyrellN6: A classic racer», u-he.com. Дата звернення: 06, Квітень 2025. [Online]. Доступний у: https://u-he.com/products/tyrelln6/
[58] P. Mantione, «14 Free Synth Plugins for Your Collection», Pro Audio Files. Дата звернення: 06, Квітень 2025. [Online]. Доступний у: https://theproaudiofiles.com/11-free-soft-synth-plugins-for-your-collection/
[59] «JUCE: AudioProcessor Class Reference». Дата звернення: 30, Березень 2025. [Online]. Доступний у: https://docs.juce.com/master/classAudioProcessor.html#a5d79591b367a7c0516e4ef4d1d6c32b2
[60] «JSON vs XML - Difference Between Data Representations - AWS». Дата звернення: 30, Березень 2025. [Online]. Доступний у: https://aws.amazon.com/compare/the-difference-between-json-xml/
[61] «Prompt design strategies | Gemini API | Google AI for Developers». Дата звернення: 27, Квітень 2025. [Online]. Доступний у: https://ai.google.dev/gemini-api/docs/prompting-strategies
[62] «Upgrade to the Google Gen AI SDKs | Gemini API | Google AI for Developers». Дата звернення: 04, Травень 2025. [Online]. Доступний у: https://ai.google.dev/gemini-api/docs/migrate
[63] «LiveBench». Дата звернення: 27, Квітень 2025. [Online]. Доступний у: https://livebench.ai/#/