

Міністерство освіти і науки України
Черкаський державний технологічний університет

Кваліфікаційна наукова
праця на правах рукопису

Бондар Віталій Володимирович

УДК 004.85

Дисертація

**Моделі та метод нейромережевої генерації об'єктів на
основі профілів швидкості узгодження потоків**

123 — Комп'ютерна інженерія

12 — Інформаційні технології

Подається на здобуття наукового ступеня
доктора філософії

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



Віталій БОНДАР

Науковий керівник: **Бабенко Віра Григорівна,**
доктор технічних наук, професор

Черкаси — 2026

АНОТАЦІЯ

Бондар В. В. Моделі та метод нейромережевої генерації об'єктів на основі профілів швидкості узгодження потоків. — Кваліфікаційна наукова праця на правах рукопису .

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 123 — Комп'ютерна інженерія. — Черкаський державний технологічний університет, Черкаси, 2026.

У дисертаційній роботі розв'язано актуальну науково-прикладну задачу підвищення якості нейромережевої генерації об'єктів за обмеженого обчислювального бюджету на основі моделей генеративного процесу та масштабування обчислень із розробленням удосконаленого методу узгодження потоків, що використовує нелінійні профілі швидкості вздовж траєкторій перетворення розподілів.

Глибокі генеративні нейромережеві моделі, насамперед дифузійні моделі та моделі узгодження потоків, стали поширеним і практично цінним інструментом синтезу зображень і сигналів. Сфера їх застосування охоплює як створення візуального контенту, так і доповнення навчальних наборів у галузях, де збір реальних даних ускладнений. Основні класи таких моделей, до яких належать варіаційні автокодувальники, генеративні змагальні мережі, нормалізуючі потоки, авторегресивні моделі, дифузійні моделі та методи узгодження потоків, мають спільну математичну природу, проте розвивалися здебільшого незалежно, із власними формалізмами й позначеннями. Ця відокремленість ускладнює системне порівняння підходів і перенесення напрацьованих рішень між ними та зумовлює потребу в узагальненій моделі процесу генерації.

Головним чинником, що стримує практичне застосування найпоширеніших нині дифузійних моделей та моделей узгодження потоків, залишається висока обчислювальна вартість генерації. Її визначає кількість кроків чисельного інтегрування, оскільки для прийнятної якості потрібні десятки або сотні послідовних звернень до нейронної мережі. Скорочення цієї кількості без втрати якості є передумовою застосування таких моделей у середовищах з обмеженими ресурсами та з вимогами до часу відгуку. У ширшому контексті розподіл обчислень під час генерації, відомий як масштабування обчислень під час інференсу, став загальним засобом керування якістю без зміни архітектури мережі, проте різноманітність відповідних методів і брак моделі, яка описувала б їх незалежно від архітектури, утруднюють їх аналіз та застосування. Отже, невирішеними залишаються три пов'язані питання: відсутність узагальненої моделі генеративного процесу, брак формальної моделі масштабування обчислень під час генерації та висока обчислювальна вартість генерації. Потреба розв'язати ці питання й визначає актуальність теми дослідження.

Об'єктом дослідження є процеси нейромережевої генерації зображень та сигналів на основі перетворення розподілів ймовірності.

Предметом дослідження є метод узгодження потоків з нелінійними профілями швидкості з використанням моделей генеративного процесу та масштабування обчислень під час генерації.

Мета роботи полягає в підвищенні якості нейромережевої генерації об'єктів за обмеженого обчислювального бюджету на основі узагальненої моделі генеративного процесу та структурно-топологічної моделі масштабування обчислень із розробленням удосконаленого методу узгодження потоків, що використовує нелінійні профілі швидкості вздовж траєкторій перетворення розподілів.

Для досягнення цієї мети проаналізовано наявні генеративні моделі, виявлено їхні спільні риси й принципові відмінності та визначено місце

методу узгодження потоків серед інших підходів, розроблено узагальнену модель генеративного процесу, яка подає генеративні моделі різних типів як функції трансформації розподілів ймовірності, розроблено структурно-топологічну модель масштабування обчислень під час генерації зі структурною складовою у вигляді політик керування додатковими обчисленнями та топологічною складовою у вигляді базових обчислювальних топологій, розроблено удосконалений метод побудови траєкторій узгодження потоків на основі нелінійних профілів швидкості, обґрунтовано схеми чисельного інтегрування для таких траєкторій і способи адаптації вже навчених моделей без перенавчання, проведено експериментальну перевірку методу на задачах генерації зображень порівняно з узгодженням потоків зі сталою швидкістю, а також оцінено його застосовність поза межами генерації зображень на прикладі синтезу електрокардіограм.

Формалізацію генеративних моделей як функцій трансформації розподілів і побудову узагальненої моделі генеративного процесу виконано методами теорії ймовірностей та математичного аналізу. Схеми дискретизації траєкторій та аналіз їхньої похибки обґрунтовано методами теорії звичайних диференціальних рівнянь і чисельного інтегрування. Модель масштабування обчислень під час генерації побудовано й базові обчислювальні топології виділено методами системного аналізу та формалізації. Профілі швидкості за заданими крайовими умовами побудовано методами теорії наближень і поліноміальної інтерполяції. Навчання нейромережевих моделей узгодження потоків та їх адаптацію виконано методами оптимізації та градієнтного навчання. Якість генерації та обчислювальні витрати запропонованого підходу оцінено методами обчислювального експерименту на наборах CIFAR-10, DrawBench, Parti та PTB-XL із застосуванням метрик FID, CLIPScore, динамічного скривлення часу та відстані Вассерштейна.

Наукова новизна одержаних результатів полягає в розробленні узагальненої моделі генеративного процесу та структурно-топологічної моделі

масштабування обчислень під час генерації, а також удосконаленого методу узгодження потоків із нелінійними профілями швидкості. Запропоновані моделі базуються на поданні генеративних моделей як функцій трансформації розподілів ймовірності та на декомпозиції масштабування обчислень за політиками керування й обчислювальною топологією, що дозволило систематично порівнювати генеративні моделі різних класів і зіставляти в єдиних термінах методи масштабування обчислень із різних галузей машинного навчання. Удосконалений метод базується на заміні сталого цільового векторного поля $x_1 - x_0$ часозалежним векторним полем $v(t) (x_1 - x_0)$, заданим профілем швидкості, і не залежить від архітектури мережі та модальності даних, що дозволило підвищити якість генерації за малої кількості кроків інтегрування та пришвидшити збіжність навчання. Подальшого розвитку набули методи переналаштування розкладу на етапі інференсу, у яких сітку інтегрування моделей узгодження потоків задає профіль швидкості, що дозволило застосовувати нелінійні профілі до вже навчених моделей без перенавчання як для генерації зображень, так і для синтезу одновимірних сигналів на прикладі електрокардіограми.

Практичне значення одержаних результатів визначається тим, що запропоновані моделі та методи дозволяють будувати алгоритми генерації, які потребують меншої кількості звернень до нейронної мережі й не вимагають зміни архітектури моделі, а також алгоритми розширення навчальних вибірок. Прикладну цінність цих алгоритмів визначають три властивості: переналаштування розкладу інтегрування застосовне до вже навчених моделей без перенавчання, тож прискорення генерації досягається з мінімальними витратами на впровадження; алгоритми не залежать від архітектури та модальності даних; скорочення кількості звернень до нейронної мережі прямо зменшує обчислювальні витрати, енергоспоживання та час відгуку, розширюючи застосовність генеративних моделей у середовищах з обмеженими ресурсами та в інтерактивних сценаріях. Ці переваги підтвер-

джено кількісно: на наборі CIFAR-10 за навчання з нуля гальмівний профіль накату знижує FID на 17.4% при чотирьох кроках інтегрування відносно лінійного базового, а за адаптації переднавченої моделі без перенавчання – до 19.8%; гальмівні профілі додатково пришвидшують збіжність навчання, досягаючи фінальної якості лінійного варіанта приблизно за чверть навчального бюджету; на масштабній латентній моделі Stable Diffusion 3.5 за малої кількості кроків симетричний профіль плавного входу-виходу підвищує CLIPScore на наборах текстових запитів Parti та DrawBench; у задачі синтезу електрокардіограм FlowECG зберігає якість генерації за 10–25 кроків, тоді як базова дифузійна модель SSSD-ECG потребує близько 200 кроків; розширення навчальної вибірки PTB-XL синтетичними сигналами підвищує макроусереднену AUC ансамблю класифікаторів на діагностичній задачі з 0.9387 до 0.9502. Результати дисертаційної роботи впроваджено в навчальний процес Черкаського державного технологічного університету та Черкаського національного університету імені Богдана Хмельницького.

За темою дисертації опубліковано десять наукових праць, зокрема дві статті, що індексуються наукометричною базою даних Scopus, чотири статті в наукових фахових виданнях України категорії „Б“, три тези доповідей на всеукраїнських і міжнародних наукових конференціях та один препринт статті, розміщений у репозиторії arXiv.

Ключові слова: машинне навчання, нейронні мережі, штучний інтелект, генеративні моделі, узгодження потоків, профіль швидкості, масштабування обчислень під час генерації, чисельне інтегрування, генерація зображень, одновимірні сигнали, синтез електрокардіограм, синтетичні дані.

ABSTRACT

Bondar V. V. Models and method for neural network-based object generation using flow-matching velocity profiles. — Qualification scientific work in the form of manuscript.

Thesis for doctor of philosophy degree in speciality 123 — Computer engineering. — Cherkasy state technological university, Cherkasy, 2026.

In the dissertation, the topical scientific and applied problem of improving the quality of neural network generation of objects under a constrained computational budget is solved, based on models of the generative process and of computation scaling, by developing an enhanced flow matching method that uses non-linear velocity profiles along the distribution-transformation trajectories.

Deep generative neural network models, primarily diffusion models and flow matching models, have become a widespread and practically valuable tool for synthesising images and signals. Their scope of application ranges from the creation of visual content to the augmentation of training sets in domains where collecting real data is difficult. The main classes of such models, which include variational autoencoders, generative adversarial networks, normalising flows, autoregressive models, diffusion models, and flow matching methods, share a common mathematical nature, yet they have developed largely independently, with their own formalisms and notation. This separation hampers the systematic comparison of the approaches and the transfer of established solutions between them, and creates a need for a generalised model of the generation process.

The principal factor that constrains the practical use of the currently most widespread diffusion and flow matching models remains the high computational cost of generation. This cost is governed by the number of numerical integration

steps, since acceptable quality requires tens or hundreds of sequential evaluations of the neural network. Reducing this number without loss of quality is a prerequisite for deploying such models in resource-constrained settings and in applications with response-time requirements. In a broader context, the allocation of computation during generation, known as inference-time scaling, has become a general means of controlling quality without changing the network architecture, yet the heterogeneity of such methods and the lack of a model that would describe them independently of the architecture hinder their analysis and application. Therefore, three related problems remain unresolved: the absence of a generalised model of the generative process, the lack of a formal model of inference-time scaling, and the high computational cost of generation. The need to solve these problems determines the relevance of the research topic.

The object of the research is the processes of neural network generation of images and signals based on the transformation of probability distributions.

The subject of the research is the flow matching method with non-linear velocity profiles, using models of the generative process and of computation scaling during object generation.

The aim of the work is to improve the quality of neural network generation of objects under a constrained computational budget, based on a generalised model of the generative process and a structural-topological model of computation scaling, by developing an enhanced flow matching method that uses non-linear velocity profiles along the distribution-transformation trajectories.

To achieve this aim, existing generative models were analysed, their common features and principal differences identified and the place of flow matching among other approaches established; a generalised model of the generative process presenting generative models of different types as probability-distribution transformation functions was developed; a structural-topological model of inference-time scaling was developed, with a structural component in the form of policies governing the additional computation and a topological component in the form of basic

computational topologies; an enhanced method for constructing flow matching trajectories based on non-linear velocity profiles was developed, numerical integration schemes for such trajectories and techniques for adapting already trained models without retraining were substantiated; the method was experimentally validated on image generation tasks against constant-velocity flow matching; and its applicability beyond image generation was assessed using electrocardiogram synthesis as a case study.

The formalisation of generative models as distribution transformation functions and the construction of the generalised model of the generative process were carried out using methods of probability theory and mathematical analysis. The trajectory discretisation schemes and the analysis of their error were grounded in methods of the theory of ordinary differential equations and numerical integration. The model of inference-time scaling was built and the basic computational topologies were identified using methods of systems analysis and formalisation. Velocity profiles for given boundary conditions were constructed using methods of approximation theory and polynomial interpolation. The training and adaptation of neural network flow matching models were performed using methods of optimisation and gradient-based learning. The quality of generation and the computational cost of the proposed approach were evaluated by computational experiment on the CIFAR-10, DrawBench, Parti, and PTB-XL datasets using the FID and CLIPScore metrics, dynamic time warping, and the Wasserstein distance.

The scientific novelty of the obtained results consists in the development of a generalised model of the generative process and a structural-topological model of inference-time scaling, as well as an enhanced flow matching method with non-linear velocity profiles. The proposed models are based on presenting generative models as probability-distribution transformation functions and on decomposing inference-time scaling by governing policies and computational topology, which made it possible to systematically compare generative models of different classes and to relate, in common terms, scaling methods from different areas of machine

learning. The enhanced method is based on replacing the constant target vector field $x_1 - x_0$ with the time-dependent vector field $v(t) (x_1 - x_0)$ defined by the velocity profile and is independent of the network architecture and the data modality, which made it possible to improve generation quality at a small number of integration steps and to accelerate training convergence. Methods for re-tuning the integration schedule at the inference stage were further developed: the integration grid of flow matching models is defined by the velocity profile, which made it possible to apply non-linear profiles to already trained models without retraining, both in image generation and in the synthesis of one-dimensional signals, with electrocardiograms as a case study.

The practical value of the obtained results is determined by the fact that the proposed models and methods make it possible to build generation algorithms that require fewer evaluations of the neural network and do not call for changes to the model architecture, as well as algorithms for augmenting training sets. The applied value of these algorithms rests on three properties: schedule re-tuning is applicable to already trained models without retraining, so generation is accelerated at minimal adoption cost; the algorithms are independent of architecture and data modality; reducing the number of network evaluations directly lowers computational cost, energy consumption, and response time, broadening the applicability of generative models in resource-constrained and interactive settings. These advantages are confirmed quantitatively: on CIFAR-10, when training from scratch, the decelerating coasting profile lowers FID by 17.4% at four integration steps relative to the linear baseline, and when adapting a pretrained model without retraining – by up to 19.8%; decelerating profiles additionally speed up training convergence, reaching the final quality of the linear variant in about a quarter of the training budget; on the large-scale latent model Stable Diffusion 3.5, at a small number of steps the symmetric ease-in-out profile increases CLIPScore on the Parti and DrawBench prompt sets; in electrocardiogram synthesis, FlowECG preserves generation quality at 10–25 steps, whereas the baseline diffusion model

SSSD-ECG requires about 200 steps; augmenting the PTB-XL training set with synthetic signals raises the macro-averaged AUC of a classifier ensemble on the diagnostic task from 0.9387 to 0.9502. The results of the dissertation have been introduced into the educational process of Cherkasy State Technological University and Bohdan Khmelnytsky National University of Cherkasy.

On the topic of the dissertation, ten scientific works have been published, including two articles indexed in the Scopus scientometric database, four articles in category B professional scientific journals of Ukraine, three conference abstracts at national and international scientific conferences, and one article preprint deposited in the arXiv repository.

Key words: machine learning, neural networks, artificial intelligence, generative models, flow matching, velocity profile, inference-time scaling, numerical integration, image generation, one-dimensional signals, electrocardiogram synthesis, synthetic data.

Список публікацій здобувача за темою дисертації

1. V. Bondar, V. Babenko, R. Trembovetskyi, Y. Korobeinyk, and V. Dzyuba, “Deep generative models as the probability transformation functions”, *Communications in Computer and Information Science*, vol. 2871, pp. 97–111, 2026, ISSN: 1865-0929. DOI: [10.1007/978-3-032-16808-5_8](https://doi.org/10.1007/978-3-032-16808-5_8) [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-032-16808-5_8 **Видання індексується у наукометричній базі Scopus.**
2. V. Bondar, S. Semenov, V. Babenko, and D. Holovniak, “Flowecg: Using flow matching to create a more efficient ecg signal generator”, *Lecture Notes in Electrical Engineering*, vol. 1570, pp. 310–319, 2026, ISSN: 1876-1100. DOI: [10.1007/978-3-032-18415-3_32](https://doi.org/10.1007/978-3-032-18415-3_32) [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-032-18415-3_32 **Видання індексується у наукометричній базі Scopus.**
3. V. Bondar and V. Babenko, “Investigation of the main aspects of the generative models applications in machine learning”, *Herald of Khmelnytskyi National University. Technical sciences*, vol. 347, no. 1, pp. 69–72, 2025, ISSN: 2307-5732. DOI: [10.31891/2307-5732-2025-347-8](https://doi.org/10.31891/2307-5732-2025-347-8) [Online]. Available: <https://heraldts.khmnu.edu.ua/index.php/heraldts/article/view/1026> **Фахове видання категорії ”Б”.**
4. В. В. Бондар, В. Г. Бабенко та Д. Є. Козлов, „Масштабування обчислень під час генерації як універсальний принцип для генеративних моделей“, *Телекомунікаційні та Інформаційні Технології*, № 4 (89), с. 229–234, 2025, ISSN: 2412-4338. DOI: [10.31673/2412-4338.2025.048926](https://doi.org/10.31673/2412-4338.2025.048926) url: <https://tit.dut.edu.ua/telecommunication/article/view/2683> **Фахове видання категорії ”Б”.**
5. V. Bondar and V. Babenko, “Comparative analysis of adversarial-diffusion hybrid architectures of generative models”, *Infocommunication and computer technologies*, vol. 1, no. 09, pp. 82–88, Jul. 2025, ISSN: 2788-5518. DOI: [10.](https://doi.org/10.31673/2412-4338.2025.048926)

36994/2788-5518-2025-01-09-10 [Online]. Available: <https://visn-icct.uu.edu.ua/index.php/icct/article/view/180> **Фахове видання категорії "Б"**.

6. V. Bondar, "Inference-time scaling as a universal principle in machine learning: Classification and cross-domain analysis", *Visnyk of Kherson National Technical University*, no. 1 (96), pp. 200–206, 2026, ISSN: 2078-4481. DOI: 10.35546/kntu2078-4481.2026.1.28 [Online]. Available: https://journals.kntu.kherson.ua/index.php/visnyk_kntu/article/view/1395 **Фахове видання категорії "Б"**.
7. V. Bondar, "Velocity scheduled flow matching", *arXiv preprint arXiv:2607.11442*, 2026. DOI: [10.48550/arXiv.2607.11442](https://doi.org/10.48550/arXiv.2607.11442)

Список публікацій, які засвідчують апробацію матеріалів дисертації

1. V. Bondar and V. Babenko, “Analysis and survey on the specifics of application of hybrid adversarial-diffusion generative models”, in *Proceedings of XIII International Scientific and Practical Conference “Science in the Modern World: Innovations and Challenges”*, (Toronto), Toronto, Canada, Sep. 2025, pp. 78–87. [Online]. Available: <https://sci-conf.com.ua/wp-content/uploads/2025/09/SCIENCE-IN-THE-MODERN-WORLD-INNOVATIONS-AND-CHALLENGES-4-6.09.25.pdf>
2. В. В. Бондар та В. Г. Бабенко, „Масштабування під час інференсу у дифузійних моделях: аналогія з міркуванням у великих мовних моделях“, в *Матеріали тез доповіді IV Міжнародної науково-практичної інтернет-конференції “Інновації та перспективні шляхи розвитку інформаційних технологій”(ІПШПІТ-2025)*, (Черкаси), Черкаси, Україна: Черкаський державний технологічний університет, листоп. 2025, с. 95—97. url: https://drive.google.com/file/d/1vfK7HzALRZHFTE8SKi6P_c-D3X4K3YPK/view
3. V. Bondar, “The main types of models in machine learning”, in *Scientific Research Methodology – 2024: Ukrainian Scientific and Practical Conference*, (Cherkasy), Cherkasy: Cherkasy State Technological University, Nov. 2024, pp. 88–90. [Online]. Available: https://er.chdtu.edu.ua/bitstream/ChSTU/5470/1/Proceedings.%20SRM_2024.pdf

ЗМІСТ

Перелік умовних позначень та скорочень	19
Вступ	23
Розділ 1. Аналіз та дослідження принципів роботи глибоких генеративних моделей	31
1.1. Відмінність між генеративними та дискримінативними моделями	31
1.2. Практичне застосування генеративних моделей	33
1.2.1. Комп'ютерний зір та мультимедіа	33
1.2.2. Медичні застосування та синтез електрокардіограм .	34
1.2.3. Розширення даних та синтез навчальних вибірок . . .	36
1.2.4. Творчі індустрії та дизайн	37
1.2.5. Виклики та обмеження	37
1.3. Види генеративних моделей	38
1.3.1. Автокодувальники	38
1.3.2. Варіаційні автокодувальники	40
1.3.3. Векторно-квантовані автокодувальники	42
1.3.4. Авторегресивні моделі	44
1.3.5. Генеративні змагальні мережі	47
1.3.6. Нормалізуючі потоки	50
1.3.7. Дифузійні моделі	53
1.3.8. Узгодження потоків	58
1.4. Гібридні підходи в генеративному моделюванні	62
1.4.1. Змагально-дифузійні гібриди	62
1.4.2. Гібриди на основі потоків	66

1.5.	Масштабування обчислень під час генерації	67
1.5.1.	Історичний контекст та обмеження масштабування під час навчання	67
1.5.2.	Масштабування під час генерації у великих мовних моделях	68
1.5.3.	Масштабування під час генерації у дифузійних моделях	71
1.5.4.	Масштабування під час генерації у моделях узгодже- ння потоків	73
1.5.5.	Масштабування у інших класах генеративних моделей	73
1.6.	Оцінювання генеративних моделей: якість та обчислювальна вартість	74
1.7.	Висновки до розділу 1	76
Розділ 2. Побудова моделей генеративного процесу та масшта-		
	бування обчислень під час генерації	79
2.1.	Узагальнена модель генеративного процесу	79
2.1.1.	Уніфікована теоретична перспектива	79
2.1.2.	Аналіз різних моделей через призму трансформації розподілів	83
2.1.3.	Спільні властивості та відмінності	86
2.2.	Структурно-топологічна модель масштабування обчислень під час генерації	89
2.2.1.	Формальне означення моделі та її властивості	90
2.2.2.	Базові обчислювальні топології масштабування	92
2.2.3.	Чотири фундаментальні інваріанти масштабування під час генерації	103
2.2.4.	Практичні наслідки та перспективи	105
2.3.	Висновки до розділу 2	107
Розділ 3. Розробка удосконаленого методу узгодження потоків		

на основі застосування профілів швидкостей	110
3.1. Побудова та формальне означення методу узгодження потоків із профілем швидкості	110
3.2. Ймовірнісний шлях та маргінальне векторне поле	113
3.3. Цільова функція навчання	116
3.4. Зв'язок із суміжними роботами	118
3.5. Поліноміальні профілі швидкості	119
3.6. Переналаштування розкладу на етапі інференсу	122
3.7. Аналіз похибки дискретизації	126
3.8. Умови застосовності та межі методу	129
3.9. Висновки до розділу 3	132
Розділ 4. Практична реалізація та валідація удосконаленого методу узгодження потоків	135
4.1. Набори даних та критерії оцінювання генеративних моделей	135
4.1.1. Набори даних	135
4.1.2. Метрики якості генерації	137
4.2. Генерація зображень	140
4.2.1. Експериментальне середовище	140
4.2.2. Навчання з профілями швидкості на CIFAR-10	141
4.2.3. Адаптація переднавченої моделі на CIFAR-10	144
4.2.4. Генерація зображень за текстовим запитом	145
4.3. Синтез електрокардіограм методом узгодження потоків	149
4.3.1. FlowECG: адаптація архітектури SSSD-ECG для узгодження потоків	150
4.3.2. Результати та аналіз ефективності	152
4.3.3. Профілі швидкості у FlowECG	154
4.3.4. Розширення навчальних даних синтетичними ЕКГ	157
4.4. Висновки до розділу 4	163

Висновки	166
Список використаних джерел	171
Додаток А. Візуальне порівняння згенерованих зображень за різних профілів швидкості	193
Додаток Б. Акти про впровадження результатів дисертації	197
Додаток В. Декларація використання систем штучного інтелекту	202

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AUC	площа під ROC-кривою (Area Under the Curve)
macro-AUC	макроусереднена площа під ROC-кривою
CIFAR-10	набір зображень CIFAR-10 (Canadian Institute For Advanced Research)
CLIP	модель зіставлення зображення й тексту у спільному просторі (Contrastive Language–Image Pre-training)
CLIPScore	метрика відповідності зображення тексту на основі моделі CLIP
DDIM	детерміноване неявне семплювання дифузійних моделей (Denoising Diffusion Implicit Models)
DrawBench	бенчмарк оцінювання генерації зображень за текстом
DTW	динамічне скривлення часу (Dynamic Time Warping)
FID	відстань Фреше між розподілами ознак мережі Inception v3 (Fréchet Inception Distance)
FlowECG	метод синтезу ЕКГ на основі узгодження потоків
GAN	генеративна змагальна мережа (Generative Adversarial Network)
ITS	масштабування обчислень під час генерації (Inference-Time Scaling)
MCTS	пошук за методом Монте-Карло по дереву (Monte Carlo Tree Search)
MMD	максимальне середнє розходження (Maximum Mean Discrepancy)
NeRF	нейронні поля випромінювання (Neural Radiance Fields)

NFE	кількість обчислень функції / кроків інтегрування (Number of Function Evaluations)
Parti	бенчмарк текстових запитів PartiPrompts для генерації зображень
PTB-XL	набір клінічних записів електрокардіограм PTB-XL
ROC	робоча характеристична крива (Receiver Operating Characteristic)
SCP	стандартизовані діагностичні коди ЕКГ (SCP-ECG)
SD3.5	модель генерації зображень за текстом Stable Diffusion 3.5
SpecSim	спектральна схожість (Spectral Similarity)
SSSD-ECG	базова дифузійна модель синтезу ЕКГ (Structured State-Space Diffusion)
TorchCFM	бібліотека та переднавчена модель умовного узгодження потоків
VAE	варіаційний автокодувальник (Variational Autoencoder)
VQ-VAE	векторно-квантований варіаційний автокодувальник
VSMF	узгодження потоків із профілем швидкості (Velocity Scheduled Flow Matching)
ЕКГ	електрокардіограма
t	час уздовж траєкторії, $t \in [0, 1]$ ($t=0$ – шум, $t=1$ – дані)
τ	репараметризований час, $\tau = \alpha(t)$
x_0, x_1	зразок базового розподілу (шум) та цільового розподілу (дані)
x_t	інтерпольований зразок у момент часу t
p_0, p_1	базовий і цільовий розподіли
$\mathcal{Z}, \mathcal{X}$	латентний простір і простір даних
F_θ	функція трансформації розподілів, $\mathcal{Z} \rightarrow \mathcal{X}$

p_θ	розподіл, породжений функцією трансформації, $p_\theta = (F_\theta)_\# p_0$
$\mathcal{G}$	узагальнена модель генеративного процесу
$\mathcal{A}$	множина допоміжних відображень, задіяних лише під час навчання
K	кількість кроків трансформації (розділ 2)
p_t	граничний розподіл (ймовірнісний шлях) у момент t
$v(t)$	профіль швидкості (скаляр), $\int_0^1 v(t) dt = 1$
$\alpha(t)$	профіль руху, $\alpha(t) = \int_0^t v(s) ds$
$\rho(\tau)$	густина розподілу навчальних даних, $\rho(\tau) = 1/v(\alpha^{-1}(\tau))$
$u_t(x)$	маргінальне цільове векторне поле швидкостей
u_t^v	цільове векторне поле під профілем швидкості v
$v_\theta(x, t)$	нейромережева апроксимація поля швидкостей
θ	параметри моделі
$\mathcal{L}$	функція втрат
$\Phi_{\alpha, N}$	оператор генерації (N -кроковий дискретний потік)
π	спряження розподілів (план транспортування)
$\Pi(p_0, p_1)$	множина спряжень розподілів p_0 і p_1
$\mathcal{N}(0, I)$	стандартний нормальний розподіл
N	кількість кроків інтегрування
C	обчислювальний бюджет (розділи 1, 2)
$C_{\min}$	обчислювальна вартість одного прямого проходу моделі
Q	метрика якості результату генерації
$\text{ITS}(f_\theta, x, C)$	оператор масштабування обчислень під час генерації
$\pi_{\text{gen}}, \pi_{\text{sel}}, \pi_{\text{alloc}}$	політики генерації, відбору та виділення обчислень (розділ 2; не плутати зі спряженням π)
D_{KL}	дивергенція Кульбака–Лейблера
$\nabla_x \log p(x)$	функція оцінки (скор)

E

математичне сподівання

ВСТУП

Актуальність теми. Глибокі генеративні моделі стали одним із базових інструментів сучасного машинного навчання: вони дають змогу синтезувати зображення, сигнали та інші типи даних, а також доповнювати навчальні набори в задачах, де збір реальних даних ускладнений. Розвиток цього напрямку охоплює кілька поколінь моделей. Варіаційні автокодувальники, запропоновані Д. Кінгмою та М. Веллінгом [1], і нормалізуючі потоки, формалізовані в роботах Д. Резенде, Ш. Мохамеда та Дж. Папамакаріоса [2, 3], поєднали ймовірнісне моделювання з нейронними мережами. Генеративні змагальні мережі, введені І. Гудфелоу зі співавторами [4], та їхні стабілізовані модифікації [5] забезпечили високу якість синтезу, проте залишалися складними в навчанні. Авторегресивні моделі [6] досягали точного оцінювання густини ціною повільної послідовної генерації. Сучасне покоління моделей зумовили дифузійні моделі Дж. Хо зі співавторами [7] та підхід на основі оцінювання градієнта логарифма густини Я. Сонга [8, 9], згодом уточнені й узагальнені в роботах П. Дхарівала, А. Нікола та Т. Карраса зі співавторами [10, 11]. Найновіший етап пов'язаний із методом узгодження потоків Ю. Ліпмана зі співавторами [12], спрямленими потоками [13] та стохастичними інтерполянтами [14]; ці підходи нині є основою найпродуктивніших систем синтезу зображень, зокрема Stable Diffusion 3 [15] і Flux [16].

Попри спільну математичну природу, перелічені класи моделей розвивалися здебільшого незалежно, із власними формалізмами та позначеннями, що ускладнює системне порівняння підходів і перенесення напрацьованих рішень між ними та зумовлює потребу в узагальненій моделі процесу генерації. Водночас головним чинником, що стримує практичне застосу-

вання найпоширеніших нині дифузійних моделей і моделей узгодження потоків, залишається висока обчислювальна вартість генерації, яку визначає кількість кроків чисельного інтегрування: для прийнятної якості потрібні десятки або сотні послідовних звернень до нейронної мережі. Скорочення цієї кількості без втрати якості є передумовою застосування таких моделей у середовищах з обмеженими ресурсами та з вимогами до часу відгуку. Ширше, розподіл обчислень під час генерації, відомий як масштабування обчислень під час інференсу, став загальним засобом керування якістю без зміни архітектури мережі, проте різноманітність відповідних методів і брак моделі, яка описувала б їх незалежно від архітектури, утруднюють їх аналіз та застосування.

Отже, невирішеними залишаються три пов'язані питання: відсутність узагальненої моделі генеративного процесу, брак формальної моделі масштабування обчислень під час генерації та висока обчислювальна вартість генерації. Потреба розв'язати ці питання й визначає актуальність теми дослідження.

У дисертаційній роботі розв'язано актуальну науково-прикладну задачу підвищення якості нейромережевої генерації об'єктів за обмеженого обчислювального бюджету на основі моделей генеративного процесу та масштабування обчислень із розробленням удосконаленого методу узгодження потоків, що використовує нелінійні профілі швидкості вздовж траєкторій перетворення розподілів.

Зв'язок роботи з науковими програмами, планами, темами. Тематика дослідження відповідає пріоритетним тематичним напрямам наукових досліджень і науково-технічних розробок, затвердженим постановою Кабінету Міністрів України від 30 квітня 2024 року №476, зокрема напрямом „Системи штучного інтелекту“ та „Глибоке навчання, великі дані (big data), нейроподібні мережі“. Розроблення нових методів нейромережевої генерації даних, що дають змогу знизити обчислювальну вартість генеративних

систем на основі глибокого навчання, безпосередньо відповідає зазначеним пріоритетним напрямам.

Об’єкт дослідження – процеси нейромережевої генерації зображень та сигналів на основі перетворення розподілів ймовірності.

Предмет дослідження – метод узгодження потоків з нелінійними профілями швидкості з використанням моделей генеративного процесу та масштабування обчислень під час генерації.

Мета і завдання дослідження. Метою дослідження є підвищення якості нейромережевої генерації об’єктів за обмеженого обчислювального бюджету на основі узагальненої моделі генеративного процесу та структурно-топологічної моделі масштабування обчислень із розробленням удосконаленого методу узгодження потоків, що використовує нелінійні профілі швидкості вздовж траєкторій перетворення розподілів.

Для досягнення мети дослідження були поставлені наступні **завдання**:

- здійснити аналіз наявних генеративних моделей, виявити їхні спільні риси та принципові відмінності й визначити місце методу узгодження потоків серед інших підходів до генеративного моделювання;
- розробити узагальнену модель генеративного процесу, яка подає генеративні моделі різних типів як функції трансформації розподілів ймовірності для систематичного порівняння підходів і перенесення рішень між архітектурами;
- розробити модель масштабування обчислень під час генерації, виділивши в ній структурну складову у вигляді політик керування додатковими обчисленнями та топологічну складову у вигляді базових обчислювальних топологій, для зіставлення методів масштабування з різних галузей машинного навчання;
- розробити удосконалений метод побудови траєкторій узгодження потоків на основі нелінійних профілів швидкості, обґрунтувати схе-

- ми чисельного інтегрування для таких траєкторій та запропонувати способи адаптації вже навчених моделей узгодження потоків для використання з новим методом без повторного навчання з нуля;
- провести експериментальну перевірку розробленого удосконаленого методу узгодження потоків з профілями швидкості на задачах генерації зображень і здійснити кількісну та якісну оцінку покращення порівняно з узгодженням потоків із константною швидкістю інтегрування;
 - оцінити застосовність узгодження потоків та розробленого удосконаленого методу з профілями швидкості поза межами генерації зображень на прикладі синтезу сигналів електрокардіограми.

Методи дослідження. Для досягнення поставленої мети застосовано такі методи: методи теорії ймовірностей та математичного аналізу – для формалізації генеративних моделей як функцій трансформації розподілів і побудови узагальненої моделі генеративного процесу; методи теорії звичайних диференціальних рівнянь та чисельного інтегрування – для обґрунтування схем дискретизації траєкторій і аналізу властивостей запропонованих профілів швидкості; методи системного аналізу та формалізації – для побудови моделі масштабування обчислень під час генерації й виділення базових обчислювальних топологій; методи теорії наближень та поліноміальної інтерполяції – для побудови профілів швидкості за заданими крайовими умовами; методи оптимізації та градієнтного навчання – для навчання нейромережових моделей узгодження потоків та адаптації наявних архітектур до запропонованих профілів; методи обчислювального експерименту – для кількісного оцінювання якості генерації та обчислювальних витрат запропонованого підходу на наборах CIFAR-10, DrawBench, Parti та PTB-XL із застосуванням метрик FID, CLIPScore, динамічного скривлення часу та відстані Вассерштейна.

Наукова новизна одержаних результатів полягає в розробленні мо-

делей генеративного процесу та масштабування обчислень під час генерації, а також удосконаленого методу узгодження потоків із нелінійними профілями швидкості. Запропоновані моделі та метод базуються на поданні генеративних моделей як функцій трансформації розподілів ймовірності, на декомпозиції масштабування обчислень за політиками керування та обчислювальною топологією і на заміні сталого цільового векторного поля часозалежним, що дозволило підвищити якість генерації за малої кількості кроків інтегрування без зміни архітектури мережі. Основні одержані результати такі.

- Уперше розроблено узагальнену модель генеративного процесу, яка, на відміну від наявних описів окремих класів у власних формалізмах, подає будь-яку генеративну модель як функцію трансформації базового розподілу в розподіл даних, доповнену цільовою функцією навчання та множиною допоміжних відображень, і задає обмеження такого подання в математичній формі, що дозволило систематично порівнювати підходи та переносити напрацьовані рішення між методами.
- Уперше розроблено структурно-топологічну модель масштабування обчислень під час генерації, яка, на відміну від наявних трактувань окремих технік, подає будь-який такий метод оператором із трійкою політик – генерації, відбору та розподілу обчислювального бюджету – і зводить його обчислювальну топологію до шести базових форм, що дозволило зіставляти в єдиних термінах методи масштабування з різних галузей машинного навчання.
- Удосконалено метод узгодження потоків у частині побудови траєкторій перетворення розподілів, який, на відміну від стандартного формулювання зі сталим цільовим векторним полем $x_1 - x_0$, використовує часозалежне поле $v(t)(x_1 - x_0)$, задане профілем швидкості, і не залежить від архітектури мережі та модальності даних, що дозволило підвищити якість генерації за малої кількості кроків інтегрування та пришвидшити

збіжність навчання.

- Набули подальшого розвитку методи переналаштування розкладу на етапі інференсу, які, на відміну від нерівномірних сіток дискретизації, побудованих для дифузійних рівнянь, задають сітку інтегрування моделей узгодження потоків профілем швидкості, що дозволило застосовувати нелінійні профілі до вже навчених моделей без перенавчання як для генерації зображень, так і для синтезу одновимірних сигналів на прикладі електрокардіограми.

Практичне значення одержаних результатів визначається тим, що запропоновані моделі та методи дозволяють будувати алгоритми генерації, які потребують меншої кількості звернень до нейронної мережі й не вимагають зміни архітектури моделі. Прикладну цінність цих алгоритмів визначають три властивості: переналаштування розкладу інтегрування може бути застосоване до вже навчених моделей без перенавчання, тож прискорення генерації досягається з мінімальними витратами на впровадження; алгоритми не залежать від архітектури та модальності даних; скорочення кількості звернень до нейронної мережі прямо зменшує обчислювальні витрати, енергоспоживання та час відгуку, розширюючи застосовність генеративних моделей у середовищах з обмеженими ресурсами та в інтерактивних сценаріях.

Ці переваги підтверджено кількісно. На наборі CIFAR-10 за навчання з нуля гальмівний профіль нахату знижує FID на 17.4% при чотирьох кроках інтегрування відносно лінійного базового, а за адаптації переднавченої моделі без перенавчання – до 19.8%; гальмівні профілі додатково пришвидшують збіжність навчання, досягаючи фінальної якості лінійного варіанта приблизно за чверть навчального бюджету. На масштабній латентній моделі Stable Diffusion 3.5 за малої кількості кроків симетричний профіль плавного входу-виходу підвищує CLIPScore на наборах текстових запитів Parti та DrawBench. У задачі синтезу електрокардіограм FlowECG зберігає якість ге-

генерації за 10–25 кроків, тоді як базова дифузійна модель SSSD-ECG потребує близько 200 кроків; розширення навчальної вибірки PTB-XL синтетичними сигналами підвищує макроусереднену AUC ансамблю класифікаторів на діагностичній задачі з 0.9387 до 0.9502.

Одержані результати можуть бути використані для зменшення обчислювальних витрат генеративних систем та для розширення навчальних наборів у задачах, де збір реальних даних ускладнений.

Результати дисертаційної роботи впроваджено в навчальний процес Черкаського державного технологічного університету та Черкаського національного університету імені Богдана Хмельницького (додаток Б).

Особистий внесок здобувача. Усі наукові результати, викладені в дисертації, здобувач отримав особисто. У працях, опублікованих у співавторстві, здобувачеві належать формулювання ідеї, аналіз попередніх робіт, виведення математичних моделей, програмна реалізація, постановка й аналіз обчислювальних експериментів та підготовка тексту. Зокрема, у працях [17–20] здобувачем виконано аналіз принципів роботи та порівняння основних типів генеративних моделей, у тому числі гібридних змагально-дифузійних архітектур; праця [19] виконана одноосібно. У праці [21] здобувачем побудовано узагальнену модель генеративного процесу, яка подає генеративні моделі як функції трансформації розподілів ймовірності. У працях [22–24] здобувачем розроблено модель масштабування обчислень під час генерації та виділено базові обчислювальні топології; праця [23] виконана одноосібно. У праці [25] здобувачеві належать ідея застосування узгодження потоків до синтезу електрокардіограм (метод FlowECG), програмна реалізація та проведення експериментів. У праці [26], виконаній одноосібно, здобувачем запропоновано удосконалений метод узгодження потоків із нелінійними профілями швидкості, розроблено спосіб чисельного інтегрування таких траєкторій на нерівномірній часовій сітці, заданій профілем, обґрунтовано переналаштування розкладу інтегрування вже на-

вчених моделей без перенавчання, виконано програмну реалізацію методу та проведено серію обчислювальних експериментів. Системи генеративного штучного інтелекту під час підготовки дисертації застосовувалися для мовного та стилістичного редагування тексту й технічної підтримки верстки під наглядом автора; декларацію їх використання наведено в додатку В.

Апробація результатів дисертації. Основні положення та результати дисертації доповідалися й обговорювалися на таких наукових конференціях: Всеукраїнській науково-практичній конференції „Scientific Research Methodology – 2024“ (Черкаси, 2024); XIII Міжнародній науково-практичній конференції „Science in the Modern World: Innovations and Challenges“ (Торонто, Канада, 2025); Міжнародній науковій конференції „Sensors, Devices and Systems – 2025“ (Черкаси, 2025); XXXI Міжнародній науковій конференції „Information and Software Technologies – 2025“ (Каунас, Литва, 2025); IV Міжнародній науково-практичній інтернет-конференції „Інновації та перспективні шляхи розвитку інформаційних технологій“ (Черкаси, 2025).

Публікації. За темою дисертації опубліковано десять наукових праць, зокрема: дві статті, що індексуються наукометричною базою даних Scopus; чотири статті в наукових фахових виданнях України категорії „Б“; три тези доповідей на всеукраїнських та міжнародних наукових конференціях; а також один препринт статті, розміщений у репозиторії arXiv.

Структура та обсяг дисертації. Дисертація складається з переліку умовних позначень та скорочень, вступу, чотирьох розділів, висновків, списку використаних джерел та трьох додатків. Загальний обсяг дисертації становить 202 сторінки, з них основного тексту – 148 сторінок. Робота містить 18 рисунків, 10 таблиць та 2 алгоритми. Список використаних джерел налічує 166 найменувань.

РОЗДІЛ 1

АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРИНЦИПІВ РОБОТИ ГЛИБОКИХ ГЕНЕРАТИВНИХ МОДЕЛЕЙ

1.1. Відмінність між генеративними та дискримінативними моделями

У контексті машинного навчання велику частину задач зводять до оцінювання ймовірності належності об'єкта до певного класу або передбачення значення цільової змінної на основі наявної інформації про вхідні дані. Залежно від того, як саме модель підходить до цієї фундаментальної проблеми, усі підходи можна класифікувати на два великі класи: дискримінативні та генеративні [27, 28].

Дискримінативні моделі фокусуються на безпосередньому оцінюванні умовної ймовірності цільової змінної y за умови спостереження вхідних даних x . Математично такі моделі апроксимують функцію $f_{\theta}(x) \approx p(y|x)$, де θ позначає параметри моделі, котрі підлягають навчанню. Цей підхід концептуально спрямований на вивчення меж між класами у просторі ознак, а не на моделювання повного розподілу даних. Типовими представниками дискримінативних моделей є логістична регресія, метод опорних векторів, а також більшість архітектур нейронних мереж, призначених для розв'язання задач класифікації та регресії [27].

Найчастіше генеративну модель визначають як модель, здатну створювати нові об'єкти, які подібні до тих, що присутні у заданому наборі даних. Іншими словами генеративні моделі, здійснюють моделювання граничного розподілу вхідних даних $p(x)$ або спільного розподілу входів та виходів $p(x, y)$. У такому формулюванні модель намагається зрозуміти,

як саме розташовані дані у відповідному просторі, вивчаючи властивості їхнього розподілу в цілому. Важливою властивістю генеративних моделей є можливість використання їх для розв'язання дискримінативних задач через застосування теореми Байєса:

$$p(y|x) = \frac{p(x, y)}{p(x)} = \frac{p(x|y) \cdot p(y)}{p(x)}. \quad (1.1)$$

Оскільки у задачах класифікації для фіксованого x граничний розподіл $p(x)$ є сталою величиною, достатньо розглядати лише пропорційність $p(y|x) \propto p(x, y)$, що дозволяє використовувати спільний розподіл для передбачення класу [28].

Ключова перевага генеративних моделей полягає у їхній здатності створювати нові зразки даних шляхом семплювання з вивченого розподілу $p(x)$. Така властивість відкриває широкі можливості для синтезу даних, доповнення навчальних вибірок та розв'язання задач безумовної генерації. Крім того, генеративні моделі зазвичай не потребують розмічених даних під час навчання, оскільки вони моделюють лише розподіл вхідних даних без явного зв'язку з цільовими мітками [29].

Генеративні моделі глибокого навчання здобули широке застосування завдяки успішній генерації високоякісних зображень, текстових послідовностей, музичних композицій, а також сенсорних даних, електронних медичних записів, траєкторій мобільності та фінансових часових рядів [17, 30]. Здатність цих моделей створювати реалістичний і змістовний вміст відкрила нові можливості у багатьох галузях, від охорони здоров'я до креативних індустрій.

Навчання обох типів моделей зазвичай здійснюється через максимізацію правдоподібності даних відносно параметрів моделі. Для генеративних моделей оптимізаційна задача набуває вигляду:

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^N \log p(x_i|\theta), \quad (1.2)$$

де N позначає кількість зразків у навчальній вибірці. Альтернативно цю задачу можна сформулювати як мінімізацію дивергенції Кульбака-Лейблера між емпіричним розподілом даних $p_{\text{data}}(x)$ та модельним розподілом $p(x|\theta)$:

$$\theta^* = \arg \min_{\theta} D_{\text{KL}}(p_{\text{data}}(x) \parallel p(x|\theta)). \quad (1.3)$$

Явне обчислення обох виразів неможливе без певних припущень про природу даних чи накладання обмежень на модель машинного навчання. Саме цими припущеннями та обмеженнями і відрізняються різноманітні генеративні моделі машинного навчання.

1.2. Практичне застосування генеративних моделей

Генеративні моделі знаходять застосування у різноманітних прикладних галузях. У комп'ютерному зорі їх використовують для синтезу та редагування зображень, підвищення роздільності, генерації відео та тривимірних сцен [31]. У медицині генеративні підходи застосовують для створення синтетичних медичних зображень, розширення навчальних вибірок та генерації анонімізованих даних пацієнтів [32]. Фармакологія використовує ці моделі для генерації молекулярних структур із заданими властивостями та прискорення пошуку потенційних лікарських сполук [33, 34]. У робототехніці генеративні моделі створюють синтетичні сценарії для навчання агентів у симульованих середовищах [35]. Фінансова сфера застосовує їх для моделювання часових рядів та оцінки ризиків [36]. Обробка природної мови спирається на авторегресивні генеративні моделі для генерації тексту, машинного перекладу та узагальнення документів [37].

1.2.1. Комп'ютерний зір та мультимедіа

У комп'ютерному зорі генеративні моделі застосовуються для широкого спектру задач. Підвищення роздільності зображень (super-resolution) є одні-

єю з поширених задач, де моделі навчаються передбачати деталі високої роздільності на основі зображень низької якості. Дифузійні моделі показали високі результати в цій задачі, використовуючи умовну генерацію для відтворення текстур та дрібних деталей [31].

Задача інпейнтингу полягає у відновленні пошкоджених або відсутніх частин зображення. Генеративні моделі здатні заповнювати прогалини у зображеннях, враховуючи контекст навколишньої області та зберігаючи узгодженість переходів. Ці методи застосовуються для реставрації фотографій, видалення небажаних об'єктів та редагування зображень [31].

Механізм умовної генерації забезпечує керування процесом генерації через описи природною мовою. Моделі сімейства Stable Diffusion та DALL-E генерують зображення на основі текстових підказок [15, 38], що розширює можливості для дизайну та створення контенту.

Генерація відео є складнішою задачею через необхідність забезпечення просторової когерентності кадрів та темпоральної узгодженості. Розширення дифузійних моделей для роботи у просторово-часовому вимірі дозволяє генерувати відеокліпи зі збереженням ідентичності об'єктів [31].

У мультимедійних застосуваннях генеративні моделі використовуються для синтезу мовлення, конверсії голосу та генерації музики. Авторегресивні моделі на основі WaveNet створюють реалістичне мовлення для систем синтезу мови та голосових асистентів [39].

1.2.2. Медичні застосування та синтез електрокардіограм

Медична сфера є однією з галузей активного застосування генеративних моделей, передусім через проблему обмеженої доступності даних [32]. Збір медичних зображень та даних пацієнтів стикається з практичними та етичними обмеженнями, що зумовлює потребу в синтетичній генерації даних.

Генеративні змагальні мережі застосовуються для синтезу медичних зображень: рентгенівських знімків, МРТ-сканів, гістопатологічних зображень та інших модальностей [32]. Синтетичні зображення використовуються для розширення навчальних наборів даних, що є важливим для рідкісних захворювань із обмеженою кількістю реальних зразків. Моделі, навчені на комбінованих наборах із реальними та синтетичними зображеннями, демонструють кращу узагальнюючу здатність порівняно з моделями, навченими виключно на обмежених реальних даних.

Перевагою генеративних моделей у медичній сфері є можливість генерації синтетичних даних пацієнтів із збереженням конфіденційності. Це дозволяє дослідникам обмінюватися даними та проводити дослідження без порушення приватності пацієнтів та регуляторних вимог, таких як GDPR або HIPAA. Варіаційні автокодувальники та генеративні змагальні мережі генерують електронні медичні записи, що зберігають статистичні властивості справжніх даних, але не містять інформації про конкретних пацієнтів.

Генеративні моделі також застосовуються для покращення якості медичних зображень: зменшення шуму, підвищення роздільності та реконструкції із неповних даних. Дифузійні моделі показали результати в задачах реконструкції МРТ-зображень, що дозволяє скоротити час сканування без втрати діагностичної якості [31].

Поряд із медичними зображеннями генеративні моделі активно застосовують і до одновимірних фізіологічних сигналів, насамперед електрокардіограм. Методи синтезу електрокардіограм пройшли кілька принципово різних етапів разом із розвитком глибокого навчання. Перші підходи ґрунтувались на математичному моделюванні серцевої електрофізіології та встановлених фізіологічних перетвореннях – зокрема, методі Довера [40]. За прийнятної точності у типових клінічних сценаріях ці методи погано відтворюють повний спектр патологічних варіацій, оскільки побудовані не

на даних, а на фіксованих аналітичних моделях.

Якісний перехід відбувся з появою генеративних змагальних мереж. Архітектура WaveGAN, початково розроблена для аудіосинтезу, виявилась придатною і для генерації ЕКГ [41]. Метод Pulse2Pulse вперше продемонстрував можливість генерувати повноцінні 10-секундні записи у 12 відведеннях зі збереженням клінічно значущих співвідношень між інтервалами [42]. Було запропоновано ще кілька спеціалізованих варіантів: SynSigGAN орієнтований на синтез біомедичних сигналів загалом [43], а поєднання GAN із трансформерами підвищило точність класифікації синтетичних записів [44]. Проте GAN лишаються схильними до нестабільності навчання та колапсу мод.

Найвищу якість генерації на сьогодні демонструють дифузійні моделі, що значною мірою усунули ці проблеми. SSSD-ECG поєднав структуровані моделі простору станів із процесом стохастичного знешумлення, встановивши новий орієнтир якості на PTB-XL [45]. DiffECG розширив застосовність дифузійного підходу на відновлення пропусків і прогнозування [46], а DSAT-ECG покращив результати завдяки інтеграції трансформерів [47]. BioDiffusion поширив ту саму ідею на синтез біомедичних сигналів різної природи [48].

Спільне обмеження дифузійних підходів – висока обчислювальна вартість генерації: зворотний процес знешумлення потребує сотень прямих проходів через нейронну мережу. Це ускладнює розгортання в клінічних умовах із обмеженими ресурсами або вимогами до реального часу.

1.2.3. Розширення даних та синтез навчальних вибірок

Окремим напрямком є розширення навчальних вибірок у галузях, де збір даних стикається з практичними чи етичними обмеженнями. Це стосується медичної візуалізації, автономних систем та інших галузей, де

отримання достатньої кількості анотованих даних є складним.

У задачах класифікації зображень розширення даних за допомогою генеративних моделей покращує продуктивність дискримінативних моделей. На відміну від традиційних методів аугментації (обертання, масштабування, зміна кольору), генеративні моделі створюють семантично різноманітні зразки, що представляють варіації поз, освітлення та контексту.

Умовна генерація дозволяє створювати збалансовані навчальні вибірки. У випадках дисбалансу класів у навчальних даних генеративні моделі синтезують додаткові зразки для недостатньо представлених класів, що покращує здатність моделей навчатися ознакам рідкісних категорій.

1.2.4. Творчі індустрії та дизайн

Сфера дизайну та створення контенту також використовує генеративні підходи. Індустрії моди застосовують генеративні змагальні мережі для створення дизайнів продуктів та систем віртуальної примірки [49].

В архітектурному проектуванні ці методи дозволяють дизайнерам досліджувати варіанти планувань та візуалізацій, генеруючи альтернативні рішення на основі заданих обмежень. Це прискорює ітераційний процес та розширює простір можливих рішень.

Ігрова індустрія використовує генеративні моделі для створення текстур, ландшафтів, персонажів та інших елементів. Процедурна генерація знижує витрати на розробку та забезпечує різноманітність ігрових світів.

1.2.5. Виклики та обмеження

Практичне застосування генеративних моделей стикається з низкою викликів. Питання якості та контрольованості генерації залишаються актуальними, особливо у задачах, що вимагають високої точності результатів.

Етичні питання, пов'язані з генерацією синтетичного контенту, потре-

бують розгляду та регулювання. Можливість створення дипфейків та оманливого контенту зумовлює потребу в методах детекції синтетичних даних та етичних стандартах використання генеративних технологій.

Обчислювальні вимоги генеративних моделей, особливо дифузійних, залишаються обмежуючим фактором для широкого впровадження. Оптимізація та прискорення генерації є напрямком активних досліджень.

1.3. Види генеративних моделей

Притаманна складність і концептуальна невизначеність генеративних задач призвели до виникнення численних та різноманітних підходів до їх розв'язання. Серед найбільш поширених методологій у машинному навчанні виділяють авторегресивні моделі, варіаційні автокодувальники, генеративні змагальні мережі, дифузійні моделі та моделі на основі потоків [17]. Кожен із цих типів моделей пропонує власні методологічні рішення для фундаментальної задачі генерації об'єктів, що апроксимують розподіл ймовірності заданого набору даних, надаючи унікальні переваги та міркування для різних сценаріїв застосування [50].

1.3.1. Автокодувальники

Автокодувальник було незалежно описано в різних роботах 80х років ХХ століття (Румельхарт, Хінтон, Вілльямс [51]; Ле Кун [52]; Бурлар, Кемп [53]), але популярності він набув після статті Хінтона та Салахутдінова в 2006 році [54].

Автокодувальник є спеціалізованою архітектурою нейронної мережі, яка призначена для відновлення вхідних даних через їх попереднє стиснення до компактного представлення. Основна ідея методу полягає у навчанні мережі здійснювати компресію інформації з подальшою декомпресією таким чином, щоб відновлені дані максимально точно відповідали початковим.

Архітектура автокодувальника включає два ключові компоненти: кодувальник E (encoder) та декодувальник D (decoder). Кодувальник виконує відображення вхідних даних у латентний простір Z меншої розмірності, в той час як декодувальник здійснює зворотне перетворення, відновлюючи початкові дані з їх стисненого представлення. Загальну схему подано на рисунку 1.1: кодувальник стискає вхідні дані X до латентного представлення Z , з якого декодувальник відновлює наближення X' початкових даних. Залежно від специфіки задачі, процес навчання такої мережі може відбуватися за двома основними сценаріями.

Перший підхід ґрунтується на прямій реконструкції вхідних даних, коли мережа навчається мінімізувати похибку між оригінальними та відновленими значеннями. Математично це можна сформулювати як задачу оптимізації параметрів θ :

$$\theta^* = \arg \min_{\theta} \|X - D(E(X))\|, \quad (1.4)$$

де X позначає вхідні дані, а норма визначає міру розбіжності між оригіналом та його реконструкцією.

Другий підхід реалізує концепцію шумоочищення (denoising), за якої мережа навчається відновлювати початкові дані з їх пошкоджених версій $\tilde{X}$. У цьому випадку цільова функція набуває вигляду:

$$\theta^* = \arg \min_{\theta} \|X - D(E(\tilde{X}))\|. \quad (1.5)$$

Для створення пошкоджених версій вхідних даних зазвичай застосовують один з двох методів: випадкове маскування окремих елементів або внесення адитивного гауссівського шуму до початкового сигналу [55].

Після завершення навчання декодувальник D здатен функціонувати як незалежна генеративна модель, яка створює об'єкти з простору X шляхом перетворення точок латентного простору Z . Проте для того, щоб декодувальник міг ефективно виконувати генеративні функції, необхідна наявність

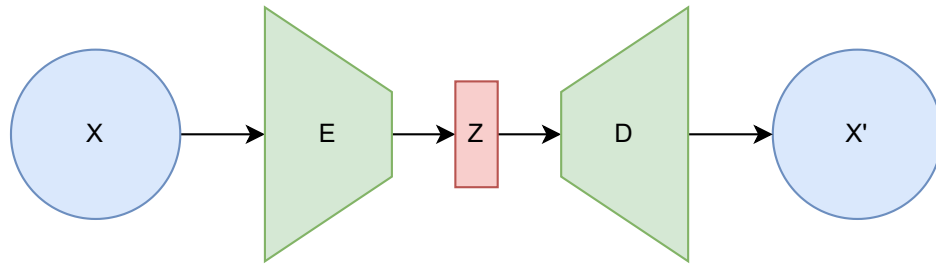


Рис. 1.1. Архітектура автокодувальника

відомого розподілу імовірності у латентному просторі Z та можливість здійснювати з нього вибірку. За виконання цих умов автокодувальник перетворюється на повноцінну генеративну модель, що дозволяє синтезувати нові об'єкти через семплювання з апіорного розподілу у латентному просторі та наступне декодування отриманих представлень.

1.3.2. Варіаційні автокодувальники

Класична архітектура автокодувальника не накладає жодних обмежень на структуру латентного простору векторів Z , внаслідок чого процес семплювання стає непрактичним для генеративних задач. Відсутність чітко визначеного розподілу у латентному просторі унеможливорює гарантію того, що випадково обрана точка $z \in Z$ буде декодована у реалістичний об'єкт з простору даних. Це фундаментальне обмеження вдалося подолати у роботі Кінгми та Веллінга “Auto-Encoding Variational Bayes” [1], де було запропоновано архітектуру варіаційного автокодувальника (VAE).

Варіаційний автокодувальник базується на обмеженні латентного розподілу до стандартного нормального $z \sim \mathcal{N}(0, I)$. У цій концепції кодувальник та декодувальник інтерпретуються як функції, що навчаються перетворенням між латентним простором та простором даних X : кодувальник $E(x)$ апроксимує апостеріорний розподіл $p(z|x)$, у той час як декодувальник $D(z)$ моделює умовний розподіл $p(x|z)$.

Автори показали, що нижня оцінка логарифмічної правдоподібності

даних виражається через такий функціонал:

$$-\mathcal{L}_{\text{VAE}} = \log p_{\theta}(x|z) - D_{\text{KL}}(p_{\phi}(z|x) \parallel p(z)) \leq \log p(x), \quad (1.6)$$

де D_{KL} позначає дивергенцію Кульбака-Лейблера, $p_{\theta}(x|z)$ відповідає функції декодувальника D , котра здійснює відображення з латентного простору Z у простір даних X , тоді як $p_{\phi}(z|x)$ представляє функцію кодувальника E , який перетворює дані з простору X у латентний простір Z .

Навчання варіаційного автокодувальника відбувається через одночасну оптимізацію двох складових цільової функції. Перша компонента забезпечує якість реконструкції вхідного об'єкта $x \in X$, підтримуючи відповідність між оригіналом та відновленими даними. Друга компонента регулює структуру латентного представлення, наближаючи передбачений розподіл $p_{\phi}(z|x)$ до апіорного стандартного нормального $p(z) = \mathcal{N}(0, I)$.

Важливою характеристикою VAE є генерація кодувальником не детермінованого латентного представлення, а саме параметрів імовірнісного розподілу. Схема реконструкції записується як $x' = D(z' \sim E(x))$, де символ $\sim$ означає операцію семплювання з передбаченого розподілу.

Однак така схема породжує технічну проблему: стохастичне семплювання є недиференційованою операцією, що унеможливлює пряме застосування алгоритму зворотного розповсюдження помилки під час навчання мережі.

Вирішення цієї проблеми досягається через накладання додаткової умови на вихід кодувальника E : він має передбачати лише параметри нормального розподілу – математичне сподівання μ та стандартне відхилення σ . Така умова дозволяє застосувати репараметризаційний трюк, коли семплювання виконується опосередковано через допоміжну випадкову величину $\varepsilon \sim \mathcal{N}(0, I)$. Модифікована схема має вигляд:

$$x' = D(E_{\sigma}(x) \cdot \varepsilon + E_{\mu}(x)), \quad (1.7)$$

де $E_\mu(x)$ та $E_\sigma(x)$ позначають виходи кодувальника, котрі відповідають за середнє значення та стандартне відхилення відповідно. За такого формулювання стохастична компонента ε не залежить від параметрів моделі, роблячи всю операцію диференційованою та придатною для градієнтної оптимізації. Репараметризація забезпечує поширення градієнтів через усю мережу, що уможливлює ефективне навчання обох компонентів архітектури. Схему варіаційного автокодувальника з репараметризаційним трюком наведено на рисунку 1.2.

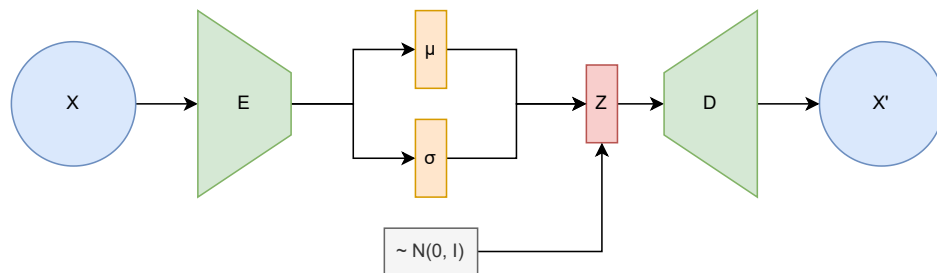


Рис. 1.2. Архітектура варіаційного автокодувальника

Варіаційні автокодувальники знайшли широке застосування у задачах генерації зображень, тексту та інших типів даних завдяки вдалому балансу між якістю генерації та обчислювальною ефективністю. Структурований латентний простір VAE виявився корисним для інтерполяції між об'єктами, маніпуляції атрибутами та навчання роз'єднаних представлень, котрі широко використовуються у різноманітних задачах машинного навчання.

1.3.3. Векторно-квантовані автокодувальники

Альтернативним способом структурування латентного простору Z автокодувальника, який водночас зберігає можливість генерації нових об'єктів через декодувальник, виступає перехід до дискретного представлення. Такий підхід запропонували Ван ден Оорд та співавтори у роботі “Neural Discrete Representation Learning”, представивши архітектуру векторно-квантованого варіаційного автокодувальника (VQ-VAE) [56].

У запропонованій моделі кожний об'єкт латентного простору $z \in Z$ формується з набору векторів, котрі зберігаються у фіксованій “книзі кодів”. Така архітектура забезпечує узгодження між розподілом передбачень кодувальника $E(x)$ та апріорним розподілом через операцію квантизації, що визначається так:

$$z(x) = \text{Quantize}(E(x)) = e_k, \quad k = \arg \min_j \|E(x) - e_j\|, \quad (1.8)$$

де e_j позначають вектори з книги кодів, а k – індекс найближчого вектора до передбачення кодувальника.

Однак реалізація даного підходу стикається з двома істотними технічними проблемами. По-перше, вибір елемента з квантованого простору є недиференційованою операцією, що ускладнює застосування градієнтних методів оптимізації. По-друге, постає питання про ефективний спосіб визначення оптимальних фіксованих векторів у квантованому просторі.

Для розв'язання цих проблем автори використовують механізм подвійного наближення. Цей механізм одночасно навчає вектори квантованого простору, зміщуючи їх у напрямку передбачень кодувальника E , і паралельно притягує передбачення кодувальника до квантованих значень. Завдяки такій схемі квантовані вектори поступово апроксимують найтипівіші неквантовані передбачення кодувальника E . Результуюча функція втрат записується у вигляді:

$$\mathcal{L} = \|x - D(e_k)\| + \|\text{sg}[E(x)] - e_k\| + \beta \cdot \|E(x) - \text{sg}[e_k]\|, \quad (1.9)$$

де оператор $\text{sg}[\cdot]$ позначає зупинку обчислення градієнта під час оптимізації, тоді як β є гіперпараметром, котрий контролює баланс між компонентами функції втрат.

Перший доданок забезпечує якість реконструкції вхідних даних через декодувальник. Другий доданок відповідає за наближення квантованих векторів до виходів кодувальника, у той час як третій притягує передбачення

кодувальника до квантованих представлень.

Проблема недиференційованості операції квантизації вирішується за допомогою припущення $\nabla E(x) \approx \nabla e_k$, що обґрунтовується наявністю оптимізаційної функції, яка забезпечує умову $E(x) \approx e_k$. Практично це означає, що під час зворотного розповсюдження помилки градієнти від декодувальника передаються безпосередньо до кодувальника, обходячи недиференційовану операцію квантизації.

Векторно-квантовані автокодувальники продемонстрували високу ефективність у семантичному представленні складних об'єктів, зокрема зображень та аудіосигналів. Дискретна природа латентного простору робить VQ-VAE природним вибором декодувальника у багатьох сучасних генеративних моделях, котрі здійснюють генерацію безпосередньо у латентному просторі [57].

1.3.4. Авторегресивні моделі

Авторегресивний підхід до генеративного моделювання базується на фундаментальному припущенні про послідовну природу даних. Коли приклад вибірки може бути представлений як послідовність елементів, виникає можливість здійснювати генерацію не цілого об'єкта одночасно, а поелементно, де кожний наступний елемент залежить виключно від попередніх. Таке тлумачення дозволяє переформулювати задачу генеративного моделювання як апроксимацію добутку умовних розподілів:

$$p(x) = \prod_{i=1}^n p(x_i | x_1, x_2, \dots, x_{i-1}) = \prod_{i=1}^n p(x_i | x_{<i}), \quad (1.10)$$

де x_i позначає i -й елемент послідовності, а $x_{<i}$ – всі попередні елементи.

Найбільшого поширення авторегресивний підхід набув у галузі моделювання природної мови. Архітектура GPT (Generative Pre-trained Transformer), запропонована Редфордом та співавторами у 2018 році [58], заклала основу

для сучасного покоління великих мовних моделей. Принцип передбачення наступного токена, запропонований в оригінальній моделі GPT [58], виявився ефективним і продовжує використовуватися в усіх сучасних архітектурах типу “decoder-only”, зокрема GPT-4 [59], Gemini [60], моделях сімейства LLaMA [61], Qwen3 [62] та інших [37].

Продемонстрована ефективність авторегресивних моделей у обробці природної мови стимулювала і відновлення адаптації даного підходу для задач комп’ютерного зору. Візуальні авторегресивні моделі можна класифікувати за трьома основними підходами залежно від механізму представлення послідовності.

Перший підхід реалізує попиксельне передбачення шляхом перетворення двовимірних зображень у одновимірні послідовності пікселів. Піонерською роботою у цьому напрямку стала архітектура PixelRNN, представлена Ван ден Оордом та співавторами у 2016 році [63]. Ця модель використовує рекурентні нейронні мережі на основі двовимірних блоків LSTM [64] для вивчення залежностей між пікселями. Обробка зображення відбувається послідовно: рядок за рядком, піксель за пікселем, при цьому для кожного пікселя передбачається розподіл імовірностей над можливими значеннями інтенсивності.

Паралельно було запропоновано архітектуру PixelCNN [6], яка замість рекурентних мереж використовує згорткові нейронні мережі з маскованими згортками. Ключовою інновацією є механізм маскування ваг згорткових фільтрів, що унеможливорює доступ моделі до “майбутніх” пікселів під час передбачення поточного. Хоча PixelRNN демонструє кращу здатність захоплювати складні залежності, PixelCNN забезпечує значно швидше навчання завдяки можливості паралелізації згорткових операцій.

Подальший розвиток архітектури призвів до появи Gated PixelCNN [65], що усуває проблему “сліпої зони” – області зображення справа зверху від поточного пікселя, яка недоступна при використанні звичайних маскова-

них згорток. Це досягається через впровадження окремих вертикального та горизонтального стеків згорток. Архітектура PixelCNN++ [66] додатково оптимізує модель через використання дискретизованої логістичної суміші для моделювання розподілу значень пікселів та низку архітектурних вдосконалень, що підвищують якість генерації та обчислювальну ефективність.

Істотним обмеженням попиксельного підходу є низька швидкість генерації об'єктів з великою кількістю елементів. Зображення розміром 256×256 пікселів у кольоровому форматі RGB утворює послідовність довжиною понад 196 тисяч елементів, що робить пряме авторегресивне моделювання на рівні пікселів обчислювально непрактичним для зображень високої роздільності. Крім того, припущення про природу залежностей між елементами може негативно впливати на здатність моделі відтворювати повний розподіл даних [6].

Другий підхід базується на токенізованому представленні зображень та адаптує техніки з обробки природної мови для візуальної генерації. Замість роботи безпосередньо з пікселями, зображення спочатку перетворюється у послідовність дискретних токенів за допомогою векторно-квантованого автокодувальника, такого як VQ-VAE [56] або VQ-GAN [67]. На першому етапі згортковий кодувальник з векторною квантизацією стискає зображення до компактного дискретного представлення – книги кодів, що містить семантично багаті візуальні фрагменти. На другому етапі авторегресивний трансформер навчається моделювати послідовність токенів. Такий двоетапний підхід суттєво скорочує довжину послідовності порівняно з попиксельним моделюванням, роблячи застосування трансформерів обчислювально здійсненним для зображень високої роздільності.

Третій підхід реалізує ієрархічну генерацію на основі масштабів. Замість послідовного передбачення токенів у порядку растрового сканування, модель генерує зображення від грубого до детального представлення, авторегресивно передбачаючи карти токенів різних роздільностей. Архітектура

VAR (Visual AutoRegressive modeling), представлена Тянем та співавторами у 2024 році [68], переформулює авторегресивне навчання на зображеннях як передбачення “наступного масштабу” замість “наступного токена” чи “наступного пікселя”.

1.3.5. Генеративні змагальні мережі

Розглянуті вище генеративні моделі об’єднує спільна риса: кожна з них тим чи іншим способом виводить оптимізаційну функцію втрат, що описує правдоподібність густини розподілу даних. Ян Гудфелоу у класичній роботі “Generative Adversarial Networks” [4] запропонував кардинально інший підхід – вивчити функцію втрат за допомогою ще однієї нейронної мережі.

Якщо спробувати навчити дискримінативну модель для оцінювання якості згенерованих зразків, а потім використати цю оцінку для навчання генеративної моделі, дуже швидко відбудеться перенавчання останньої. Саме тому в генеративних змагальних мережах обидві моделі навчаються одночасно, але з різними цільовими функціями. Основну модель, що створює елементи цільової вибірки, прийнято називати генератором G . Дискримінативну модель, котра апроксимує функцію втрат, називають дискримінатором або критиком D . Генератор перетворює вектор шуму z , вибраний із простого розподілу p_Z , на синтетичний зразок $X' = G(z)$, тоді як дискримінатор отримує на вхід реальні зразки X_{data} та згенеровані X' і надає генератору сигнал для покращення (рис. 1.3).

Щоб дискримінатор міг апроксимувати функцію втрат для генератора, його власна функція втрат спирається на передбачення певної оцінки, яка дозволяє розділити екземпляри від генератора та екземпляри з розподілу даних. Генератор же оптимізує ту саму функцію втрат, але з протилежною метою: унеможливити розрізнення синтетичних та справжніх зразків. Через таку динаміку генератор поступово наближає розподіл своїх генерацій до

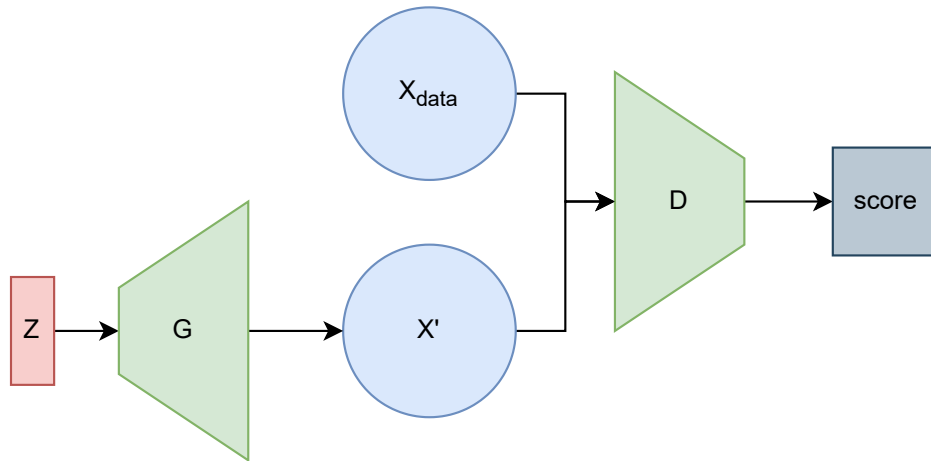


Рис. 1.3. Структурна схема генеративної змагальної мережі

розподілу реальних даних.

В оригінальній статті Гудфелоу пропонує розглядати дискримінатор як бінарний класифікатор та застосовувати бінарну перехресну ентропію як функцію втрат. Ця функція є негативною логарифмічною правдоподібністю біноміального розподілу [4]. Тоді змагальне навчання можна записати як мінімаксну гру:

$$\min_G \max_D (\mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_Z} [\log(1 - D(G(z)))]). \quad (1.11)$$

На практиці це реалізується як почергові кроки оптимізації дискримінатора та генератора. Можна також зауважити, що перший доданок не залежить від генератора, тому під час його оптимізаційного кроку цей доданок можна опустити.

Одразу в оригінальній статті Гудфелоу зазначає, що подібна мінімаксна гра може мати проблеми за високої впевненості дискримінатора у невідповідності згенерованого екземпляра. У цьому випадку похідна другого доданку буде близькою до нуля, а отже генератор не отримуватиме корисного сигналу для навчання. Тому автор пропонує так звану ненасичену змагальну функцію втрат [4]:

$$\max_D (\mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_Z} [\log(1 - D(G(z)))]), \quad (1.12)$$

$$\min_G (-\mathbb{E}_{z \sim p_Z} [\log D(G(z))]) . \quad (1.13)$$

Через змагальну природу процесу навчання під час оптимізації виникають нестабільності, що заважають знаходженню бажаного оптимуму. Ціла низка робіт присвячена пошуку інших функцій втрат для стабілізації тренування генеративних змагальних мереж [69]. Однією з дуже поширених модифікацій є змагальні мережі Вассерштайна, де зроблена спроба використати відстань Вассерштайна для порівняння розподілів як функцію апроксимації дискримінатора [5]. Автори показують, що за умови, коли дискримінатор є одноліпшицевим відображенням, мінімаксна функція втрат набуває неочікувано простої форми:

$$\min_G \max_D (\mathbb{E}_{z \sim p_Z} [D(G(z))] - \mathbb{E}_{x \sim p_{\text{data}}} [D(x)]) . \quad (1.14)$$

Обмеження Ліпшиця пропонується забезпечувати або через пряме обмеження значень параметрів дискримінатора [5], або через накладення регуляризації на градієнт дискримінатора під час його оптимізації [70].

Генеративні змагальні мережі виявились універсальним інструментом у багатьох галузях сучасних досліджень. У комп'ютерному зорі вони досягли суттєвого успіху в перетворенні зображень типу image-to-image, підвищенні роздільності та генерації фотореалістичних зображень [17, 49]. Медична сфера застосовує змагальні мережі для синтезу медичних зображень, розширення обмежених наборів даних та генерації синтетичних даних пацієнтів із збереженням конфіденційності [32]. У мультимедійних застосуваннях GAN виявились ефективними для задач конверсії голосу, генерації музики та синтезу відео. Індустрії моди та дизайну використовують генеративні змагальні мережі для створення нових дизайнів продуктів та систем віртуальної примірки. У промисловому секторі генеративні змагальні мережі знайшли застосування в синтезі даних, що дозволяє ефективно тренувати моделі машинного навчання без потреби у збиранні великих обсягів реальних даних [71, 72].

1.3.6. Нормалізуючі потоки

Нормалізуючі потоки утворюють окремий клас генеративних моделей, здатних одночасно забезпечувати ефективне семплювання та точну оцінку густини розподілу. Така подвійна властивість вигідно вирізняє їх серед інших підходів до генеративного моделювання, адже зазвичай доводиться обирати між якістю оцінки густини та швидкістю породження нових зразків. Завдяки своїй математичній елегантності та обчислювальній ефективності нормалізуючі потоки знайшли застосування у широкому спектрі прикладних задач: синтезі зображень, моделюванні шумових процесів, генерації відео та аудіосигналів, побудові графових структур, навчанні з підкріпленням, комп'ютерній графіці та симуляції фізичних систем [3].

В основі нормалізуючих потоків лежить ідея послідовного перетворення простого базового розподілу, як правило стандартного нормального, у складний цільовий розподіл даних. Таке перетворення реалізується через композицію оборотних та диференційованих функцій. Власне оборотність і відкриває принципову можливість не лише генерувати нові зразки прямим перетворенням, а й обчислювати точне значення густини ймовірності для будь-якого зразка з цільового розподілу.

Для оцінки густини зразок спочатку відображається у простір базового розподілу зворотним перетворенням, після чого враховується зміна об'єму, що її спричинює послідовність інверсних відображень [2]. Математично цей зв'язок описується класичною формулою заміни змінних:

$$p(x) = p(f^{-1}(x)) \left| \det \left(\frac{\partial f^{-1}(x)}{\partial x} \right) \right| = p(z_K) \left| \det \left(\frac{\partial z_K}{\partial x} \right) \right|, \quad (1.15)$$

де f позначає композицію всіх перетворень потоку, z_K є результатом зворотного відображення зразка x у базовий простір, а визначник Якобіана враховує локальну зміну об'єму при перетворенні.

У випадку, коли потік складається з послідовності K елементарних

перетворень $f_1, f_2, \dots, f_K$, що відображає базовий розподіл $p(z_0)$ у цільовий розподіл даних $p(x)$ (рис. 1.4), формула для логарифмічної правдоподібності набуває вигляду:

$$\log p(x) = \log p(z_0) - \sum_{i=0}^{K-1} \log \left| \det \left(\frac{\partial f_{i+1}^{-1}(z_{i+1})}{\partial z_{i+1}} \right) \right|, \quad (1.16)$$

де z_0 відповідає базовому розподілу, а кожен наступний $z_{i+1} = f_{i+1}(z_i)$ є результатом застосування чергового перетворення. Еквівалентно цю формулу можна записати через прямі перетворення:

$$\log p(x) = \log p(z_0) + \sum_{i=1}^K \log \left| \det \left(\frac{\partial f_i(z_{i-1})}{\partial z_{i-1}} \right) \right|. \quad (1.17)$$

Наявність явного виразу для правдоподібності уможливорює пряму оптимізацію параметрів моделі через максимізацію логарифмічної правдоподібності навчальних даних. Це суттєво спрощує процедуру навчання порівняно з моделями, що потребують варіаційних наближень або змагальних схем оптимізації.

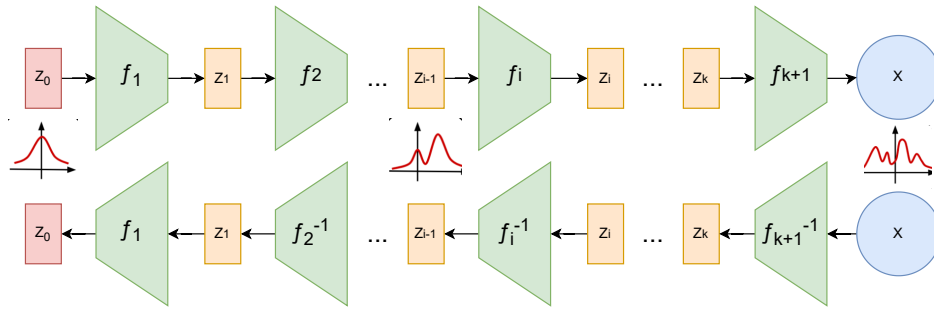


Рис. 1.4. Структурна схема нормалізуючого потоку

На практиці функції f_i реалізують за допомогою нейронних мереж достатньої виразності, щоб апроксимувати складні нелінійні перетворення. Однак при цьому накладаються дві критичні вимоги: перетворення мусить бути строго оборотним, а визначник його Якобіана має обчислюватися ефективно. Наївний підхід до обчислення визначника матриці розмірності $n \times n$ потребує $O(n^3)$ операцій, що є неприйнятним для даних великої

розмірності. Через ці обмеження коло допустимих архітектур нейронних мереж істотно звужується.

Показовим прикладом архітектури, що задовольняє обидві вимоги, слугує афінний спарений шар, уперше описаний Дінхом та співавторами [73]. Ця конструкція реалізує оборотну бієкцію завдяки поділу вхідного вектора на дві частини: перша залишається без змін і водночас параметризує афінне перетворення другої. Формально пряме перетворення визначається як:

$$y_{1:d} = x_{1:d}, \quad y_{d+1:n} = x_{d+1:n} \odot \exp(s(x_{1:d})) + t(x_{1:d}), \quad (1.18)$$

де $s(\cdot)$ та $t(\cdot)$ позначають довільні нейронні мережі, що обчислюють параметри масштабування та зсуву відповідно, а символ $\odot$ означає поелементне множення.

Зворотне перетворення має явний аналітичний вигляд:

$$x_{1:d} = y_{1:d}, \quad x_{d+1:n} = (y_{d+1:n} - t(y_{1:d})) \odot \exp(-s(y_{1:d})). \quad (1.19)$$

Характерною рисою афінного спареного шару є структура його матриці Якобіана, котра має нижньотрикутну блочну форму:

$$J = \begin{bmatrix} I_d & 0_{d \times (n-d)} \\ \frac{\partial y_{d+1:n}}{\partial x_{1:d}} & \text{diag}(\exp(s(x_{1:d}))) \end{bmatrix}. \quad (1.20)$$

Визначник такої матриці дорівнює добутку діагональних елементів, причому перші d елементів є одиницями. Отже, обчислення визначника зводиться до простого підсумовування:

$$\log |\det(J)| = \sum_{j=1}^{n-d} s_j(x_{1:d}), \quad (1.21)$$

що потребує лише $O(n)$ операцій замість $O(n^3)$ для загального випадку.

Пошук нових архітектур для нормалізуючих потоків триває й донині. Крім афінних спарених шарів, дослідники запропонували чимало альтер-

нативних конструкцій: автоматично оборотні резидуальні мережі [74], неперервні потоки на базі звичайних диференціальних рівнянь [75], а також різноманітні вдосконалення базових архітектур, що підвищують виразність моделей без втрати обчислювальної ефективності [3].

1.3.7. Дифузійні моделі

Дифузійні моделі становлять окремий клас генеративних моделей, що здобули широке визнання завдяки здатності породжувати високоякісні зразки у різноманітних прикладних галузях. Концептуальні засади цього підходу було незалежно сформульовано кількома дослідницькими групами під різними назвами та з відмінним теоретичним обґрунтуванням: дифузійні ймовірнісні моделі [76], оціночні мережі, зумовлені шумом [77], та знешумлюючі дифузійні ймовірнісні моделі [7]. Попри розбіжності у первісних формулюваннях, усі ці підходи зводяться до єдиної теоретичної конструкції, яку нині прийнято називати дифузійними моделями.

Стрімке поширення дифузійних моделей розпочалося після публікації роботи Дхарівала та Нікхоль “Diffusion Models Beat GANs on Image Synthesis” [10], де було емпірично продемонстровано перевагу дифузійного підходу над генеративними змагальними мережами у задачах синтезу зображень. Відтоді дифузійні моделі знайшли застосування у генерації зображень, відео, аудіосигналів, прогнозуванні руху та створенні анімацій, утвердившись як один із провідних інструментів сучасного генеративного моделювання.

Принцип роботи дифузійних моделей ґрунтується на поступовому перетворенні складного розподілу даних у стандартний гауссівський розподіл шляхом ітеративного додавання шуму та перемасштабування. Прямий процес послідовно переводить зразок даних x_0 через проміжні стани $x_1, x_2, \dots, x_T$ до стану повного зашумлення, а зворотний відновлює

дані з шуму багаторазовим застосуванням навченої функції знешумлення (рис. 1.5). Пряма дифузія формально описується рекурентним співвідношенням:

$$x_t = a(t) \cdot x_{t-1} + \sigma(t) \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \quad (1.22)$$

де $a(t)$ та $\sigma(t)$ є функціями розкладу шуму, що визначають швидкість перетворення вихідного розподілу в нормальний. Після достатньо великої кількості таких перетворень розподіл зашумлених даних наближається до стандартного нормального $\mathcal{N}(0, I)$.

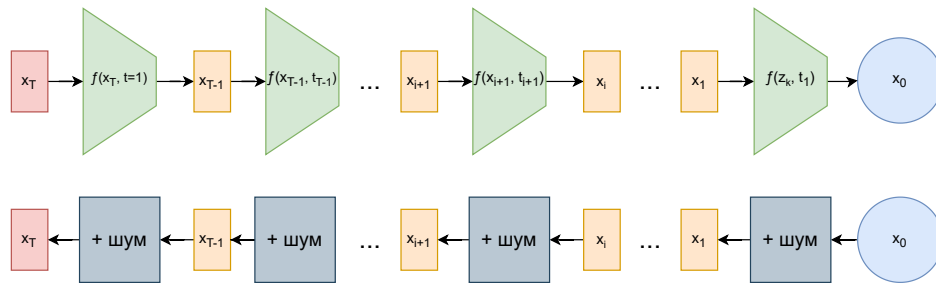


Рис. 1.5. Структурна схема дифузійної моделі

Важливою властивістю прямого процесу є можливість безпосереднього семплювання зашумленого зразка на довільному кроці t без послідовного обчислення всіх проміжних станів. Оскільки сума нормально розподілених випадкових величин залишається нормально розподіленою, граничний розподіл x_t за умови початкового зразка x_0 виражається замкненою формулою:

$$x_t = \bar{a}(t) \cdot x_0 + \bar{\sigma}(t) \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \quad (1.23)$$

де $\bar{a}(t)$ та $\bar{\sigma}(t)$ є кумулятивними коефіцієнтами, обчисленими з послідовності елементарних перетворень.

Генеративна здатність дифузійних моделей реалізується через навчання зворотного процесу, що послідовно відновлює початкові дані зі стану повного зашумлення. Зворотне перетворення здійснюється багаторазовим застосуванням єдиної параметризованої функції f_θ , яка отримує зашумлений зразок x_t та часовий параметр t , що характеризує рівень зашумленості.

За загальноприйнятою конвенцією значення $t = 0$ відповідає незашумленому зразку, тоді як $t = 1$ позначає стан повної відсутності сигналу.

Найпоширенішим підходом до навчання дифузійних моделей є мінімізація середньоквадратичної похибки між справжнім доданим шумом та його передбаченням моделлю:

$$\mathcal{L}_{\text{noise}} = \mathbb{E}_{t, x_0, \varepsilon} \|\varepsilon - f_{\theta}(x_t, t)\|^2, \quad \text{де} \quad x_t = \bar{a}(t) \cdot x_0 + \bar{\sigma}(t) \cdot \varepsilon. \quad (1.24)$$

Така параметризація, запропонована Хо та співавторами [7], виявилася ефективнішою за пряме передбачення незашумленого зразка завдяки кращій обумовленості оптимізаційної задачі.

Дифузійну модель можна інтерпретувати як функцію, що задає перетворення вектора залежно від часу. Запис у формі $\Delta x = f_{\theta}(x, t)\Delta t$, або еквівалентно $f_{\theta} = \frac{dx}{dt}$, показує, що функція перетворення є розв'язком певного диференціального рівняння. Така інтерпретація відкриває можливість застосування багатого арсеналу методів чисельного розв'язання диференціальних рівнянь [11].

Сонг та Ермонг запропонували альтернативне формулювання, де дифузійна модель визначається як функція оцінки, що апроксимує градієнт логарифму густини розподілу $\nabla_x \log p_{\text{data}}(x)$ [8]. Основними складовими їхнього підходу є наближення оцінки та ланжевенівська динаміка.

У такому формулюванні задача оптимізації полягає у мінімізації функціонала:

$$\mathbb{E}_{p_{\text{data}}} \|f_{\theta}(x) - \nabla_x \log p_{\text{data}}(x)\|^2. \quad (1.25)$$

Безпосереднє обчислення цього виразу є неможливим через невідомість істинного розподілу даних. Проте можна показати, що мінімізація еквівалентна оптимізації виразу:

$$\mathbb{E}_{p_{\text{data}}} \left[\text{tr}(\nabla_x f_{\theta}(x)) + \frac{1}{2} \|f_{\theta}(x)\|^2 \right], \quad (1.26)$$

який, однак, залишається непрактичним через обчислювальну складність знаходження сліду якобіана для глибоких нейронних мереж.

Вирішення цієї проблеми досягається через введення густини розподілу шуму $q_\sigma(\tilde{x}|x)$ та застосування техніки наближення оцінки. Сонг та Ермонг показують, що функцію втрат можна апроксимувати виразом:

$$\mathcal{L}_{\text{score}} = \mathbb{E}_{p_{\text{data}}, q_\sigma} \|f_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)\|^2, \quad (1.27)$$

який є зручним для практичного застосування, оскільки градієнт логарифму умовного розподілу шуму має явний аналітичний вигляд [8].

Динаміка Ланжевена забезпечує механізм семплювання з розподілу ймовірності $p(x)$, використовуючи лише функцію оцінки $\nabla_x \log p(x)$ та рекурентне правило:

$$\hat{x}_t = \hat{x}_{t-1} + \frac{\epsilon}{2} \nabla_x \log p(\hat{x}_{t-1}) + \sqrt{\epsilon} \cdot z_t, \quad z_t \sim \mathcal{N}(0, I), \quad (1.28)$$

де ϵ є параметром кроку. За умов $\epsilon \rightarrow 0$ та $T \rightarrow \infty$ розподіл $\hat{x}_T$ збігається до цільового $p(x)$. Підставляючи навчену нейронну мережу як функцію оцінки, отримуємо генеративну модель, здатну семплювати з розподілу даних [8].

За своєю суттю динаміка Ланжевена є чисельним методом розв'язку стохастичного диференціального рівняння. Каррас зі співавторами продемонстрували, що процес семплювання може бути обраний незалежно від процедури оптимізації функції оцінки [11]. Це дозволяє застосовувати різноманітні чисельні методи, засновані як на стохастичних диференціальних рівняннях (SDE), так і на звичайних диференціальних рівняннях (ODE) за припущення нульового вінерівського шуму в ланжевенівській динаміці. Дослідження показують, що підхід на основі SDE забезпечує кращу якість генерації, хоча й потребує ретельнішого налаштування гіперпараметрів [11].

Хоча дифузійні процеси традиційно формуються з використанням диференціальних рівнянь [7], нещодавні розробки показали, що той самий процес може бути описаний з використанням простіших математичних основ [78].

Головним практичним обмеженням дифузійних моделей є необхідність багаторазового застосування функції f_θ , котра реалізується глибокою нейронною мережею, для генерації одного зразка. Типова кількість кроків семплювання становить від кількох десятків до кількох тисяч, що суттєво уповільнює процес генерації порівняно з моделями, котрі породжують зразки за один прохід мережі. Це стимулювало активні дослідження у напрямку прискорення семплювання через вдосконалення чисельних методів [11, 79, 80], дистиляцію моделей [81–83] та альтернативні формулювання генеративного процесу.

Окремим напрямком досліджень є оптимізація розкладу шуму – послідовності коефіцієнтів $a(t)$ та $\sigma(t)$, що визначають інтенсивність зашумлення на кожному кроці. Чен показав, що оптимальний розклад залежить від характеристик даних: для зображень більшої роздільності він зміщується в бік інтенсивнішого зашумлення через підвищену надлишковість пікселів [84]. Сабур та співавтори запропонували метод Align Your Steps, що оптимізує розклад семплювання через мінімізацію верхньої межі дивергенції Кульбака-Лейблера між істинним та лінеаризованим розподілами [85]. Альтернативний підхід полягає у навчанні адаптивного розкладу безпосередньо з даних: архітектура MuLAN застосовує багатовимірний гауссівський шум із різною інтенсивністю для різних ділянок зображення залежно від їхнього вмісту [86].

Дифузійні моделі здобули значну популярність після публікації згаданої вище роботи [10]. З того часу ці моделі знайшли застосування у численних дослідницьких галузях. У комп'ютерному зорі дослідники успішно застосовували їх для розв'язання фундаментальних задач підвищення

роздільності, інпейнтингу та відновлення зображень. Сфера застосування дифузійних моделей розширилась на синтез відео та завершення хмар точок. Галузь мультимодального навчання також активно застосовує дифузійні моделі, про що свідчать нещодавні роботи з генерації зображень за текстом, синтезу відео за текстом та створення тривимірних об'єктів за текстом [31]. В області міждисциплінарних досліджень ці моделі було адаптовано для моделювання молекулярних графів та продемонстровано практичну корисність у дизайні матеріалів і реконструкції медичних зображень. Їх застосування поширюється на аналіз темпоральних даних, де дослідники використовували їх для доповнення часових рядів, прогнозування та обробки хвильових сигналів [17, 31].

1.3.8. Узгодження потоків

Пошук альтернативних формулювань генеративного процесу, спрямований на подолання обмежень дифузійних моделей, привів до розробки методу узгодження потоків. Цей підхід належить до класу неперервних нормалізуючих потоків, проте вирізняється принциповою особливістю: він не потребує симуляції траєкторій під час навчання [12]. Така властивість усуває обчислювальні витрати, пов'язані з чисельним інтегруванням диференціальних рівнянь або оцінюванням функції скору на етапі оптимізації параметрів.

Центральною концепцією методу є навчання залежного від часу векторного поля швидкостей $u_t(x)$, яке забезпечує транспортування зразків від базового розподілу p_0 (зазвичай стандартного нормального) до цільового розподілу даних p_1 . На відміну від дифузійних моделей, де траєкторія визначається стохастичним процесом, узгодження потоків спирається на детерміністичний інтерполяційний шлях. Найпростішим варіантом є лінійна

інтерполяція:

$$x_t = t \cdot x_1 + (1 - t) \cdot x_0, \quad (1.29)$$

де x_1 позначає елемент цільового розподілу даних, а x_0 – відповідний елемент вихідного розподілу, як правило стандартного нормального. Параметр $t \in [0, 1]$ визначає позицію на інтерполяційному шляху: за $t = 0$ маємо чистий шум, а за $t = 1$ – зразок з розподілу даних. Зауважимо, що ця часова конвенція протилежна до прийнятої вище для дифузійних моделей, де $t = 0$ відповідає незашумленим даним, а $t = 1$ – повному шуму, тобто там x_0 позначає дані. Надалі в роботі дотримуємося саме конвенції узгодження потоків: x_0 – шум, x_1 – дані, а час іде від шуму до даних.

Ключовою перевагою узгодження потоків є гнучкість у виборі вихідного розподілу. На відміну від дифузійних моделей, де вихідний розподіл фіксується як стандартний нормальний, узгодження потоків дозволяє використовувати довільні розподіли, що відкриває можливості для задач перетворення зображень, перенесення стилю та інших застосувань, де природним є перехід між двома нетривіальними розподілами.

Методологія навчання моделей узгодження потоків спирається на відоме векторне поле, яке згодом апроксимується нейронною мережею. Для лінійної інтерполяції цільове векторне поле має простий вигляд:

$$u_t(x_t | x_0, x_1) = x_1 - x_0, \quad (1.30)$$

тобто є константним уздовж траєкторії та вказує напрямок від шуму до даних.

Ліпман та співавтори довели фундаментальний результат: градієнти за параметрами моделі для задачі регресії на маргінальне поле u_t та для задачі регресії на умовне поле, задане парою зразків, збігаються, тож обидві задачі мають спільний мінімум [12]. Ця властивість уможливорює формулювання практичної функції втрат без явного обчислення маргінального векторного

поля. Нейронна мережа $v_\theta(x, t)$ параметризує векторне поле швидкостей та навчається через мінімізацію середньоквадратичної похибки:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}(0,1), x_t \sim p_t} \|v_\theta(x_t, t) - u_t(x_t)\|^2, \quad (1.31)$$

де p_t позначає граничний розподіл інтерпольованих зразків у момент часу t . На практиці цільове поле u_t обчислюється через умовне векторне поле для конкретної пари зразків (x_0, x_1) :

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, x_0 \sim p_0, x_1 \sim p_1} \|v_\theta(x_t, t) - (x_1 - x_0)\|^2, \quad (1.32)$$

де $x_t = t \cdot x_1 + (1 - t) \cdot x_0$ є інтерпольованим зразком. Такий підхід, відомий як умовне узгодження потоків, дозволяє уникнути явного обчислення граничного векторного поля.

Генерація нових зразків у моделях узгодження потоків відбувається через розв'язання звичайного диференціального рівняння:

$$\frac{dx}{dt} = v_\theta(x, t), \quad (1.33)$$

з початковою умовою $x(0) \sim p_0$. Детерміністичне інтегрування від $t = 0$ до $t = 1$ породжує зразок $x(1)$, розподілений згідно з цільовим розподілом p_1 . Завдяки детерміністичній природі цього процесу встановлюється однозначне відображення між шумом та даними, що дозволяє застосовувати ефективні чисельні методи: метод Ейлера, схеми Рунге-Кутти вищих порядків або адаптивні алгоритми з контролем похибки.

Узгодження потоків має низку переваг порівняно з дифузійними моделями. Лінійність інтерполяційних траєкторій полегшує апроксимацію векторного поля, що позитивно впливає на якість генерації за обмеженої кількості кроків семплювання. Детерміністичний характер генеративного процесу, своєю чергою, виключає варіативність, притаманну стохастичному шуму в ланжевенівській динаміці.

Разом з тим експериментальні дослідження засвідчили, що тривалість навчання моделей узгодження потоків може перевищувати відповідний показник для класичних дифузійних моделей [12]. Це спостереження спонукало до пошуку способів пришвидшення збіжності, серед яких виокремилися методи оптимального транспорту.

Суть проблеми полягає ось у чому: випадкове спарювання елементів базового та цільового розподілів породжує траєкторії, що можуть перетинатися між собою, а це вносить неоднозначність у векторне поле. Оптимальний транспорт пропонує спосіб знаходження такого спарювання, яке мінімізує загальну вартість переміщення маси та породжує траєкторії без перетинів [87].

Формально задача оптимального транспорту між розподілами p_0 та p_1 полягає у знаходженні плану транспортування $\pi(x_0, x_1)$, що мінімізує очікувану вартість:

$$\pi^* = \arg \min_{\pi \in \Pi(p_0, p_1)} \mathbb{E}_{(x_0, x_1) \sim \pi} [c(x_0, x_1)], \quad (1.34)$$

де $c(x_0, x_1)$ є функцією вартості, як правило квадратом евклідової відстані, а $\Pi(p_0, p_1)$ позначає множину всіх спільних розподілів з граничними p_0 та p_1 .

Застосування оптимального транспортного плану для спряження елементів під час навчання моделей узгодження потоків призводить до прямолінійніших траєкторій та швидшої збіжності [87, 88]. Лю та співавтори запропонували підхід під назвою Rectified Flow, де модель ітеративно перенавчається на власних згенерованих траєкторіях, що поступово вирівнює траєкторії руху та підвищує ефективність семплювання [13].

Подальшим розвитком стали методи міні-батчевого оптимального транспорту, де оптимальне спряження обчислюється не для повних розподілів, а для окремих батчів під час навчання [87]. Такий підхід є обчислювально здійсненним та забезпечує суттєве покращення порівняно з

випадковим спряженням, хоча й не досягає глобального оптимуму.

Моделі узгодження потоків продемонстрували свою перспективність у широкому спектрі прикладних задач: генерації зображень високої роздільності [15], синтезі відеопослідовностей [89], перетворенні між доменами та умовній генерації [16]. Теоретична елегантність цього підходу разом із практичною ефективністю зумовлює зростаючий інтерес дослідницької спільноти до подальшого вдосконалення методів узгодження потоків та розширення сфери їх застосування.

1.4. Гібридні підходи в генеративному моделюванні

Кожен із розглянутих генеративних підходів має характерні переваги та обмеження. Дифузійні моделі забезпечують стабільне навчання та повне покриття розподілу даних [7, 79], проте потребують десятків або сотень кроків семплювання. Генеративні змагальні мережі здатні створювати зображення за один прохід мережі [4], однак їхнє навчання часто є нестабільним через колапс мод [5]. Нормалізуючі потоки дозволяють обчислювати правдоподібність у явному вигляді [2], але накладають обмеження на архітектуру мережі через вимогу оборотності перетворень [3].

Ці відмінності зумовили розвиток гібридних моделей, які поєднують різні генеративні підходи в одній архітектурі. Мета таких моделей полягає у використанні переваг кожного підходу для компенсації обмежень інших [18].

1.4.1. Змагально-дифузійні гібриди

Серед гібридних архітектур значну увагу привертає поєднання дифузійних моделей із генеративними змагальними мережами. Така комбінація дозволяє використовувати стабільність навчання та повноту покриття розподілу, властиві дифузійним моделям, разом зі здатністю змагальних

мереж генерувати деталізовані зображення за меншу кількість кроків [18].

Інтеграція цих підходів реалізується кількома способами. Один із поширених методів полягає у застосуванні змагальної функції втрат для дистиляції багатокрокової дифузійної моделі в генератор, здатний працювати за один або кілька кроків.

Зауер та співавтори запропонували архітектуру Adversarial Diffusion Distillation (ADD), що реалізує цей підхід [83]. У цій схемі попередньо навчена дифузійна модель слугує джерелом знань для компактнішої моделі. Водночас модель-учень оптимізується за змагальною функцією втрат, де дискримінатор розрізняє реальні та згенеровані зображення. Поєднання дистиляційної та змагальної складових функції втрат забезпечує як відтворення структури розподілу, так і деталізацію зображень. За результатами експериментів, така архітектура досягає якості генерації, порівнянної з багатокроковою дифузією, за 1–4 кроки знешумлення [90].

Лін та співавтори розвинули цей напрямок у роботі SDXL-Lightning [91], застосувавши прогресивну дистиляцію замість одноетапного скорочення кількості кроків. Модель послідовно навчається працювати спочатку з 16, потім з 8, далі з 4 кроками. На кожному етапі змагальна функція втрат контролює збереження якості зображень. Такий підхід дозволяє генерувати зображення роздільності 1024×1024 пікселів за один крок без суттєвої втрати деталізації. Інший варіант дистиляції запропоновано у роботі Diffusion2GAN [92], де генеративна змагальна мережа навчається відтворювати відображення від шуму до зображення за один прохід.

Інший спосіб інтеграції змагального навчання представлено в архітектурі Structure-Guided Adversarial Training (SADM) [93]. На відміну від попередніх підходів, тут дискримінатор оцінює не окремі зображення, а структурну узгодженість групи зразків та відповідність їхніх латентних представлень латентним представленням реальних даних. Змагальна функція втрат застосовується безпосередньо до векторного поля швидкостей,

що визначає напрямок руху під час знешумлення. Це забезпечує додатковий сигнал для корекції траєкторій генерації та покращує як реалістичність зразків, так і різноманітність генерацій.

В архітектурі Constant Acceleration Flow (CAF) [94] дискримінатор оцінює не згенеровані зображення, а траєкторії знешумлення. Процес дифузії моделюється як рух із заданим прискоренням, і дискримінатор навчається розрізняти траєкторії для реальних та синтетичних даних. При відхиленні передбачень моделі від траєкторій реальних даних змагальний сигнал коригує параметри генератора. Така схема зменшує ймовірність перетину траєкторій та артефактів, що виникають при грубій апроксимації дифузійного процесу.

Розглянуті вище підходи використовують змагальні механізми для вдосконалення дифузійних моделей. Можливий і зворотний напрямок інтеграції: застосування дифузійних процесів для стабілізації навчання генеративних змагальних мереж. Архітектура Diffusion-GAN [95] реалізує цю ідею через додавання контрольованої кількості гауссівського шуму до зображень перед подачею на дискримінатор. Дискримінатор навчається розрізняти зашумлені реальні та синтетичні зображення при різних рівнях шуму. Така схема забезпечує більш плавний градієнтний сигнал для генератора, оскільки дискримінатор оцінює зображення на різних етапах зашумлення, а не лише фінальний результат. Це стабілізує процес навчання та покращує якість генерації.

Архітектура Latent Denoising Diffusion GAN (LDDGAN) [96] виконує дифузійний процес у латентному просторі, створеному автокодувальником, замість простору пікселів. Менша розмірність латентного простору прискорює генерацію. Після декодування латентного коду в зображення змагальний дискримінатор оцінює його реалістичність. У цій схемі дифузійна складова забезпечує різноманітність генерацій, а змагальна — деталізацію текстур та чіткість зображень.

У роботі LaDiffGAN [97] дифузійний процес застосовується лише під час навчання, а не при генерації. Генератор отримує латентний код із додаванням гауссівського шуму різної інтенсивності та навчається відновлювати чистий код і створювати зображення одночасно. Функція втрат складається з дифузійної компоненти для відновлення латентного коду та змагальної компоненти для оцінки реалістичності зображення. Таке поєднання підвищує стійкість генератора до варіацій шуму та зменшує ризик колапсу мод.

Змагальні механізми застосовуються також для оптимізації параметрів дифузійного процесу. Ханг та співавтори показали, що дискримінатор може слугувати критерієм для автоматичного підбору розкладу шуму під час навчання, що покращує якість генерації без зміни архітектури основної моделі [98].

Експериментальні результати свідчать, що гібридні змагально-дифузійні архітектури перевершують як дифузійні моделі, так і генеративні змагальні мережі за метриками якості та швидкості генерації [18]. Водночас кожен тип гібридної архітектури має свою сферу застосування. Методи дистиляції забезпечують швидку генерацію після тривалого навчання, що робить їх придатними для застосувань із жорсткими вимогами до часу відгуку. Архітектури з вбудованими змагальними компонентами досягають вищої якості зображень, проте потребують більше обчислювальних ресурсів під час генерації. Дифузійно-підтримувані генеративні змагальні мережі зберігають здатність генерувати зображення за один прохід, водночас підвищуючи стабільність навчання. Вибір архітектури залежить від вимог конкретної задачі.

1.4.2. Гібриди на основі потоків

Крім змагально-дифузійних комбінацій, гібридні архітектури будуються на основі моделей узгодження потоків. Поширеним варіантом є поєднання потоків із варіаційними автокодувальниками [1]: VAE створює структурований латентний простір із відомим розподілом, а модель потоків виконує генерацію в цьому просторі. Така схема використовується в моделях сімейства Flux [16] для генерації зображень високої роздільності. Робота з компактними латентними векторами замість простору пікселів суттєво зменшує обчислювальні витрати.

Інший напрямок гібридизації полягає у поєднанні потоків з авторегресивними моделями [89]. Авторегресивні моделі добре моделюють послідовні залежності, що підтверджується їхньою ефективністю в генерації тексту [37, 58]. Водночас пряме застосування авторегресії до зображень породжує послідовності зовеликої довжини. Гібридний підхід розділяє задачу на два етапи: авторегресивна модель генерує послідовність дискретних токенів, що кодують високорівневу структуру зображення, а модель потоків перетворює кожен токен у детальний фрагмент із текстурами та плавними переходами.

Окрім парних комбінацій, існують багатокomпонентні архітектури, що інтегрують три або більше генеративних підходів. Типова схема включає такі етапи: варіаційний автокодувальник стискає зображення в латентний простір [1], векторна квантизація дискретизує цей простір [56], авторегресивний трансформер моделює послідовність дискретних токенів, а модель потоків або дифузії виконує фінальну деталізацію. Архітектура Stable Diffusion 3 [15] є прикладом такої інтеграції: латентна дифузія поєднується з трансформерними блоками уваги та механізмами умовної генерації за текстовими підказками. Кожен компонент виконує окрему функцію, а їхнє поєднання забезпечує результати, недсяжні для окремих підходів.

1.5. Масштабування обчислень під час генерації

Обчислювальна вартість генерації, названа вище серед основних обмежень практичного впровадження, тривалий час розглядалася виключно як наслідок розміру моделі та обраної архітектури. Проте останнє десятиліття досліджень засвідчило, що обсяг обчислень, витрачених безпосередньо під час породження зразка, є самостійним ресурсом, розпоряджатися яким можна незалежно від навчання [99–101].

1.5.1. Історичний контекст та обмеження масштабування під час навчання

Протягом останніх десятиліть домінуючим підходом до покращення продуктивності нейронних мереж було збільшення обчислювальних ресурсів на етапі навчання. Каплан та співавтори встановили емпіричні степеневі залежності між якістю мовних моделей та основними ресурсними параметрами [102]. Зокрема, втрати перехресної ентропії $\mathcal{L}$ описуються степеневими функціями:

$$\mathcal{L}(N) = \left(\frac{N_c}{N}\right)^{\alpha_N}, \quad \mathcal{L}(D) = \left(\frac{D_c}{D}\right)^{\alpha_D}, \quad \mathcal{L}(C) = \left(\frac{C_c}{C}\right)^{\alpha_C}, \quad (1.35)$$

де N позначає кількість параметрів моделі, D – обсяг навчальних даних, C – загальний обсяг обчислень під час тренування, а N_c , D_c , C_c та α_N , α_D , α_C – емпіричні константи. Ці закономірності виявилися стійкими на семи порядках величини, що забезпечило теоретичне підґрунтя для створення моделей з мільярдами та сотнями мільярдів параметрів.

Подальші дослідження уточнили оптимальний розподіл ресурсів між розміром моделі та обсягом навчальних даних. Гофман та колеги на основі аналізу понад 400 моделей різного масштабу дійшли висновку, що більшість великих мовних моделей є недонавченими з погляду співвідношення параметрів і токенів [103]. Для досягнення обчислювального оптимуму роз-

мір моделі та кількість навчальних токенів мають зростати пропорційно, тобто подвоєння кількості параметрів потребує подвоєння обсягу даних. Справедливість цього твердження підтвердила модель Chinchilla із 70 мільярдами параметрів, навчена на 1,4 трильйона токенів: вона перевершила модель Gopher з 280 мільярдами параметрів на широкому спектрі задач за однакових витрат на тренування.

Проте ця парадигма нашоюхнулася на фундаментальні обмеження. Вичерпання доступних якісних текстових корпусів, експоненціальне зростання енергетичних витрат на тренування наднових моделей та практичні складнощі розгортання систем гігантського розміру створюють передумови для пошуку альтернативних стратегій оптимізації. У цьому контексті особливої актуальності набуває концепція масштабування обчислень на етапі використання моделі, коли додаткові ресурси виділяються безпосередньо під час генерації відповіді. Останні дослідження охоплюють різноманіття таких підходів – від методів декодування [104] до загальних стратегій масштабування [105, 106] та їх зв'язку з процесами верифікації [107].

1.5.2. Масштабування під час генерації у великих мовних моделях

Механізм масштабування під час генерації у великих мовних моделях реалізується переважно через техніку ланцюжка міркувань (chain-of-thought). Вей та співавтори виявили, що генерація проміжних кроків обґрунтування перед фінальною відповіддю суттєво підвищує здатність моделей розв'язувати задачі, що потребують багатокрокового міркування [108]. Примітно, що цей ефект проявляється як емерджентна властивість: моделі із кількістю параметрів понад сто мільярдів набувають цієї здатності без додаткового навчання чи модифікації архітектури.

Формально техніку ланцюжка міркувань можна описати через маргіналізацію по простору проміжних міркувань. Якщо стандартна авторегресійна

генерація моделює розподіл $p(y|x)$ безпосередньо, то з використанням ланцюжка міркувань модель обчислює:

$$p(y|x) = \sum_r p(y|r, x) \cdot p(r|x), \quad (1.36)$$

де r представляє послідовність проміжних кроків обґрунтування довжиною k токенів, а x та y – вхідний запит і фінальну відповідь відповідно. Збільшення k відповідає залученню більших обчислювальних ресурсів під час генерації, оскільки кожен додатковий токен потребує повного прямого проходу через модель. Експерименти засвідчили, що модель PaLM з 540 мільярдами параметрів, з лише вісьмома прикладами ланцюжка міркувань у навчальному датасеті, досягає найкращих результатів на бенчмарку математичних задач GSM8K [108].

Паралельне семплювання є альтернативним шляхом масштабування під час генерації. Ванг та співавтори запропонували метод самоузгодженості [109]: модель генерує кілька незалежних ланцюжків міркувань для того самого запиту. Фінальну відповідь обирають за принципом більшості – той варіант, що зустрічається найчастіше серед кандидатів. Ланцюжок міркувань інвестує обчислення в одну глибоку траєкторію, самоузгодженість – у багато незалежних. Це дозволяє підвищити точність на задачах арифметичного та символічного міркування без додаткового навчання моделі.

Деревовидний пошук у просторі міркувань розширює можливості масштабування. Йао та співавтори запропонували дерево міркувань [110] – метод, що організовує генерацію як дослідження дерева можливих продовжень. У кожній вершині модель породжує кілька варіантів наступного кроку міркування, оцінює їхню перспективність і розвиває найбільш обіцяючі гілки. Це дозволяє систематично досліджувати простір розв’язків замість лінійної генерації одного ланцюжка думок. Для складних задач планування та комбінаторного міркування такий підхід виявляється ефективнішим за просте паралельне семплювання.

Снелл та колеги провели систематичне дослідження оптимального розподілу обчислень під час генерації, що виявило значний потенціал цього підходу [99]. Автори проаналізували два основні механізми масштабування: по-перше, пошук серед множини згенерованих відповідей із використанням верифікаторів; по-друге, адаптивне уточнення розподілу моделі для конкретного запиту. Аналіз засвідчив, що ефективність різних стратегій критично залежить від складності запиту, що мотивує застосування обчислювально оптимальної стратегії з адаптивним розподілом ресурсів.

Практичні результати цього дослідження є показовими: обчислювально оптимальна стратегія підвищує ефективність масштабування під час генерації вчетверо порівняно з підходом „найкраще з N “, коли просто обирається найкраща з N незалежно згенерованих відповідей. За умов фіксованого обчислювального бюджету невелика модель з додатковими обчисленнями під час генерації здатна перевершити модель у чотирнадцять разів більшу за розміром [99]. Ця закономірність фундаментально змінює уявлення про компроміс між розміром моделі та обчисленнями на етапі використання.

Останні системи інтегрують масштабування під час генерації безпосередньо в процес навчання. Модель o1 від OpenAI [111] та DeepSeek-R1 [112] використовують навчання з підкріпленням для формування розширених ланцюжків міркувань – потенційно тисячі токенів на складні запити. Довжина та структура міркування адаптуються до складності задачі автоматично. DeepSeek-R1 демонструє, що навчання з підкріпленням без попереднього навчання з учителем дозволяє моделі самостійно виробити стратегії міркування, причому дистильовані версії зберігають значну частину цієї здатності за істотно менших обчислювальних витрат.

1.5.3. Масштабування під час генерації у дифузійних моделях

Дифузійні моделі від початку володіють природною можливістю варіювати обчислювальні витрати через регулювання кількості кроків розшумлення. Нагадаємо, що генеративний процес розпочинається з випадкового шуму $x_T \sim \mathcal{N}(0, I)$ та ітеративно уточнює наближення до цільового розподілу:

$$x_T \rightarrow x_{T-1} \rightarrow \dots \rightarrow x_1 \rightarrow x_0, \quad (1.37)$$

де кожен перехід $x_t \rightarrow x_{t-1}$ моделюється навченою нейронною мережею. За своєю суттю збільшення загальної кількості кроків T є прямим аналогом збільшення довжини ланцюжка міркувань у мовних моделях – обидва підходи виділяють додаткові обчислення для послідовного уточнення результату.

Водночас традиційне збільшення кроків розшумлення має обмежену ефективність. Як засвідчують емпіричні дослідження, після кількох десятків кроків приріст якості швидко згасає внаслідок накопичення похибок апроксимації та дискретизації. Саме це спостереження мотивувало пошук альтернативних механізмів масштабування обчислень під час генерації [105].

Одним із напрямків є оптимізація процесу семплювання за фіксованого бюджету кроків. Сонг та колеги запропонували підхід DDIM [79] – детерміністичний процес, що дозволяє генерувати з тієї самої навченої моделі за значно меншою кількістю кроків порівняно зі стандартним підходом. Лу та співавтори розвинули цю ідею через DPM-Solver [80], застосувавши вдосконалені методи чисельного інтегрування до диференціальних рівнянь дифузії. Це дозволило досягати якості, порівнянної з сотнями кроків, лише за десять-двадцять оцінок моделі. Ксу та колеги запропонували „перезапуск семплювання“ [113] – метод, що періодично повертає процес до вищих рівнів шуму для виправлення похибок, накопичених під час розшумлення. Усі ці підходи оптимізують компроміс між якістю та обчислювальним

бюджетом без потреби перенавчання моделі.

Ма та співавтори запропонували принципово новий підхід, що розглядає масштабування під час генерації як задачу пошуку оптимальних початкових шумів для дифузійного процесу. Замість спрямування всіх обчислень виключно на кроки розшумлення, частина ресурсів виділяється на дослідження простору початкових шумів із використанням верифікаторів для оцінювання якості проміжних результатів. Цей підхід демонструє концептуальну схожість із методами пошуку в мовних моделях, де верифікатор оцінює якість згенерованих відповідей [105].

Структура запропонованого методу включає два основні компоненти: верифікатори для надання зворотного зв'язку та алгоритми пошуку для знаходження кращих кандидатів шуму. Серед верифікаторів досліджено оракульні (з доступом до фінальної метрики), контрольовані (на основі попередньо навчених класифікаторів) та ансамблеві підходи. Алгоритми пошуку варіюються від випадкового пошуку до локальних методів нульового порядку та пошуку вздовж траєкторій [105].

Механізми керованої генерації додають обчислення для направлення процесу розшумлення. „Спрямування за класифікатором“ та „спрямування без класифікатора“ [114] використовують додаткові оцінки на кожному кроці для спрямування генерації відповідно до бажаних атрибутів зображення. Це збільшує обчислювальну вартість кожного кроку, але дозволяє контролювати властивості результату без перенавчання моделі. Сила спрямування регулює баланс між різноманітністю та відповідністю заданим умовам – ще один параметр масштабування під час генерації.

Експериментальні результати на задачах генерації зображень продемонстрували суттєве покращення якості порівняно з простим збільшенням кроків розшумлення. Важливо, що спостережувана закономірність масштабування має характер, подібний до того, що виявлено для мовних моделей: якість зростає логарифмічно або за степеневим законом із збільшенням

обчислювального бюджету на етапі генерації.

1.5.4. Масштабування під час генерації у моделях узгодження потоків

Моделі узгодження потоків розширюють принципи масштабування під час генерації через контроль точності чисельного інтегрування. Як показано вище, генерація у цих моделях відбувається шляхом розв'язання звичайного диференціального рівняння $dx/dt = v_\theta(x, t)$, де v_θ – вивчене векторне поле, що визначає траєкторію від базового розподілу p_0 до цільового p_1 . У цьому контексті масштабування під час генерації реалізується через регулювання кількості кроків чисельного інтегратора, причому збільшення кроків відповідає виділенню більших обчислювальних ресурсів для точнішого наближення неперервної траєкторії.

Порівняно з дифузійними моделями узгодження потоків забезпечує пряміші траєкторії від шуму до даних завдяки детерміністичній природі процесу. Ця властивість полегшує апроксимацію векторного поля та позитивно впливає на якість генерації за обмеженої кількості кроків. Застосування методів оптимального транспорту для спряження елементів вихідного та цільового розподілів додатково вирівнює траєкторії руху та пришвидшує збіжність.

Дослідження масштабованості узгодження потоків для складних наукових задач моделювання підтвердили ефективність цього підходу [115]. Моделі продемонстрували здатність точніше апроксимувати складні розподіли при збільшенні кількості кроків інтегрування.

1.5.5. Масштабування у інших класах генеративних моделей

Масштабування під час генерації виходить за межі авторегресивних та дифузійних архітектур. Нейронні поля випромінювання (NeRF) для

синтезу просторових сцен – один із прикладів. NeRF застосовує адаптацію параметрів під час тестування, виділяючи додаткові градієнтні кроки для настроювання моделі під конкретну вхідну сцену [116]. Збільшення бюджету тестової оптимізації підвищує точність тривимірної реконструкції [117], демонструючи той самий компроміс між обчисленнями та якістю. Генеративно-змагальні мережі – інший клас. Вони можуть використовувати ітеративне уточнення латентних векторів або генерувати множину зразків із подальшим відбором найкращих дискримінатором.

1.6. Оцінювання генеративних моделей: якість та обчислювальна вартість

Оцінка та порівняння генеративних моделей ускладнені самою природою задачі. У дискримінативних задачах кожному входу відповідає відома правильна відповідь, і якість моделі зводиться до частки збігів або до середньої похибки. Генеративна модель еталона для окремого зразка не має: зіставляти доводиться цілі розподіли, доступні лише через скінченні вибірки. Тайс та співавтори показали, що три поширені способи оцінювання – усереднена логарифмічна правдоподібність, оцінки Парзена та візуальна якість зразків – у просторах високої розмірності майже не пов’язані між собою [118]: високий результат за однією метрикою нічого не гарантує щодо інших. Модель може мати велику правдоподібність і водночас породжувати непереконливі зразки.

Відповіддю стала ціла низка непрямих метрик, кожна з яких спроможна охопити лише частину картини. Борджі налічує понад два десятки таких метрик і доходить висновку, що питання лишається відкритим [119]. Саджаді та співавтори звернули увагу на принципове обмеження одновимірної оцінки, а саме те, що вона не розрізняє двох різних проблем: коли модель породжує правдоподібні, проте одноманітні зразки, і коли вона покриває

весь розподіл, але неякісно [120]. Найпоширеніша нині відстань Фреше між розподілами ознак [121] також має власні вади, адже вона припускає нормальність розподілу ознак, залежить від обсягу вибірки й від мережі, якою ці ознаки отримано. Джаясумана та співавтори показали, що на моделях генерації за текстовим запитом ця метрика розходиться з оцінками людей і не відображає поступового поліпшення моделей [122].

Остаточним орієнтиром лишається оцінювання людьми. Стайн та колеги провели найбільше на сьогодні психофізичне дослідження сприйняття згенерованих зображень і зіставили його результати з 17 метриками: жодна з них не корелює сильно з людськими оцінками, а перцептивна перевага дифузійних моделей у поширених показниках просто не проявляється [123]. Проте таке оцінювання дороге й погано відтворюване. Отані та співавтори, оглянувши 37 робіт із генерації за текстовим запитом, виявили, що більшість із них спираються лише на автоматичні метрики або описують процедуру залучення людей настільки стисло, що відтворити результати просто неможливо [124].

У менш поширених предметних областях становище ще складніше. Занкі та співавтори, оглянувши 79 робіт із синтезу електрокардіограм, відзначають відсутність узгодженого набору метрик: кожне дослідження звітує про власну підмножину показників, що унеможливорює об'єктивне зіставлення методів [125]. Тому в роботах із генерації сигналів усталилася практика подавати кілька взаємодоповнювальних метрик одночасно [46], а також додатково оцінку прикладної користі синтетичних даних, зокрема через якість класифікатора, навченого з їх залученням [126].

Окремим аспектом, якого метрики якості не охоплюють зовсім, є обчислювальна вартість генерації. Для ітеративних моделей, таких як дифузійні та моделі узгодження потоків, її визначає кількість звернень до нейронної мережі під час породження одного зразка, яку позначають NFE (Number of Function Evaluations). Ця величина прямо задає час відгуку та енергоспожи-

вання і є саме тим ресурсом, який розподіляють методи масштабування обчислень. Порівняння якості поза цим контекстом частково втрачає сенс, адже результат майже будь-якої моделі можна поліпшити, збільшивши кількість кроків інтегрування. Робочою формою оцінювання стає залежність якості від заданого обчислювального бюджету, а не окреме число. Сюди ж належить вартість переходу до нового методу генерації, а саме те, чи потребує він повторного навчання, чи застосовний до вже навченої моделі.

Отже, для порівняння генеративних рішень ключовими є відповідність згенерованих даних цільовому розподілу та заданій умові, обчислювальна вартість генерації і властивості впровадження (потреба в перенавчанні та залежність від архітектури й модальності даних). Жодна окрема метрика не є універсальною, тому в цій роботі для кожної задачі застосовано власний набір взаємодоповнювальних метрик, а зіставлення методів проводиться за однакової кількості кроків інтегрування.

1.7. Висновки до розділу 1

Проведений аналіз охоплює весь діапазон підходів до генеративного моделювання – від класичних автокодувальників до методу узгодження потоків – і виявляє їхні спільні риси та принципові відмінності, що є необхідним підґрунтям для розгортання подальшого дослідження.

Попри архітектурне різноманіття, усі розглянуті моделі – варіаційні автокодувальники, авторегресивні системи, генеративні змагальні мережі, нормалізуючі потоки та дифузійні моделі – є різними способами розв’язати одну задачу: апроксимувати невідомий розподіл реальних даних і забезпечити можливість семплювання з нього. Кожен підхід пропонує власне бачення того, як параметризувати перетворення між простим базовим і складним цільовим розподілами, та передбачає відповідні компроміси щодо обраних властивостей – точність оцінювання густини, стабільність

навчання, швидкість інференсу чи якість синтезованих зразків. Цей погляд на генеративне моделювання як на простір компромісів довкола єдиного завдання трансформації розподілів є вихідною точкою для побудови узагальненої моделі генеративного процесу.

Окремий підсумок стосується масштабування обчислень під час генерації. Аналіз показав, що додаткові обчислення, витрачені на етапі породження зразка, підвищують якість результату в усіх розглянутих класах моделей – через довші ланцюжки міркувань у мовних моделях, більшу кількість кроків розшумлення у дифузійних та точніше чисельне інтегрування траєкторії в моделях узгодження потоків. Проте ці механізми описано в літературі розрізнено, у термінах кожної окремої архітектури, без спільного формального апарату, який дозволив би зіставляти їх між собою та переносити напрацьовані рішення з одного класу моделей на інший.

Ще один підсумок стосується способу зіставлення генеративних рішень. Огляд підходів до оцінювання засвідчив, що жодна окрема метрика якості не є універсальною, поширені показники розходяться між собою та з оцінками людей, а обчислювальна вартість генерації лишається поза їх увагою. Тому коректне порівняння методів потребує набору взаємодоповнювальних метрик і однакового обчислювального бюджету для всіх зіставляваних варіантів.

Найбезпосередніше до предмета дослідження стосується метод узгодження потоків. На відміну від дифузійних моделей, він навчає детерміністичне векторне поле, що транспортує зразки від базового розподілу до цільового вздовж неперервних траєкторій, причому цільове поле задається умовно – для конкретних пар (шум, дані) – без симуляції траєкторій під час навчання. Разом із тим аналіз показав, що залишається відкритим питання, яким має бути профіль швидкості вздовж траєкторії: стандартний підхід із постійною швидкістю є лише найпростішим варіантом, і більш виважений вибір цього профілю здатен покращити якість генерації за обмеженої кількості кроків

інтегрування.

Аналіз гібридних архітектур і прикладних областей підтвердив, що узгодження потоків уже є центральним компонентом найефективніших систем синтезу зображень – Stable Diffusion 3 [15], Flux [16] – і активно поширюється на задачі синтезу сигналів, що прямо стосується об'єкта дослідження.

Отже, прискорення генерації та розподіл обчислень під час неї є предметом інтенсивних досліджень, проте невирішеними залишаються два пов'язані питання. Бракує формальних моделей, які описували б генеративний процес і розподіл обчислень під час генерації незалежно від конкретної архітектури. Бракує й методів, у яких обмежений обчислювальний бюджет враховано в самій побудові процесу генерації, а не лише у способі його дискретизації чи в доопрацюванні вже навченої моделі. У результаті виникає науково-прикладна задача розроблення методів нейромережевої генерації об'єктів, які дають змогу ефективніше використовувати наявний обчислювальний бюджет, забезпечуючи високу якість синтезованих зразків.

РОЗДІЛ 2

ПОБУДОВА МОДЕЛЕЙ ГЕНЕРАТИВНОГО ПРОЦЕСУ ТА МАСШТАБУВАННЯ ОБЧИСЛЕНЬ ПІД ЧАС ГЕНЕРАЦІЇ

2.1. Узагальнена модель генеративного процесу

Попередній аналіз основних класів генеративних моделей виявив суттєву відмінність у їхніх архітектурах, методах навчання та теоретичному обґрунтуванні. Варіаційні автокодуювальники спираються на варіаційне наближення та максимізацію нижньої оцінки правдоподібності, генеративні змагальні мережі використовують мінімаксну гру між двома нейронними мережами, нормалізуючі потоки базуються на формулі заміни змінних, а дифузійні моделі та методи узгодження потоків описуються через стохастичні та звичайні диференціальні рівняння. Незважаючи на ці розбіжності, у цьому підрозділі побудовано узагальнену модель генеративного процесу, яка охоплює всі перелічені підходи, подаючи кожен із них як функцію трансформації ймовірнісних розподілів [19, 21].

2.1.1. Уніфікована теоретична перспектива

Концептуальною основою для розуміння генеративних моделей слугує гіпотеза многовиду [29], яка стверджує, що високовимірні дані реального світу зосереджені поблизу низьковимірного многовиду, вкладеного в простір ознак. Ця геометрична перспектива природно приводить до ідеї трансформації розподілів: генеративна модель вивчає відображення з простого низьковимірного розподілу на многовид даних у високовимірному просторі. Хоча частина моделей будує відображення безпосередньо у просторі даних, інші оперують у латентному просторі, але всі вони реалізують

ідею трансформації розподілів.

Функція трансформації ймовірності є математичною операцією, що відображає випадкову величину з одного розподілу ймовірності у нову випадкову величину з іншим цільовим розподілом. Ця концепція належить до фундаментальних понять теорії ймовірностей і забезпечує механізм модифікації розподілів, що є ключовим для моделювання складних реальних явищ.

У формальних математичних термінах, нехай задано випадкову величину X з функцією густини ймовірності $p_X(x)$ та функцію перетворення f , яка породжує нову величину $y = f(x)$. Тоді густина ймовірності трансформованої величини $p_Y(y)$ виражається через класичну формулу заміни змінних [2]:

$$p_Y(y) = p_X(f^{-1}(y)) \left| \det \left(\frac{\partial f^{-1}(y)}{\partial y} \right) \right|. \quad (2.1)$$

Ця формула забезпечує збереження маси ймовірності під час трансформації, гарантуючи, що інтеграл від функції густини залишається рівним одиниці.

Математичний каркас функцій трансформації ймовірностей створює теоретичну основу для генеративних моделей, котрі навчаються відображати прості розподіли ймовірності у складніші цільові розподіли даних. Ключовим спостереженням є те, що процес генерації у переважній більшості сучасних моделей можна розглядати як послідовність перетворень, що трансформують елементи наперед визначеного простого розподілу в елементи складного розподілу навчальних даних.

Це спостереження дозволяє задати генеративну трансформацію в термінах, спільних для всіх розглянутих класів моделей.

Означення 2.1. Нехай $(\mathcal{Z}, p_0)$ – простір із наперед заданим базовим розподілом, а $\mathcal{X}$ – простір даних із невідомим розподілом p_1 . Функцією трансформації розподілів називаємо вимірне відображення $F_\theta : \mathcal{Z} \rightarrow \mathcal{X}$,

параметризоване $\theta \in \Theta$, яке породжує на $\mathcal{X}$ образ базового розподілу

$$p_\theta = (F_\theta)_\# p_0, \quad p_\theta(B) = p_0(F_\theta^{-1}(B)) \quad (2.2)$$

для кожної вимірної множини $B \subseteq \mathcal{X}$.

Співвідношення (2.2) задає розподіл p_θ для довільного вимірного F_θ , тоді як формула заміни змінних (2.1) дає його густину лише за оборотного диференційовного відображення. Генерація зразка полягає у семплюванні $z \sim p_0$ та обчисленні $x = F_\theta(z)$, а навчання добирає параметр θ так, щоб наблизити p_θ до p_1 . Спосіб цього добору й відрізняє класи генеративних моделей між собою, тож в описі моделі його належить виділити окремо від самої трансформації.

Означення 2.2. Узагальненою моделлю генеративного процесу називаємо кортеж

$$\mathcal{G} = \langle p_0, F_\theta, \mathcal{L}, \mathcal{A} \rangle, \quad (2.3)$$

компоненти якого визначено так:

1. базовий розподіл p_0 на латентному просторі $\mathcal{Z}$ – наперед заданий простий розподіл, що слугує джерелом стохастичності;
2. функція трансформації розподілів F_θ за означенням 2.1, яка відображає $\mathcal{Z}$ у простір даних $\mathcal{X}$ і породжує розподіл $p_\theta = (F_\theta)_\# p_0$;
3. цільова функція навчання $\mathcal{L}$, мінімізація якої визначає параметр θ ;
4. множина допоміжних відображень $\mathcal{A}$ – компоненти, задіяні лише під час навчання: енкодер, дискримінатор, функція оцінювання чи модель зашумлення.

Генерація зразка задіює лише дві перші компоненти кортежу (2.3): достатньо взяти $z \sim p_0$ й обчислити $x = F_\theta(z)$, тоді як $\mathcal{L}$ та $\mathcal{A}$ у цьому обчисленні участі не беруть. Саме цей поділ і робить модель узагальненою: за однакової ролі F_θ класи генеративних моделей відрізняються переважно компонентами $\mathcal{L}$ та $\mathcal{A}$.

Функція трансформації не обов'язково є одноразовим відображенням. У загальному випадку вона розкладається в композицію $F_\theta = f_\theta^{(K)} \circ \dots \circ f_\theta^{(1)}$ з $K \geq 1$ послідовних кроків, і саме значення K визначає обчислювальну вартість генерації: генеративні змагальні мережі та автокодувальники відповідають $K = 1$, нормалізуючі потоки – кількості шарів перетворення, дифузійні моделі та узгодження потоків – кількості кроків розшумлення чи чисельного інтегрування, авторегресивні моделі – довжині згенерованої послідовності. Саме ця компонента моделі й стає предметом масштабування обчислень, розглянутого в підрозділі 2.2.

Область застосовності кортежу (2.3) обмежена трьома умовами. По-перше, базовий розподіл p_0 фіксований і не залежить від θ : моделі, що задають розподіл неявно – через функцію енергії чи ненормовану густину, без відображення з наперед заданого розподілу, – до цього кортежу не зводяться. По-друге, модель описує структуру генеративного процесу, а не якість наближення: вона задає розподіл p_θ , тоді як розходження $D(p_\theta \| p_1)$ лишається поза кортежем, а рівність $p_\theta = p_1$ у загальному випадку недосяжна. По-третє, формула (2.1) застосовна лише за оборотного диференційовного F_θ ; для решти класів розподіл p_θ визначено співвідношенням (2.2), проте його густина може не мати замкненої форми.

Побудована модель дозволяє абстрагуватися від специфічних деталей реалізації та зосередитися на спільній структурі генеративного процесу. Незалежно від того, чи модель використовує латентний простір із нормальним розподілом, чи дискретні коди з книги кодів, чи багатокрокову процедуру знешумлення – фундаментальна операція залишається тією самою: перетворення відомого вхідного розподілу у цільовий розподіл даних.

2.1.2. Аналіз різних моделей через призму трансформації розподілів

Застосуємо побудовану модель для систематичного аналізу кожного класу генеративних моделей. Цей підхід дозволить виявити спільну структуру, що лежить в основі зовні різнорідних архітектур.

Нормалізуючі потоки є найбільш прямолінійною реалізацією концепції функції трансформації розподілів. Фундаментальний принцип цих моделей полягає у вивченні оборотної функції F , яка відображає розподіл спостережуваних даних у стандартний нормальний розподіл. Якщо y є елементом розподілу даних p_{data} , то $z = F(y)$ розподілений згідно з $\mathcal{N}(0, I)$. Вимога оборотності уможливлює генерацію елементів складного розподілу через зворотне перетворення: $y = F^{-1}(z)$, де елементи z семплюються з нормального розподілу. Більш того, методологія навчання нормалізуючих потоків теоретично обґрунтована принципом збереження маси розподілу під час трансформації, що безпосередньо впливає з формули (2.1).

Дифузійні моделі та методи узгодження потоків природно вписуються у рамки неперервних нормалізуючих потоків, хоча й використовують іншу параметризацію трансформації. Узгодження потоків можна розглядати як узагальнення дифузійних моделей, що оперує в рамках неперервних трансформацій [12]. Обидва підходи базуються на векторних полях, які визначають диференціальні рівняння, що описують траєкторію трансформації між розподілами.

Ключовим аспектом цих моделей є вивчення залежних від часу векторних полів, котрі керують динамікою трансформації в кожній точці простору даних. Хоча методології навчання та оптимізації варіюються, загальний процес семплювання має послідовну структуру: елемент вибирається з початкового розподілу (гауссівського для дифузійних моделей) і проходить через послідовні трансформації. Ці трансформації параметризовані

своїм положенням уздовж часової траєкторії, врешті-решт збігаючись до елемента цільового розподілу.

Цей процес можна формально виразити як композитну функцію:

$$y = F(x) = f_{t_n} \circ f_{t_{n-1}} \circ \dots \circ f_{t_1}(x), \quad x \sim p_0, \quad (2.4)$$

де кожна f_{t_i} представляє трансформацію на часовому кроці t_i , $i = 1, \dots, n$. Це формулювання має фундаментальну схожість із нормалізуючими потоками, з важливою відмінністю: підхід зосереджується на окремих аспектах композитної функції та дозволяє варіювати кількість ітерацій залежно від конкретних вимог. Таким чином, повна функція F виступає функцією трансформації розподілу, відображаючи елементи вихідного розподілу p_0 у цільовий розподіл даних.

Генеративні змагальні мережі представляють окремий підхід порівняно з попередньо розглянутими моделями. Фундаментальна концепція цих мереж спирається на мінімаксну гру між двома нейронними мережами, що на перший погляд відхиляється від конвенційних трансформацій розподілів ймовірності. Проте під час передбачення процес виключно використовує генератор G , який приймає вхідний елемент із наперед визначеного розподілу, зазвичай стандартного нормального, і породжує елемент цільового розподілу:

$$y = G(z), \quad z \sim \mathcal{N}(0, I). \quad (2.5)$$

Отже, незважаючи на відмінність у навчанні, генератор може розглядатися як функція трансформації розподілу ймовірності під час передбачення, подібно до інших генеративних моделей.

Автокодувальники, включаючи їхні варіаційні та квантовані варіанти, можуть бути проаналізовані через подібну призму. Фаза передбачення охоплює лише компонент декодувальника, який отримує вхідні дані з наперед визначеного розподілу – нормального для варіаційних автокодувальників

або дискретного квантованого для VQ-VAE. Декодувальник функціонує як механізм трансформації розподілу:

$$y = D(z), \quad z \sim p_Z, \quad (2.6)$$

перетворюючи відомий вхідний розподіл p_Z у цільовий розподіл даних.

Авторегресивні моделі на перший погляд відрізняються від інших генеративних моделей відсутністю явного латентного розподілу як основи генеративного процесу. Проте детальніший аналіз виявляє складніший механізм. Передбачення наступних елементів у цих моделях включає як інформацію про попередні елементи, так і стохастичні компоненти. Ці стохастичні елементи є необхідними для забезпечення різноманітності генерованих послідовностей. Сучасні методи семплювання, зокрема top-k та ядерне семплювання, використовують випадкові величини з рівномірного розподілу для вибору tokenів [127].

Цей генеративний процес можна формально виразити як:

$$y_n = f(y_{n-1}, y_{n-2}, \dots, y_1, z), \quad z \sim \mathcal{U}(0, 1), \quad (2.7)$$

де z представляє випадкову величину, використовувану для семплювання. При розгортанні передбачень для всіх попередніх елементів рівняння набуває вигляду:

$$y_n = f'_n(z_n) \circ f'_{n-1}(z_{n-1}) \circ \dots \circ f'_1(x_1, z_1) = F(x_1, z_1, z_2, \dots, z_n), \quad (2.8)$$

що можна записати компактніше:

$$y_n = F(x_1, \mathbf{z}), \quad \text{де } \mathbf{z} = (z_1, z_2, \dots, z_n), \quad z_i \sim \mathcal{U}(0, 1). \quad (2.9)$$

Це формулювання виявляє, що авторегресивні моделі за своєю суттю функціонують як механізми трансформації розподілів, приймаючи семплюючі розподіли як вхідні дані та перетворюючи їх у елементи цільового розподілу.

2.1.3. Спільні властивості та відмінності

Проведений аналіз дозволяє встановити, у чому саме класи генеративних моделей збігаються з узагальненою моделлю (2.3), а в чому різняться між собою [21].

Фундаментальною спільною властивістю переважної більшості генеративних моделей є використання латентного простору з наперед відомим розподілом. Генерація нового екземпляра розпочинається саме із семплювання в цьому латентному просторі. З інженерної точки зору таке рішення є природним: модель потребує джерела стохастичності для породження різноманітних екземплярів, адже суто детерміністичні обчислення завжди давали б ідентичні результати.

На рисунку 2.1 наведено візуалізацію основних генеративних моделей у нетрадиційному представленні, де латентні простори та цільові простори вирівняні між різними моделями. Кольором виділено компоненти, що безпосередньо відповідають за трансформацію розподілів, тоді як компоненти, не використовувані під час передбачення, позначено сірим. Задля наглядності багатоступеневі трансформації в нормалізуючих потоках та дифузійних моделях зображено однією узагальненою функцією.

Таке вирівняне представлення наочно демонструє центральну ідею: ядром кожної з розглянутих генеративних моделей є функція трансформації ймовірності, яка відображає латентний розподіл у цільовий розподіл даних. Латентний простір завжди формується наперед визначеним простим розподілом. Відмінності ж між моделями переважно зводяться до способу побудови функції втрат для оптимізації цієї генеративної трансформації. Кольоровий поділ на рисунку прямо відповідає структурі кортежу (2.3): виділені компоненти складають пару (p_0, F_θ) , задіяну під час генерації, а позначені сірим – множину допоміжних відображень $\mathcal{A}$.

Відмінності між моделями проявляються на кількох рівнях:

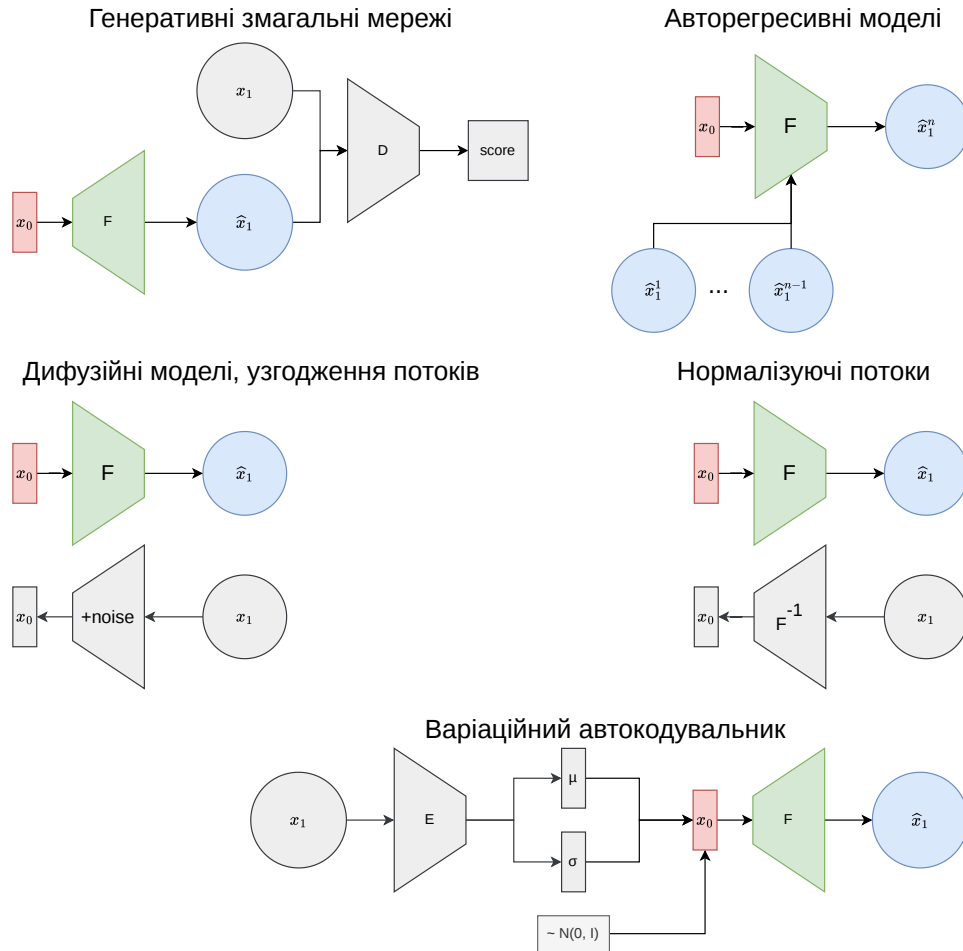


Рис. 2.1. Вирівняне представлення основних типів генеративних моделей

1. *Явність представлення густини.* Нормалізуючі потоки забезпечують точне обчислення густини розподілу, тоді як генеративні змагальні мережі моделюють розподіл неявно через параметри генератора. Варіаційні автокодувальники займають проміжне положення, надаючи лише нижню оцінку правдоподібності.
2. *Структура трансформації.* Нормалізуючі потоки використовують строго оборотні перетворення, дифузійні моделі та узгодження потоків – неперервні траєкторії у часі, а автокодувальники та генеративні змагальні мережі – прямі відображення без вимоги оборотності на рівні окремих шарів.

3. *Механізм оптимізації.* Кожен клас моделей визначає власну функцію втрат: негативна логарифмічна правдоподібність для нормалізуючих потоків, нижня оцінка ELBO для варіаційних автокодувальників, середньоквадратична похибка передбачення шуму для дифузійних моделей, середньоквадратична похибка передбачення швидкості для узгодження потоків та змагальна функція для генеративних змагальних мереж.
4. *Обчислювальна складність передбачення.* Автокодувальники та генеративні змагальні мережі потребують єдиного проходу через мережу, тоді як дифузійні моделі та узгодження потоків вимагають множинних ітерацій. Авторегресивні моделі мають складність, пропорційну довжині послідовності, що генерується.

Усі перелічені відмінності зводяться до значень компонент кортежу (2.3), як показано в таблиці 2.1. Компонента F_θ присутня в кожному рядку й відрізняється лише структурою та кількістю кроків K , тоді як цільова функція $\mathcal{L}$ і множина допоміжних відображень $\mathcal{A}$ змінюються від класу до класу.

Базовий розподіл p_0 у таблиці не наведено: для всіх класів, окрім авторегресивних моделей, ним є стандартний нормальний розподіл, а для авторегресивних – рівномірний розподіл на відрізку, з якого семплюється кожен наступний елемент послідовності.

Практичне значення узагальненої моделі проявляється у кількох аспектах. По-перше, вона сприяє перенесенню методологічних покращень між архітектурами моделей. Показовим прикладом є техніки оптимального транспорту, початково розроблені в контексті нормалізуючих потоків, які згодом було успішно адаптовано для покращення збіжності узгодження потоків [87]. По-друге, ця перспектива створює теоретичну базу для розробки універсальних підходів, що може сприяти появі ефективніших методів генеративного моделювання.

Побудована модель відкриває також нові напрямки для теоретичних

Таблиця 2.1

Класи генеративних моделей як окремі випадки узагальненої моделі

Клас моделей	Цільова функція $\mathcal{L}$	Допоміжні відображення $\mathcal{A}$	K
Варіаційні автокодувальники	нижня оцінка правдоподібності	енкодер	1
Генеративні змагальні мережі	змагальна функція	дискримінатор	1
Нормалізуючі потоки	логарифмічна правдоподібність	немає	шари
Дифузійні моделі	похибка передбачення шуму	модель зашумлення	кроки
Узгодження потоків	похибка передбачення швидкості	спряження розподілів	кроки
Авторегресивні моделі	логарифмічна правдоподібність токена	немає	довжина

досліджень. Вивчення математичних властивостей цих трансформацій може дати глибше розуміння можливостей та обмежень різних генеративних підходів. Крім того, ця перспектива полегшує розробку гібридних архітектур, що поєднують сильні сторони різних класів моделей, як-от латентні дифузійні моделі, котрі комбінують автокодувальники з дифузійним процесом.

2.2. Структурно-топологічна модель масштабування обчислень під час генерації

Попередній аналіз різних класів генеративних моделей продемонстрував, що всі вони реалізують концепцію трансформації ймовірнісних розподілів через послідовні обчислювальні операції. Розглянуті у першому розділі механізми масштабування обчислень під час генерації – від ланцюжків міркувань у великих мовних моделях до чисельного інтегрування

у моделях узгодження потоків – описано в літературі розрізнено, у термінах кожної окремої архітектури. Універсальність цього явища вказує на потребу у формальному описі принципу, незалежного від архітектури, та в єдиній математичній рамці для систематичного аналізу – від мовних моделей до NeRF. Таку рамку задає модель, побудована в цьому підрозділі [22]. Її названо структурно-топологічною, оскільки вона поєднує дві складові: структурну – декомпозицію методу на політики, що керують додатковими обчисленнями, і топологічну – форму обчислювального графу, у якій ці обчислення організовано.

2.2.1. Формальне означення моделі та її властивості

За всієї відмінності конкретних механізмів масштабування у них проглядає єдина структура. Додаткові обчислення під час генерації підвищують якість, проте це зростання має нелінійний характер. Це дозволяє задати метод масштабування формально.

Означення 2.3. Нехай f_θ – базова модель, що генерує вихід через один прямий прохід з обчислювальною вартістю $C_{\min}$, а Q – метрика якості результату. Структурно-топологічною моделлю масштабування обчислень під час генерації називаємо оператор

$$y^* = \text{ITS}(f_\theta, x, C), \quad C > C_{\min}, \quad (2.10)$$

заданий трійкою політик

$$\text{ITS} = \langle \pi_{\text{gen}}, \pi_{\text{sel}}, \pi_{\text{alloc}} \rangle, \quad (2.11)$$

компоненти якої визначено так:

1. *політика генерації* π_{gen} визначає, як створюються додаткові виходи чи кроки уточнення – послідовно, паралельно чи через пошук з розгалуженням;

2. *політика відбору* π_{sel} встановлює, як результати агрегуються або вибираються – через усереднення, голосування, оцінювання верифікатором або збіжність;
3. *політика виділення обчислень* π_{alloc} керує тим, як бюджет C розподіляється між генерацією та відбором.

Оператор (2.10) підпорядковано умові неспадання якості

$$C_2 > C_1 \geq C_{\min} \implies \mathbb{E}[Q(\text{ITS}(f_\theta, x, C_2))] \geq \mathbb{E}[Q(\text{ITS}(f_\theta, x, C_1))]. \quad (2.12)$$

Трійка політик (2.11) не залежить від домену, тож у її термінах описуються методи з різних галузей машинного навчання. Дифузійне семплювання – послідовна генерація з неявним відбором та фіксованою кількістю кроків. Метод самоузгодженості – паралельна генерація з голосуванням більшістю та фіксованим розподілом бюджету. Пошук Монте-Карло по дереву – деревоподібна генерація з відбором за критерієм UCB за обмеженого бюджету симуляцій.

Умова (2.12) задає лише напрям зміни якості, тому межі моделі визначають три додаткові обмеження.

Перше обмеження стосується вартості. Кожен додатковий вихід чи крок уточнення потребує щонайменше одного прямого проходу базової моделі, тож за n таких проходів

$$C(n) \geq n C_{\min}. \quad (2.13)$$

Вартість зростає щонайменше лінійно з обсягом додаткової роботи.

Друге обмеження стосується приросту якості: він сублінійний,

$$Q(C) \propto C^\alpha, \quad \alpha < 1, \quad \text{або} \quad Q(C) \propto \log C. \quad (2.14)$$

На відміну від (2.13), це співвідношення є емпіричною закономірністю, а не наслідком означення 2.3: воно спирається на вимірювання в конкретних доменах, зокрема на степеневу залежність якості від обчислювального бю-

джету у мовних моделях [99, 101]. Стійкість цієї закономірності в різних галузях розглянуто в підрозділі 2.2.3.

Третє обмеження стосується відбору. Якість результату не може перевищити ані якості згенерованих кандидатів, ані якості механізму, що обирає з них найкращого:

$$Q_{ITS} \leq \min(Q_{gen}, Q_{ver}), \quad (2.15)$$

де Q_{gen} – якість, обумовлена генератором, а Q_{ver} – якість, обумовлена верифікатором. Покриття (ймовірність наявності принаймні одного правильного рішення серед n зразків) зростає як степенева функція кількості семплів [100], проте практична точність досягає плато раніше, щойно механізм відбору насичується. Найслабше це обмеження виявляється в доменах з автоматизованою швидкою верифікацією: модульні тести у програмуванні, автоматична перевірка доведень у математиці, детерміновані результати в іграх.

Трійка політик описує, з чого складається метод масштабування, проте не вказує, як саме додаткові обчислення організовано в просторі та часі. Цю другу складову моделі задає топологія обчислювального графу.

2.2.2. Базові обчислювальні топології масштабування

Топологічну складову моделі утворюють шість базових обчислювальних топологій [23]. Кожна з них відповідає окремому способу організації додаткових обчислень від послідовного ланцюжка уточнень до паралельного семплювання множини варіантів, від деревовидного пошуку до адаптивного маршрутизування. Топології не є жорсткими межами між методами: реальні системи часто поєднують кілька з них, тож ідеться про базис, у якому метод розкладається, а не про взаємовиключні категорії.

Перші три топології організують обчислення за просторовою формою графу: *послідовне уточнення* будує лінійний ланцюжок залежних перетво-

рень, *паралельне семплювання* розгортає незалежні гілки, що сходяться в агрегаційному вузлі, а *дерезовидний пошук* поєднує обидві ідеї, динамічно розгалужуючи та обрізаючи простір часткових рішень. Наступні дві вводять зовнішні механізми корекції: *адаптація на етапі інференсу* модифікує самі параметри моделі, тоді як *керована генерація* зберігає параметри незмінними, проте вбудовує зовнішній сигнал у кожен крок декодування. Шоста топологія *адаптивна маршрутизація* займає особливе місце: методи цієї групи не нарощують загальний обчислювальний бюджет, а перерозподіляють наявний ресурс залежно від складності конкретного входу. Форму обчислювального графу для всіх шести топологій показано на рисунку 2.2.

Тип І: Послідовне уточнення. Вихід створюється через ланцюжок залежних кроків – кожен крок приймає результат попереднього як вхід: $x_0 \rightarrow f(x_0) = x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n$. Обчислювальна складність зростає як $O(n \cdot C_{\text{step}})$.

Політика генерації: ітеративна трансформація.

Політика відбору: зазвичай фінальний x_n (неявний відбір через збіжність).

Політика виділення обчислень: фіксована або адаптивна кількість кроків.

Найповніше цей тип представлений в ітеративному розшумленні дифузійних моделей [7]: кожен крок знімає частину шуму і передає уточнений стан наступному. Різниця між конкретними методами зводиться до того, як саме організована ця траєкторія – DDIM скорочує її за рахунок детермінованого планування [79], DPM-Solver замінює наївне дискретизування ефективнішими чисельними схемами [80], а перезапуск семплювання повертає процес на кілька кроків назад, щоб вийти з локально незадовільних станів [113]. Моделі узгодження потоків будуються на тому самому принципі: додаткові кроки числового інтегрування безпосередньо знижують похибку траєкторії [12]. У генерації тексту цей механізм з'являється у вигляді

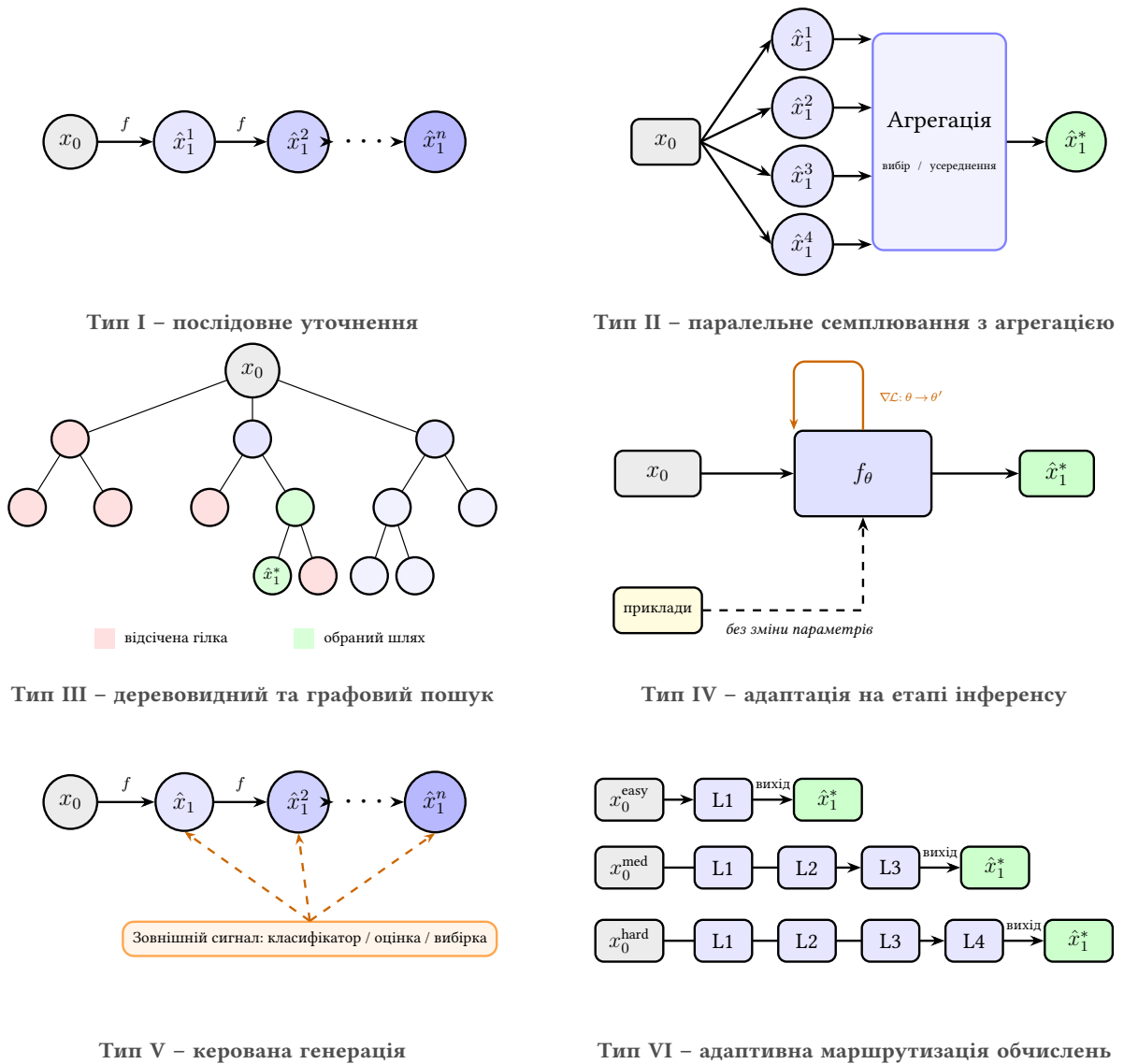


Рис. 2.2. Базові обчислювальні топології масштабування обчислень під час генерації

ді ланцюжка роздумів [108] – модель поетапно формулює проміжні кроки замість того, щоб видавати відповідь одразу, а системи o1 і DeepSeek-R1 навчаються генерувати такі ланцюжки через навчання з підкріпленням, ефективно перетворюючи час інференсу на ресурс для вирішення складних задач [111, 112].

Перевага: гарантована прогресія до кращого рішення.

Обмеження: послідовна природа унеможливорює паралелізацію, що

ускладнює використання розподілених обчислювальних ресурсів.

Тип II: Паралельне семплювання з агрегацією. Множина незалежних виходів генерується з того самого входу, потім комбінується або фільтрується. Складність $O(n \cdot C_{\text{single}})$, повністю паралелізується.

Політика генерації: незалежне семплювання n разів.

Політика відбору: голосування, усереднення або на основі верифікатора.

Політика виділення обчислень: кількість зразків.

Всі методи даного типу генерують певну кількість кандидатів паралельно, а потім агрегують результат. Відмінності між конкретними методами визначаються насамперед тим, яким саме чином відбувається агрегація. Коли є зовнішній критерій правильності, результати фільтрують через нього – так влаштований підхід „найкраще з N “ з моделями нагород [128]: із множини кандидатів залишають лише ті, що одержали найвищу оцінку верифікатора. Якщо ж такого зовнішнього критерію немає, агрегують за внутрішніми ознаками, наприклад метод самоузгодженості трактує більшість як сигнал коректності [109], а класичні ансамблеві методи [129, 130], аугментація тестових зразків [131] та дропаут Монте-Карло [132] усереднюють виходи, знижуючи дисперсію без явної фільтрації. Важлива емпірична закономірність: покриття (ймовірність хоча б одного правильного кандидата серед n зразків) зростає степеневі з збільшенням n [100], проте загальна точність виходить на плато набагато раніше, оскільки механізм відбору не завжди спроможний відрізнити правильне від правдоподібного.

Перевага: паралельне виконання без обміну даними між гілками.

Обмеження: агрегація, як вузьке місце, обмежує практичну ефективність.

Тип III: Деревовидний та графовий пошук. Методи даного типу досліджують розгалужений простір часткових рішень, оцінюючи та обрізаючи гілки під час пошуку. Складність у найгіршому випадку $O(b^d)$ для коефіцієнта розгалуження b та глибини d , проте ефективно обрізання робить

це керованим.

Політика генерації: розгалуження дерева пошуку та продовження гілок.

Політика відбору: пошук з евристичним керуванням (UCB, A^* , ширина променя).

Політика виділення обчислень: баланс між глибиною та шириною пошуку.

Класичним представником є променевий пошук [133], що на кожному кроці декодування підтримує w найперспективніших часткових послідовностей – це вже дерево, а не лінійний ланцюжок, хоча й обмежено фіксованою шириною. Повноцінний деревовидний пошук реалізує метод пошуку Монте-Карло по дереву: він циклічно виконує вибір вузла, розширення, симуляцію до термінального стану та зворотне поширення оцінки [134]. Саме цей механізм дозволив AlphaGo, AlphaZero і MuZero [135, 136] масштабувати якість гри пропорційно кількості симуляцій – аргумент, що пізніше мотивував перенесення MCTS до простору міркувань мовних моделей у вигляді RAP [137], LATS [138] та методу дерева роздумів [110]. Метод графа думок [139] виходить за межі деревовидної топології, допускаючи цикли у просторі часткових рішень. Все це – варіації одного принципу, відомого ще з класичного A^* [140]: витратити більше обчислень на пошук, щоб здобути якісніше рішення.

Перевага: адаптивне дослідження простору рішень.

Обмеження: складність розробки якісної евристики оцінювання вузлів.

Тип IV: Адаптація на етапі інференсу. Замість генерації більшої кількості виходів ці методи оновлюють параметри моделі, використовуючи сам тестовий вхід.

Політика генерації: процес адаптації параметрів моделі θ' під вхід x .

Політика відбору: на основі функції втрат на етапі інференсу (самоконтрольованої або метричної).

Політика виділення обчислень: кількість градієнтних кроків.

Загальна ідея полягає у тому, щоб не нарощувати кількість виходів, а зробити так, щоб сама модель стала кращою на конкретному вході. Навчання на тестовому прикладі [117] використовує допоміжну самоконтрольовану задачу (наприклад, ротацію зображень або заповнення пропусків), щоб за кілька градієнтних кроків налаштувати модель до нового розподілу. В нейронних полях випромінювання NeRF такі кроки під конкретну сцену суттєво покращують реконструкцію [116]. TENT іде ще простішим шляхом [141]: замість повного зворотного поширення він оптимізує лише статистики нормалізації, мінімізуючи ентропію виходу (швидка адаптація без повного перенавчання). MAML [142] вирішує задачу в зворотному напрямку: на навчальному етапі він шукає ініціалізацію параметрів, з якої кілька градієнтних кроків на нових даних дають максимальний ефект. Особливим випадком є контекстне навчання у мовних моделях [37]: технічно воно не змінює параметри, проте кожен доданий приклад збільшує тривалість контексту і відповідно обчислювальну вартість виведення – що дозволяє включити його до цього типу за функціональним критерієм.

Перевага: персоналізація моделі під конкретний вхід.

Обмеження: потребує диференційованості та доступу до параметрів моделі.

Тип V: Керована генерація. Зовнішні сигнали тим чи іншим способом направляють процес генерації на кожному кроці, додаючи обчислювальні витрати на крок, але сприяючи більш якісній генерації.

Політика генерації: обрахунок і врахування додаткового спрямовуючого сигналу на кожному кроці.

Політика відбору: інтеграція сигналу керування у процес декодування.

Політика виділення обчислень: сила сигналу керування.

Різниця між конкретними методами – у тому, звідки береться керівний

сигнал. Спрямування за сигналом класифікатора [10] вимагає окремо навченої мережі, що обчислює градієнт у просторі зображень для кожного кроку розшумлення; практичне покращення якості при ціні прямих проходів двох нейромереж виявилось достатнім аргументом. Спрямування без класифікатора [114] позбувається цієї залежності, інтерполюючи умовний і безумовний прогнози безпосередньо в основній моделі – простіше в розгортанні, хоча й вимагає подвійного прямого проходу на кроці. У задачах мовного моделювання роль зовнішнього сигналу беруть на себе моделі винагороди за процес [128]: вони оцінюють не кінцевий вихід, а окремі кроки міркування, направляючи генерацію ще до завершення відповіді. Методи, доповнені пошуком [143], замість нагороди отримують у кожному запиті зовнішні документи – фактично довіряючи якості індексу та ретривера, а не навчній апріорній ймовірності моделі. У всіх випадках стеля якості визначається зовнішнім сигналом: слабкий класифікатор, неточна модель нагороди або поганий ретривер обмежують виграш незалежно від кількості кроків генерації.

Перевага: контроль якості через зовнішній незалежний сигнал.

Обмеження: якість верифікатора або керівного сигналу стає стелею ефективності.

Тип VI: Адаптивна маршрутизація обчислень. Тут загальний бюджет не збільшується, а перерозподіляється – різні входи отримують різну кількість обробки.

Політика генерації: різні графи обрахунків всередині моделі залежно від входу.

Політика відбору: на основі оцінки складності або впевненості моделі.

Політика виділення обчислень: ранній вихід, додаткові шари або активація підмножини експертів.

Архітектури з ранньою зупинкою [144] дозволяють впевненим передба-

ченням виходити з мережі, не проходячи глибших шарів – прості токени не повинні коштувати стільки ж, скільки складні. Моделі суміші експертів [145] реалізують подібну логіку на рівні параметрів: кожен токен активує лише підмножину ваг, де вибір підмножини залежить від самого токена. Каскади направляють запити через послідовно більші моделі, зупиняючись, коли впевненість перевищує поріг – дешева модель обробляє більшість запитів, а дорога залишається для справді складних випадків. Спекулятивне декодування [146] іде ще далі: легка чернеткова модель пропонує кілька токенів наперед, а важка верифікує їх паралельно – пропускна здатність зростає, хоча якість виходу залишається тією самою, що і без чернетки. Стратегії з явним бюджетом [147] перетворюють цю логіку на пряму директиву: обчислювальний поріг на запит задається явно, а не виводиться з архітектурних рішень. На відміну від перших п'яти типів, ці методи не наращують загальний бюджет – вони витрачають наявний ресурс точніше, розподіляючи його відповідно до реальної складності, а не рівномірно по всіх входах.

Перевага: ефективність через адаптацію бюджету до складності входу.

Обмеження: потребує надійної оцінки складності або впевненості в реальному часі.

Концептуальна єдність та реальні прояви. Усі шість типів є варіаціями одного принципу – балансу обчислень та якості. Спільна властивість: додаткові обчислення під час генерації підвищують якість виходу. Відмінність: топологія обчислювального графу – лінійний ланцюжок, паралельні гілки, дерево пошуку, модифікація параметрів, зовнішнє керування чи адаптивна маршрутизація. Усі ці методи піддаються формалізації через $ITS(f_\theta, x, C)$ та три політики: генерації, відбору, виділення обчислень.

Реальні системи часто комбінують типи. o1 та DeepSeek-R1 поєднують послідовне уточнення (Тип I) з неявним пошуком через навчені ланцюжки

міркувань (Тип III). Дифузійні методи зі спрямовуючим сигналом комбінують послідовне знешумлення (Тип I) з спрямуванням на основі сигналу класифікатора (Тип V). AlphaCode генерує мільйон програм паралельно (Тип II), потім фільтрує через модульні тести (верифікація). Особливо показово, що системи, спроектовані незалежно та в різних галузях, природно вкладаються у ту саму модель.

Показовим прикладом слугує система AlphaFold для передбачення тривимірної структури білків. Архітектура цієї системи включає механізм повторної обробки, що реалізує ітеративне уточнення передбаченої структури. На першій ітерації модель формує початкову гіпотезу про конформацію білка, яка типово містить значну кількість стереохімічних порушень – нефізичних значень довжин зв'язків та кутів між атомами. Вихідні представлення цієї ітерації подаються на вхід наступної, де мережа Evoformer має змогу скоригувати передбачення з урахуванням виявлених невідповідностей. Результати експериментів компанії DeepMind засвідчили, що перша ітерація досягає правильної топології з оцінкою GDT близько 78, проте з 90% стереохімічних порушень. До третьої ітерації оцінка GDT наближається до 90, а після восьми ітерацій кількість порушень прямує до нуля. Типова генерація використовує три-чотири ітерації повторної обробки, тоді як складні білки потребують сім і більше [148]. Така поведінка системи демонструє пряму аналогію з масштабуванням кроків розшумлення в дифузійних моделях: додаткові обчислення під час генерації забезпечують точніший результат. У термінах побудованої моделі це чистий приклад Типу I – послідовне уточнення, де кожна ітерація залежить від попередньої, подібно до DDIM у дифузії чи ланцюжку міркувань у мовних моделях. Структура обчислень ідентична: $x_0 \rightarrow f(x_0) = x_1 \rightarrow \dots \rightarrow x_n$, де кожен крок x_i є уточненням конформації білка.

Інший варіант масштабування під час генерації реалізує система AlphaCode для автоматичної генерації програмного коду. Вона застосовує

паралельне вибіркове семплювання з наступною фільтрацією та кластеризацією [149]. Для кожної задачі змагального програмування система генерує до мільйона кандидатів розв'язків, застосовуючи спеціально навчені трансформерні мережі з різними випадковими ініціалізаціями. Отримані програми фільтруються за допомогою тестових прикладів: приблизно десять тисяч кандидатів проходять початкову перевірку. Далі застосовується семантична кластеризація для групування подібних за поведінкою розв'язків, і фінально обирається не більше десяти програм для подання. У симульованих оцінюваннях на платформі Codeforces AlphaCode досягла середнього рейтингу в межах верхніх 54% учасників – рівня медіанного змагального програміста. Цей результат є першим випадком, коли штучний інтелект досяг конкурентоспроможного рівня у змагальному програмуванні. Ефективність масштабування тут визначається не кількістю згенерованих кандидатів, а якістю механізму верифікації та відбору. Це класичний Тип II (паралельне семплювання з агрегацією). Множина незалежних виходів $\{y_1, \dots, y_n\}$ генерується паралельно, потім фільтрується через верифікатор (модульні тести). Структурно ідентично до самоузгодженості у мовних моделях або „найкраще з N “ з навчання з підкріплення, проте масштаб набагато більший – мільйон кандидатів замість десятків. Насичення через вузьке місце верифікатора тут чітко проявляється: модульні тести пропускають лише 1% кандидатів, решта відсівається.

Принципово відмінний механізм масштабування реалізує система AlphaGo для гри в го, що застосовує пошук за методом Монте-Карло по дереву (MCTS) [134]. Для кожного ходу система виконує тисячі симуляцій, будуючи дерево можливих продовжень партії. Кожна симуляція проходить від поточної позиції до кінця гри, використовуючи навчену нейронну мережу для оцінювання позицій та ймовірностей ходів. Результати симуляцій агрегуються для визначення найперспективнішого ходу. Збільшення кількості симуляцій безпосередньо підвищує якість прийнятих рішень: зі ста

симуляціями система грає на аматорському рівні, з десятима тисячами – на професійному, а зі ста тисячами симуляцій досягає надлюдської майстерності. Перемога AlphaGo над чемпіоном світу Лі Седолем у 2016 році стала однією з найбільш резонансних подій у галузі штучного інтелекту. Формально MCTS реалізує обчислення апроксимації оптимальної функції цінності $V^*(s)$ для поточного стану гри, і точність цієї апроксимації зростає логарифмічно з кількістю виконаних симуляцій – закономірність, ідентична до тієї, що спостерігається в інших генеративних системах. У термінах моделі це Тип III (дерезовидний пошук). Обчислення досліджують розгалужений простір можливих продовжень, оцінюючи та обрізаючи гілки через UCS-керований відбір. Ця структура пізніше мігрувала до мовних моделей – „дерево роздумів“ та RAP використовують той самий механізм для дослідження простору міркувань.

Показово, що жодна з трьох систем не проектувалася з оглядом на обчислювальну топологію: AlphaFold народжувався як задача структурної біоінформатики, AlphaCode як застосування мовних моделей до змагального програмування, AlphaGo як аргумент про глибоке навчання в комбінаторному пошуку. Тим не менш кожна система природно займає одну з шести базових топологій, не маючи суттєвого перетину з іншою. Така незалежна збіжність переконливіша за будь-яку формальну аксіоматизацію: різні команди в різних галузях, вирішуючи якісно різні задачі, зупинились на обчислювальних топологіях, що повністю покриваються топологічною складовою моделі.

Топологічна складова моделі, таким чином, є не жорсткою межею між методами, а базисом для розуміння та систематичного проектування гібридних підходів [23].

2.2.3. Чотири фундаментальні інваріанти масштабування під час генерації

Систематичний аналіз методів різних типів виявляє закономірності, що проявляються незалежно від конкретної обчислювальної топології [23]. Розгляд понад п'ятдесяти методів із класичного машинного навчання, генеративного моделювання, алгоритмів пошуку та адаптивних обчислень дозволяє виділити чотири фундаментальні інваріанти – властивості, спільні для всіх шести типів масштабування.

Перший інваріант – універсальна сублінійна крива „вартість-якість“, задана співвідношенням (2.14). Незалежно від того, чи додаються кроки знешумлення до дифузійної моделі, токени міркування до авторегресивної системи, дерева до випадкового лісу чи симуляції до пошуку Монте-Карло по дереву – якість зростає з тим самим показником, меншим за одиницю. Ланцюжок міркувань у мовних моделях насичується після певної довжини ланцюжка [108]. Закон спадання віддачі від кроків знешумлення в дифузії добре задокументовані [7]. Так само і зростання якості генерації узгодженіми потоками сповільнюється зі збільшенням кроків інтегрування. Цей патерн настільки стійкий у різних доменах, що вказує на властивість самого ітеративного уточнення, а не артефакт конкретної архітектури чи навчального процесу [101].

Другий інваріант – вузьке місце верифікатора, формально виражене нерівністю (2.15). Усі методи, що спираються на відбір (Тип II, Тип III), стикаються з тим самим обмеженням: якість стратегії відбору обмежує ефективність. Метод самоузгодженості насичується, коли верифікатор не може відрізнити правильні рішення від правдоподібних неправильних [109]. Ефективність спрямовування без класифікатора в дифузії залежить від якості оцінювача [114]. Функція оцінювання в пошуку Монте-Карло по дереву – критичний компонент, що визначає стелю якості [134]. Покриття зростає

ступенево [100], проте практичний виграш стагнує, оскільки вибір правильного рішення стає вузьким місцем. Це вузьке місце проявляється найменше в доменах з автоматизованою верифікацією (модульні тести, автоматична перевірка доведень, детерміновані результати) і найсильніше там, де верифікація залежить від навчених моделей.

Третій інваріант – послідовно-паралельна дуальність. Майже кожен домен має обидва варіанти масштабування. У мовних моделях: ланцюжок міркувань (Тип I) проти самоузгодженості та променевого пошуку (Тип II). У дифузії: послідовне розшумлення (збільшення кроків) проти пошуку в просторі початкових шумів [105]. Пряму відповідність між нарощуванням кроків розшумлення в дифузійних моделях та подовженням ланцюжка міркувань у великих мовних моделях розглянуто в роботі [24]. У пошуку: обхід у глибину проти обходу в ширину. В різноманітних методах проявляється компроміс між глибиною (якість одного шляху) та шириною (покриття) генерації. Емпіричні результати для мовних моделей показують, що легкі задачі виграють від послідовного уточнення, а складні – від паралельного дослідження [99]. Поширення цієї закономірності на інші домени залишається відкритим емпіричним питанням, проте структурні передумови (спадна віддача на глибину, незалежна варіативність на ширину) присутні всюди, де застосовується масштабування під час генерації.

Четвертий інваріант – міграція методів між доменами. У різних доменах трапляються приклади застосування структурно подібних методів масштабування під час інференсу. Пошук Монте-Карло по дереву мігрував з ігор до міркувань мовних моделей [110, 137]. Логіка голосування з методу самоузгодженості з'являється в пошуку за рівнями шуму для дифузії. Керування через класифікатор у дифузії та декодування з винагородою у мовних моделях – структурно подібні реалізації Типу V. Топологічна складова моделі створює систематичну карту для виявлення таких переносів і, можливо важливіше, для ідентифікації напрямків, де перенесення ще не спробували.

2.2.4. Практичні наслідки та перспективи

Універсальність масштабування під час генерації відкриває нові можливості для оптимального розподілу обчислювальних ресурсів у генеративних системах. Зокрема, за фіксованого бюджету стратегія з меншою моделлю та більшою кількістю ітеративних уточнень здатна перевершувати підхід із великою моделлю без додаткових обчислень під час генерації. Це справедливо як для мовних моделей, де невелика модель з ланцюжком міркувань може перевершити модель у чотирнадцять разів більшу [99], так і для дифузійних моделей, де компактна архітектура з оптимізованим семплюванням досягає якості значно більших систем.

Систематичне дослідження цього феномену проведено в роботі про закони масштабування під час генерації. Автори експериментально продемонстрували, що модель Llemma з 7 мільярдами параметрів, оснащена вдосконаленими алгоритмами деревовидного пошуку, стабільно перевершує модель Llemma з 34 мільярдами параметрів за всіх досліджуваних обчислювальних бюджетів на бенчмарку MATH. Оптимальна стратегія масштабування під час генерації дозволяє досягати того самого рівня точності з використанням у сім разів меншого обсягу обчислень порівняно з простим семплюванням за методом „найкраще з N “. Ця закономірність суттєво впливає на економіку розгортання генеративних систем: за умов обмеженого бюджету доцільніше використовувати компактні моделі з інтелектуальними стратегіями генерації, ніж гігантські моделі з тривіальним одноразовим виведенням [101].

Принципова перевага масштабування під час генерації криється у можливості адаптивного виділення обчислень залежно від складності конкретного запиту: прості задачі вирішуються швидко з мінімальними ітераціями, тоді як складні отримують додаткові ресурси для досягнення необхідної якості. Така гнучкість залишається недосяжною в парадигмі масштабуван-

ня моделей, де розмір архітектури фіксується на етапі навчання і не може змінюватися залежно від вхідних даних.

Економічні наслідки цього зміщення парадигми є суттєвими. Традиційний підхід передбачав попередні інвестиції у тренування моделей – від десятків до сотень мільйонів доларів – з фіксованою вартістю виведення після завершення навчання. Натомість парадигма масштабування під час генерації пропонує знижені витрати на тренування менших моделей із змінною вартістю виведення, що адаптується до потреб користувачів. Така модель розподілу витрат знижує бар'єр доступу до ресурсомістких генеративних систем, оскільки менші команди та організації отримують змогу створювати конкурентоспроможні рішення без доступу до інфраструктури масштабу великих технологічних компаній.

Класифікація за обчислювальною топологією створює основу для розробки практичних стратегій вибору методу. Для легких запитів достатньо послідовного уточнення (Тип I) з невеликим бюджетом – ланцюжку роздумів з коротким ланцюжком або декілька кроків знешумлення. Для важких запитів ефективніше паралельне семплювання (Тип II) з верифікацією або деревовидний пошук (Тип III) для структурованих задач. Емпіричні результати підтверджують цей розподіл: невеликі моделі з правильно обраною стратегією масштабування систематично перевершують великі моделі з тривіальним виведенням [99, 147].

Перспективним напрямком є розробка адаптивних політик виділення обчислень [23]. Замість фіксованого бюджету система може динамічно вибирати тип і виділяти ресурси залежно від оцінки складності входу. Легкі запити обробляються мінімальним Тип I. Важкі запити отримують максимальний Тип II або Тип III. Оцінка складності – попередній прогон з малим бюджетом – визначає необхідний рівень обчислень. Така адаптивна маршрутизація поєднує ефективність (малі витрати на легкі задачі) з якістю (достатні ресурси на важкі задачі), оптимізуючи загальну продуктивність

системи.

2.3. Висновки до розділу 2

У другому розділі дисертації побудовано дві моделі: узагальнену модель генеративного процесу, що охоплює основні класи генеративних моделей, та структурно-топологічну модель масштабування обчислень під час генерації.

Встановлено, що попри суттєві відмінності в архітектурах, методах навчання та теоретичному обґрунтуванні, всі основні класи генеративних моделей – варіаційні автокодувальники, генеративні змагальні мережі, нормалізуючі потоки, дифузійні моделі, методи узгодження потоків та авторегресивні моделі – зводяться до єдиного кортежу, що містить базовий розподіл, функцію трансформації розподілів, цільову функцію навчання та множину допоміжних відображень. Показано, що генерація зразка задіює лише дві перші компоненти кортежу, тоді як відмінності між класами зосереджено у двох останніх; сам процес генерації полягає у відображенні елементів наперед визначеного простого розподілу в елементи складного розподілу даних, що формалізується через класичну формулу заміни змінних або її узагальнення. Межі моделі задано трьома умовами: фіксованістю базового розподілу, відокремленням структури процесу від якості наближення розподілу даних та застосовністю формули заміни змінних лише до оборотних трансформацій.

Показано, що розмаїття механізмів масштабування обчислень під час генерації, виявлене у першому розділі, піддається єдиному описові та виходить за межі власне генеративних моделей. У великих мовних моделях цей принцип реалізується через техніки ланцюжків міркувань, паралельне семплювання з агрегацією та деревовидний пошук у просторі міркувань. У дифузійних моделях та моделях узгодження потоків – через регулювання

кількості кроків розшумлення або чисельного інтегрування, оптимізацію процесу семплювання та пошук оптимальних початкових умов. Аналіз показав, що подібні механізми присутні в різноманітних системах штучного інтелекту, включаючи передбачення структури білків (AlphaFold), автоматичну генерацію програмного коду (AlphaCode) та стратегічні ігри (AlphaGo).

Побудовано структурно-топологічну модель масштабування обчислень під час генерації. Структурну складову моделі утворює оператор $ITS(f_\theta, x, C)$, заданий трійкою політик – генерації, відбору та виділення обчислень – і підпорядкований умові неспадання математичного сподівання якості зі зростанням обчислювального бюджету C . Межі моделі виражено в математичній формі трьома співвідношеннями: щонайменше лінійним зростанням вартості, сублінійним приростом якості та обмеженням результату з боку генератора й верифікатора. Топологічну складову моделі утворюють шість базових обчислювальних топологій, виділених на основі аналізу методів із різних галузей машинного навчання: послідовне уточнення, паралельне семплювання з агрегацією, деревовидний та графовий пошук, адаптація на етапі інференсу, керована генерація та адаптивна маршрутизація обчислень.

Виявлено чотири фундаментальні інваріанти, що проявляються незалежно від конкретної обчислювальної топології. По-перше, універсально сублінійна крива „вартість-якість“, де якість зростає як степенева функція обчислень з показником меншим за одиницю або логарифмічно. По-друге, вузьке місце верифікатора, що обмежує ефективність методів, які спираються на відбір. По-третє, послідовно-паралельна дуальність, коли майже кожен домен має обидва варіанти масштабування з різними компромісами між глибиною та шириною дослідження. По-четверте, міграція методів між доменами після виявлення їхньої спільної структури.

Обґрунтовано практичне значення масштабування під час генерації

для оптимального розподілу обчислювальних ресурсів. Встановлено, що за фіксованого бюджету невелика модель з додатковими обчисленнями під час генерації здатна перевершити модель у чотирнадцять разів більшу за розміром. Оптимальна стратегія масштабування під час генерації дозволяє досягати того самого рівня точності з використанням у сім разів меншого обсягу обчислень порівняно з тривіальним семплюванням. Ця закономірність фундаментально змінює уявлення про компроміс між розміром моделі та обчисленнями на етапі використання, створюючи передумови для зміщення парадигми оптимізації від збільшення розміру моделей до інтелектуального розподілу обчислень.

Топологічна складова моделі створює теоретичну основу для систематичного вибору стратегії масштабування залежно від характеристик задачі. Для легких запитів достатньо послідовного уточнення з невеликим бюджетом, тоді як складні запити виграють від паралельного дослідження або деревовидного пошуку. Перспективним напрямком є розробка адаптивних політик виділення обчислень, що динамічно вибирають тип масштабування та визначають обсяг ресурсів залежно від оцінки складності входу, оптимізуючи баланс між ефективністю та якістю.

Результати розділу демонструють, що масштабування під час генерації є не артефактом конкретних архітектур чи навчальних процесів, а фундаментальною властивістю інтелектуально насичених задач, що створює нову парадигму проектування та розгортання генеративних систем за обмежених ресурсів.

РОЗДІЛ 3

РОЗРОБКА УДОСКОНАЛЕНОГО МЕТОДУ УЗГОДЖЕННЯ ПОТОКІВ НА ОСНОВІ ЗАСТОСУВАННЯ ПРОФІЛІВ ШВИДКОСТЕЙ

У розділі 1 узгодження потоків розглянуто на основі лінійної інтерполяції між шумом і даними, для якої цільове векторне поле зводиться до сталого зміщення $x_1 - x_0$, а генерація відбувається через інтегрування звичайного диференціального рівняння. У цьому розділі запропоновано узагальнення зазначеної побудови: стале векторне поле замінюється часозалежним, що дає змогу керувати темпом руху зразка вздовж траєкторії, не змінюючи ні архітектури мережі, ні характеру спряження розподілів. Нижче подано побудову й формальне означення методу, його обґрунтування на рівні маргінального векторного поля, узагальнену цільову функцію навчання, зв'язок із суміжними роботами, шість конкретних профілів швидкості, спосіб адаптації вже навчених моделей без перенавчання, аналіз похибки чисельного інтегрування та межі застосовності методу. Експериментальну перевірку викладено в розділі 4.

3.1. Побудова та формальне означення методу узгодження потоків із профілем швидкості

Лінійний інтерполянт $x_t = x_0 + t(x_1 - x_0)$, де x_0 – зразок базового розподілу p_0 (зазвичай стандартного нормального), а x_1 – зразок цільового розподілу даних p_1 , на якому ґрунтуються наявні моделі узгодження потоків, неявно закріплює сталу швидкість руху зразка від шуму до даних. Цей вибір зручний з обчислювального погляду, оскільки умовна швидкість зводиться до незалежного від часу зміщення $x_1 - x_0$, а траєкторії ймовір-

нісного потоку є прямолінійними й допускають інтерпретацію в термінах оптимального транспорту. Водночас він фіксує не лише форму траєкторії, а й темп її проходження, тоді як у базових роботах з узгодження потоків стала швидкість входить безпосередньо до означення методу й не розглядається як окремий проєктний параметр [12, 13]. Предметне зіставлення із суміжними роботами наведено в підрозділі 3.4.

Запропонований у цій роботі метод узгодження потоків із профілем швидкості (Velocity Scheduled Flow Matching, VSFM) [26] знімає це обмеження, замінюючи стале цільове поле часозалежним. Замість сталого зміщення $x_1 - x_0$ умовним цільовим полем стає $v(t)(x_1 - x_0)$, де $v(t)$ – часовий профіль швидкості, який задає темп накопичення зміщення в кожен момент часу. Метод розширює клас цільових векторних полів: при сталому профілі $v(t) \equiv 1$ відтворюється звичайне узгодження потоків, а інші допустимі профілі перерозподіляють той самий рух у часі. Архітектура мережі, функція втрат за структурою, спосіб спряження зразків і чисельний інтегратор при цьому залишаються незмінними.

Не кожна функція придатна на роль профілю швидкості: на неї накладено дві вимоги.

Означення 3.1. Допустимим профілем швидкості називаємо інтегровну невід’ємну функцію $v : [0, 1] \rightarrow \mathbb{R}_{\geq 0}$, що задовольняє умову одиничного інтеграла

$$\int_0^1 v(t) dt = 1. \quad (3.1)$$

Профілем руху, відповідним профілю швидкості v , називаємо функцію $\alpha(t) = \int_0^t v(s) ds$.

Невід’ємність виключає зворотний рух уздовж траєкторії, а умова одиничного інтеграла гарантує, що отриманий рух повністю транспортує x_0 у x_1 , тобто зберігає сумарне зміщення незалежно від форми профілю. Профіль

руху за побудовою є неспадним і задовольняє рівності $\alpha(0) = 0$ та $\alpha(1) = 1$, а в точках неперервності v – також $\alpha'(t) = v(t)$. Жодну з двох умов не можна послабити без втрати цих властивостей, що показано в підрозділі 3.8.

Узагальнене умовне векторне поле швидкостей набуває часозалежного вигляду

$$u_t(x \mid x_0, x_1) = v(t) (x_1 - x_0). \quad (3.2)$$

Таке поле є розширенням умовного векторного поля $u_t(x \mid x_0, x_1) = x_1 - x_0$, розглянутого в розділі 1, а не окремою конструкцією: множник $v(t)$ змінює лише величину швидкості в кожен момент, зберігаючи напрямок руху $x_1 - x_0$, заданий спряженням розподілів.

Положення зразка в момент часу t задається афінним інтерполянтом

$$x_t = x_0 + \alpha(t) (x_1 - x_0). \quad (3.3)$$

Зміщення $x_1 - x_0$ фіксується спряженням, профіль руху $\alpha(t)$ визначає частку зміщення, вже пройденому до моменту t , а профіль швидкості $v(t)$ керує темпом її накопичення.

Профіль швидкості задає метод повністю: з нього випливають і цільове векторне поле, і функція втрат, і схема чисельного інтегрування під час генерації. Це дає змогу подати метод у замкненій формі.

Означення 3.2. Нехай задано базовий розподіл p_0 , розподіл даних p_1 , спряження $\pi \in \Pi(p_0, p_1)$ цих розподілів, параметричне сімейство векторних полів $\{v_\theta(x, t)\}_{\theta \in \Theta}$ та допустимий профіль швидкості v із профілем руху α (означення 3.1). Методом узгодження потоків із профілем швидкості називаємо спосіб побудови генеративної моделі, заданий трійкою

$$\text{VSFM}(v) = \langle u^v, \mathcal{L}_{\text{VSFM}}, \Phi_{\alpha, N} \rangle, \quad (3.4)$$

компоненти якої визначено так:

1. умовне цільове векторне поле $u_t^v(x \mid x_0, x_1) = v(t)(x_1 - x_0)$, інтегральні криві якого задають положення зразка на траєкторії, $x_t = x_0 + \alpha(t)(x_1 - x_0)$, де $(x_0, x_1) \sim \pi$ і $t \sim \mathcal{U}(0, 1)$;
2. цільова функція навчання $\mathcal{L}_{\text{VSFM}}(\theta) = \mathbb{E}_{t, x_0, x_1} \|v_\theta(x_t, t) - (x_1 - x_0)\|^2$;
3. оператор генерації $\Phi_{\alpha, N}$ – N -кроковий дискретний потік $x_{i+1} = x_i + v_\theta(x_i, \tau_i) \Delta\tau_i$, $i = 0, \dots, N-1$, де $t_i = i/N$, $\Delta\tau_i = \tau_{i+1} - \tau_i$, $x_0 \sim p_0$ і $\Phi_{\alpha, N}(x_0) = x_N$. Часовим входом мережі τ_i є той час, у якому її навчено: $\tau_i = t_i$ для мережі, навченої під профілем за $\mathcal{L}_{\text{VSFM}}$, і $\tau_i = \alpha(t_i)$ для мережі, навченої зі звичайною функцією втрат узгодження потоків.

Допустимий профіль швидкості v є єдиним параметром означення 3.2: усі три компоненти трійки (3.4) виражено через нього та через його первісну α . Цільова функція при цьому регресує стале зміщення $x_1 - x_0$, а не саме поле u^v : обидві параметризації відновлюють одне й те саме поле, а вибір незваженої форми обґрунтовано в підрозділі 3.3. Оператор генерації охоплює обидва способи застосування методу – навчання з нуля під профілем і переналаштування вже навченої лінійної моделі, – які відрізняються лише часовим входом мережі, тоді як розміри кроків $\Delta\tau_i$ у них спільні. Алгоритмічну форму цільової функції та оператора генерації подано в алгоритмах 1 та 2.

3.2. Ймовірнісний шлях та маргінальне векторне поле

Співвідношення (3.2) задає умовне векторне поле швидкостей для фіксованої пари (x_0, x_1) . Поведінку методу як генеративної моделі визначає, однак, не воно, а маргінальне векторне поле швидкостей – усереднене за спряженням розподілів умовне сподівання миттєвої швидкості в поточній точці [12]. Нехай $(X_0, X_1) \sim \pi$ – спряження базового та цільового розподілів,

а

$$X_\tau^{\text{лін}} = X_0 + \tau (X_1 - X_0), \quad \tau \in [0, 1], \quad (3.5)$$

– процес звичайного узгодження потоків, де τ – час лінійного процесу, із граничним розподілом p_τ та маргінальним векторним полем

$$u_\tau(x) = \mathbb{E}_{(X_0, X_1) \sim \pi} [X_1 - X_0 \mid X_\tau^{\text{лін}} = x]. \quad (3.6)$$

Саме поле (3.6) апроксимує нейронна мережа $v_\theta(x, t)$, що параметризує векторне поле швидкостей (розділ 1), оскільки за результатом Ліпмана та співавторів умовна й маргінальна задачі регресії мають спільний мінімум [12].

Встановимо тепер, як ймовірнісний шлях і маргінальне поле під довільним допустимим профілем виражаються через відповідні об'єкти звичайного узгодження потоків. Під профілем швидкості інтерполянт (3.3) задає процес $X_t^v = X_0 + \alpha(t)(X_1 - X_0)$ із граничним розподілом p_t^v та маргінальним векторним полем, заданим аналогічно до (3.6) через умовне сподівання миттєвої швидкості $\frac{d}{dt}X_t^v = v(t)(X_1 - X_0)$,

$$\begin{aligned} u_t^v(x) &= \mathbb{E}_{(X_0, X_1) \sim \pi} \left[\frac{d}{dt}X_t^v \mid X_t^v = x \right] \\ &= \mathbb{E}_{(X_0, X_1) \sim \pi} [v(t)(X_1 - X_0) \mid X_t^v = x]. \end{aligned} \quad (3.7)$$

Твердження 3.1. *Нехай v – допустимий профіль швидкості з профілем руху α , $(X_0, X_1) \sim \pi$, а $X_t^v = X_0 + \alpha(t)(X_1 - X_0)$ – процес під цим профілем. Тоді для всіх $t \in [0, 1]$:*

1. *виконується тотожність $X_t^v = X_{\alpha(t)}^{\text{лін}}$ як рівність випадкових величин, звідки граничний розподіл під профілем збігається з репараметризованим у часі шляхом звичайного узгодження потоків,*

$$p_t^v = p_{\alpha(t)}; \quad (3.8)$$

2. *маргінальне векторне поле під профілем задовольняє*

$$u_t^v(x) = v(t) u_{\alpha(t)}(x); \quad (3.9)$$

3. *крайові розподіли зберігаються: $p_0^v = p_0$ та $p_1^v = p_1$.*

Доведення. Підстановка $\tau = \alpha(t)$ у (3.5) дає тотожність $X_t^v = X_{\alpha(t)}^{\text{лін}}$. З невід’ємності профілю впливає монотонне неспадання α , а з умови одиничного інтеграла – рівність $\alpha(1) = 1$, тож $\alpha(t) \in [0, 1]$ є допустимим значенням часу лінійного процесу (3.5), і рівність розподілів (3.8) доведено.

Рівність (3.9) впливає з (3.7): детермінований множник $v(t)$ виноситься з-під умовного сподівання, а подія $\{X_t^v = x\}$ збігається з $\{X_{\alpha(t)}^{\text{лін}} = x\}$ за доведеною тотожністю, звідки $u_t^v(x) = v(t) \mathbb{E}_{(X_0, X_1) \sim \pi} [X_1 - X_0 \mid X_{\alpha(t)}^{\text{лін}} = x] = v(t) u_{\alpha(t)}(x)$.

Пункт 3 впливає з пункту 1 та рівностей $\alpha(0) = 0$, $\alpha(1) = 1$ означення 3.1: $p_0^v = p_{\alpha(0)} = p_0$ і $p_1^v = p_{\alpha(1)} = p_1$. $\square$

Наслідок 3.1. *При сталому профілі $v \equiv 1$ маємо $\alpha(t) = t$, і всі три компоненти означення 3.2 збігаються з відповідними компонентами звичайного узгодження потоків: u^v – стале умовне поле $x_1 - x_0$ з лінійним інтерполянтом $x_0 + t(x_1 - x_0)$, $\mathcal{L}_{\text{VSFM}}$ – стандартна функція втрат умовного узгодження потоків, а $\Phi_{\alpha, N}$ – метод Ейлера з рівномірним кроком $\Delta\tau_i = 1/N$. Зокрема, $p_t^v = p_t$ і $u_t^v = u_t$.*

Процес $X_t^v = X_{\alpha(t)}^{\text{лін}}$ є тим самим стохастичним інтерполянтом [14] узгодження потоків, пройденим за змінним темпом $t \mapsto \alpha(t)$. Тому відповідність між маргінальним полем і ймовірнісним шляхом успадковується від базової пари: маргінальне поле u_t^v , означене як умовне сподівання миттєвої швидкості процесу, породжує його граничний розподіл p_t^v за тим самим результатом теорії узгодження потоків [12], що й поле u_τ породжує шлях p_τ . Множник $v(t) = \alpha'(t)$ у полі (3.9) з’являється саме як темп заміни часу.

Співвідношення (3.8)–(3.9) формалізують профіль швидкості як монотонну заміну часу $t \mapsto \alpha(t)$ над незмінним ймовірнісним шляхом, що перерозподіляє лише темп його проходження. Звідси впливають обидва застосування методу. Під час навчання мережа апроксимує те саме

поле u_τ , тож профіль входить виключно через розподіл навчальних позицій (підрозділ 3.3). На етапі інференсу поле u_t^v відновлюється множенням $v(t) v_\theta(x, \alpha(t))$, що зводить інтегрування до рівняння навченої моделі у τ -просторі (підрозділ 3.6).

3.3. Цільова функція навчання

Цільовим полем методу є узагальнене умовне поле $u_t^v(x \mid x_0, x_1) = v(t)(x_1 - x_0)$ з (3.2) (пункт 1 означення 3.2). Його регресію можна параметризувати двома способами, що відрізняються величиною цілі та способом використання мережі на етапі інференсу.

За прямої параметризації мережа v_θ навчається відтворювати повне профілезалежне поле:

$$\mathcal{L}_{\text{VSFM}}^{\text{пов}}(\theta) = \mathbb{E}_{t, x_0, x_1} \|v_\theta(x_t, t) - v(t)(x_1 - x_0)\|^2, \quad (3.10)$$

$$x_t = x_0 + \alpha(t)(x_1 - x_0),$$

де $t \sim \mathcal{U}(0, 1)$, а пара (x_0, x_1) береться зі спряження розподілів. Навчена в такий спосіб мережа апроксимує маргінальне поле u_t^v безпосередньо, тож генерація зводиться до інтегрування рівняння $dx/dt = v_\theta(x, t)$ від $t = 0$ до $t = 1$ стандартним методом Ейлера на рівномірній часовій сітці.

За альтернативної параметризації мережа навчається передбачати стале зміщення $x_1 - x_0$:

$$\mathcal{L}_{\text{VSFM}}(\theta) = \mathbb{E}_{t, x_0, x_1} \|v_\theta(x_t, t) - (x_1 - x_0)\|^2, \quad (3.11)$$

$$x_t = x_0 + \alpha(t)(x_1 - x_0).$$

За сталого профілю $\alpha(t) \equiv t$ цей вираз збігається зі стандартною функцією втрат умовного узгодження потоків. Згідно з підрозділом 3.2, мінімізатором цієї регресії є умовне сподівання зміщення в поточній позиції, $v_\theta^*(x, t) = u_{\alpha(t)}(x)$: мережа апроксимує те саме поле u_τ із (3.6), спільне

для всіх допустимих профілів, але обчислене в репараметризованому часі $\tau = \alpha(t)$, тож навчені за різних профілів функції відрізняються лише заміною часу в другому аргументі. Профілезалежне поле відновлюється лише на етапі інференсу множенням на $v(t)$: генерація інтегрує рівняння $dx/dt = v(t) v_\theta(x, t)$ згідно з (3.9), що детально розглянуто в підрозділі 3.6.

Обидві форми дають змогу відновити те саме векторне поле швидкостей: підстановка $v_\theta \mapsto v(t) v_\theta$ у (3.10) дає $\mathbb{E}_{t, x_0, x_1} [v(t)^2 \|v_\theta - (x_1 - x_0)\|^2]$, тобто пряма форма (3.10) еквівалентна регресії зміщення, зваженій множителем $v(t)^2$, а вибрана форма (3.11) відповідає тій самій регресії без ваги. Форму (3.11) обрано тому, що ціль сталої величини $\|x_1 - x_0\|$ не залежить від профілю, тоді як ціль прямої форми $v(t)(x_1 - x_0)$ зростає поблизу $t = 0$ і прямує до нуля поблизу $t = 1$ для гальмівних профілів. Вага $v(t)^2$ при цьому змінюється від $v(0)^2$, що сягає дев'яти для розглянутих далі профілів, до нуля, тобто ефективний темп навчання стає сильно нерівномірним за часом; у наших експериментах така вага дестабілізувала оновлення градієнтів, тому подальший виклад спирається на форму (3.11).

Профіль швидкості входить у навчання через розподіл позицій x_t , а не через величину цілі регресії. За рівномірного розподілу $t \sim \mathcal{U}(0, 1)$ пройдена частка $\tau = \alpha(t)$ розподілена нерівномірно: оскільки α строго зростає там, де $v > 0$, функція розподілу випадкової величини τ дорівнює α^{-1} , а її густина –

$$\rho(\tau) = \frac{1}{v(\alpha^{-1}(\tau))}. \quad (3.12)$$

Густина (3.12) обернена до швидкості, тож навчальні приклади розріджуються там, де профіль рухається найшвидше, і згущуються там, де він рухається найповільніше. Для гальмівних профілів це зосереджує більшу частину навчальної маси в позиціях, близьких до розподілу даних, де мережа уточнює дрібну структуру зразка: у чверть $\tau \in [0.75, 1]$ потрапляє близько 50% навчальних прикладів за рівномірного сповільнення і близько

63% за плавного приземлення проти 25% за лінійного профілю. Ціна такого перерозподілу – розрідження навчальних позицій на протилежному кінці відрізка, що розглянуто в підрозділі 3.8. Повну процедуру навчання подано в алгоритмі 1.

Algorithm 1 Навчання VSFM

Require: мережа v_θ ; профіль руху α ; кількість кроків K

- 1: **for** $k = 1, 2, \dots, K$ **do**
 - 2: Вибрати $t \sim \mathcal{U}(0, 1)$, $(x_0, x_1) \sim \pi$
 - 3: $x_t \leftarrow x_0 + \alpha(t) (x_1 - x_0)$ ▷ інтерполянт через профіль руху
 - 4: $\mathcal{L} \leftarrow \|v_\theta(x_t, t) - (x_1 - x_0)\|^2$ ▷ рівняння (3.11)
 - 5: Оновити θ градієнтним спуском за $\mathcal{L}$
 - 6: **end for**
-

3.4. Зв'язок із суміжними роботами

За критерієм оптимальності профіль швидкості відрізняється від раніше досліджених характеристик інтерполянта. Шаул та співавтори встановили, що серед афінних умовних побудов стала швидкість майже оптимальна за критерієм кінетичної енергії $\int_0^1 \mathbb{E}_{x \sim p_t} \|u_t(x)\|^2 dt$, тобто мінімізує інтегральну квадратичну швидкість руху [150]. Профіль швидкості оптимізує іншу величину – локальну похибку усічення методу Ейлера за фіксованої кількості кроків інтегрування, що залежить від розподілу похибки дискретизації вздовж траєкторії, а не від інтегральної швидкості. Обидва результати сумісні: стала швидкість мінімізує кінетичну енергію, тоді як неконстантний профіль знижує похибку Ейлера за обмеженого обчислювального бюджету.

Профіль швидкості також відрізняється від характеристик, якими оперують методи дифузійного моделювання. Розклади шуму [7, 151] та схеми зважування функції втрат [9, 11] у дифузійних моделях змінюють інтенсивність зашумлення на кожному кроці або коефіцієнти перед доданка-

ми втрат, проте залишають цільове векторне поле незмінним. Профіль швидкості натомість діє безпосередньо на векторне поле, роблячи його часозалежним, і тим самим перерозподіляє рух зразка в часі.

Окрему групу становлять методи, що змінюють розподіл часових моментів під час навчання або сітку дискретизації на етапі інференсу, не торкаючись лінійного інтерполянта. До них належать зміщення розподілу часу до середини відрізка $[0, 1]$ через логіт-нормальне семплювання [15] та побудова нерівномірних сіток інтегрування для дифузійних рівнянь [11]. Процедура переналаштування на етапі інференсу, описана в підрозділі 3.6, операційно належить до цієї групи, проте параметризує нерівномірну сітку через профіль швидкості й супроводжується аналізом похибки дискретизації, що виокремлює гальмівні профілі без перебору варіантів.

Конкретні форми профілів швидкості запозичено з поліноміальних профілів руху, відомих із задач планування руху та робототехніки, де профіль задається крайовими умовами на швидкість і прискорення на кінцях часового відрізка [152, 153]. У межах цього розділу такі поліноми використано лише як готові форми профілів швидкості; жодних інших припущень із зазначених задач не залучено.

Підсумовуючи, розглянуті підходи змінюють розклад зашумлення, ваги функції втрат, розподіл навчальних часових моментів або сітку дискретизації, залишаючи цільове векторне поле незмінним. Запропоноване в цій роботі узагальнення натомість робить часозалежним саме цільове векторне поле, і в цьому полягає його відмінність від зазначених робіт.

3.5. Поліноміальні профілі швидкості

Метод конкретизовано шістьма поліноміальними профілями швидкості: базовим лінійним, трьома гальмівними (рівномірне сповільнення, накат і плавне приземлення) та двома симетричними (плавний вхід-вихід і міні-

мальний ривок). Кожен профіль отримано накладанням крайових умов на профіль руху α та його похідні на кінцях відрізка $[0, 1]$; ці умови однозначно визначають поліном мінімального степеня. Вимога занулення більшого числа похідних α на кінці відрізка підвищує гладкість профілю в цій точці ціною концентрації руху поодаль від неї. Усі шість профілів зведено в таблицю 3.1, а їхні графіки наведено на рисунку 3.1: у верхній частині рисунка зображено профілі швидкості $v(t)$, у нижній – відповідні профілі руху $\alpha(t) = \int_0^t v(s) ds$.

Профілі поділяються на два підсімейства за способом розподілу руху вздовж траєкторії. Гальмівне підсімейство характеризується монотонно спадаючою швидкістю: зразок рухається найшвидше поблизу $t = 0$ і сповільнюється до зупинки при $t = 1$. Симетричне підсімейство симетричне відносно $t = 1/2$ і має нульову швидкість на обох кінцях.

Таблиця 3.1

Шість поліноміальних профілів швидкості; кожен задовольняє

$$\alpha(0) = 0, \alpha(1) = 1 \text{ та } \int_0^1 v(t) dt = 1; \text{ тут } v_{\max} = \max_{t \in [0,1]} v(t).$$

Профіль	$\alpha(t)$	$v(t)$	$v(0)$	$v(1)$	$v_{\max}$
<i>Базовий</i>					
лінійний	t	1	1	1	1
<i>Гальмівні (монотонно спадаюча швидкість)</i>					
рівномірне сповільнення	$2t - t^2$	$2 - 2t$	2	0	2
накат	$(3t - t^3)/2$	$\frac{3}{2}(1 - t^2)$	1.5	0	1.5
плавне приземлення	$3t - 3t^2 + t^3$	$3(1 - t)^2$	3	0	3
<i>Симетричні</i>					
плавний вхід-вихід	$3t^2 - 2t^3$	$6t(1 - t)$	0	0	1.5
мінімальний ривок	$10t^3 - 15t^4 + 6t^5$	$30t^2(1 - t)^2$	0	0	1.875

Лінійний профіль (стала швидкість $v(t) = 1$, лінійний профіль руху $\alpha(t) = t$) відтворює стандартний лінійний інтерполянт узгодження потоків і слугує базовим варіантом для порівняння.

Рівномірне сповільнення задається профілем руху $\alpha(t) = 2t - t^2 = 1 - (1 - t)^2$ – єдиним квадратичним поліномом, що задовольняє умови $\alpha(0) = 0, \alpha(1) = 1$ та $\alpha'(1) = 0$. Швидкість $v(t) = 2 - 2t$ спадає зі сталою швидкістю $\alpha'' = -2$ і досягає нуля при $t = 1$, причому до моменту $t = 1/2$ пройдено $3/4$ повного зміщення. Обернена функція $\alpha^{-1}(\tau) = 1 - \sqrt{1 - \tau}$ доступна у замкненій формі.

Накат задається профілем руху $\alpha(t) = (3t - t^3)/2$ і швидкістю $v(t) =$

$\frac{3}{2}(1 - t^2)$. Додаткова умова $\alpha''(0) = 0$ поряд із крайовими умовами дає помірно початкове сповільнення, що посилюється до $t = 1$. Пікова швидкість дорівнює $3/2$, а до моменту $t = 1/2$ пройдено $\alpha(1/2) \approx 0.69$ повного зміщення.

Плавне приземлення задається профілем руху $\alpha(t) = 3t - 3t^2 + t^3 = 1 - (1 - t)^3$ і швидкістю $v(t) = 3(1 - t)^2$. Це єдиний кубічний поліном, для якого $\alpha'(1) = \alpha''(1) = 0$, тобто на кінці $t = 1$ зануляються і швидкість, і прискорення. Початкова швидкість $v(0) = 3$ є найбільшою серед шести профілів, а до моменту $t = 1/2$ пройдено $\alpha(1/2) = 7/8$ зміщення.

Плавний вхід-вихід задається профілем руху $\alpha(t) = 3t^2 - 2t^3$ і швидкістю $v(t) = 6t(1 - t)$. Це кубічний поліном Ерміта, єдиний кубічний поліном з умовами $\alpha'(0) = \alpha'(1) = 0$. Дзвоноподібна швидкість досягає максимуму $v_{\max} = 3/2$ при $t = 1/2$, і за симетрією $\alpha(1/2) = 1/2$.

Мінімальний ривок задається профілем руху $\alpha(t) = 10t^3 - 15t^4 + 6t^5$ і швидкістю $v(t) = 30t^2(1 - t)^2$. Це єдиний поліном п'ятого степеня з умовами $\alpha'(0) = \alpha'(1) = \alpha''(0) = \alpha''(1) = 0$. Цей самий поліном мінімізує інтеграл $\int_0^1 |\alpha'''(t)|^2 dt$ за вказаних крайових умов [152]; він має гладкість C^2 на обох кінцях і досягає максимуму $v_{\max} = 1.875$.

Гальмівне підсімейство перевіряє, чи є сповільнення в напрямку $t = 1$ корисним як клас, а симетричне підсімейство виконує роль контролю: схожа гладкість профілів за відсутності напрямленого зміщення до розподілу даних. Кількісне обґрунтування того, чому цей напрямок має значення для генерації від шуму до даних, наведено в підрозділі 3.7.

3.6. Переналаштування розкладу на етапі інференсу

Будь-яку модель узгодження потоків, навчену на лінійному інтерполянті, можна семплювати під довільним допустимим профілем швидкості без перенавчання.

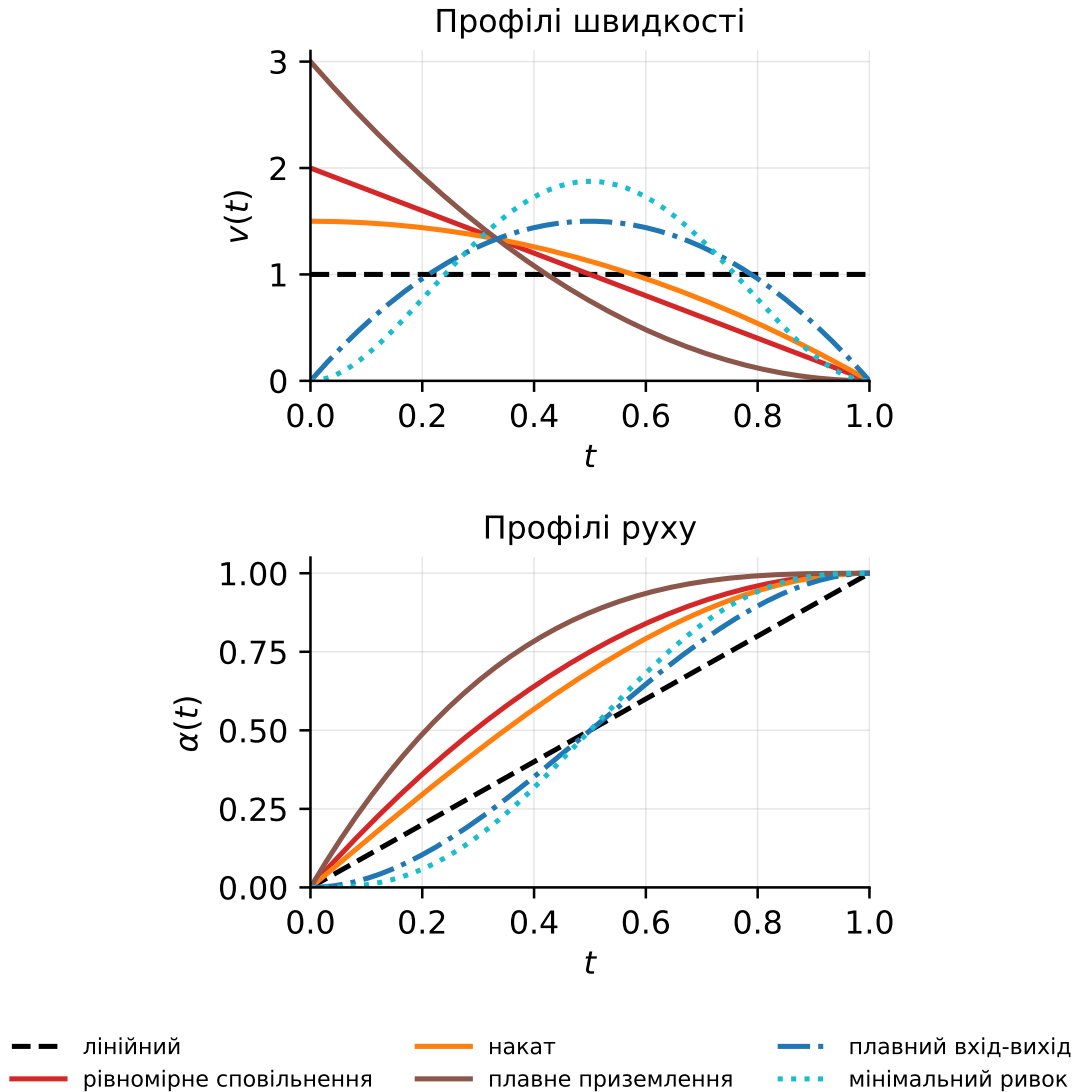


Рис. 3.1. Профілі швидкості та відповідні профілі руху

Нехай v_θ – мережа, навчена на лінійному інтерполянті. За результатом про спільний мінімум умовної та маргінальної задач регресії [12] вона апроксимує маргінальне поле (3.6),

$$v_\theta(x, \tau) \approx u_\tau(x), \quad \tau \in [0, 1], \quad (3.13)$$

причому наближення справджується для пар аргументів (x, τ) , у яких позиція x типова для граничного розподілу p_τ . Під профілем швидкості v зразок у момент t перебуває в позиції $x_t = x_0 + \alpha(t)(x_1 - x_0)$, розподіл якої за (3.8) збігається з $p_{\alpha(t)}$, а не з p_t . Звертання до мережі у вигляді $v_\theta(x, t)$ вивело б її

аргументи за межі навчального розподілу.

Щоб уникнути цього, визначимо адаптоване векторне поле

$$\tilde{v}(x, t) = v(t) v_{\theta}(x, \alpha(t)) \quad (3.14)$$

та інтегруємо рівняння $dx/dt = \tilde{v}(x, t)$. Підстановка наближення (3.13) дає $\tilde{v}(x, t) \approx v(t) u_{\alpha(t)}(x) = u_t^v(x)$ за рівністю (3.9): адаптоване поле відтворює маргінальне поле під профілем з точністю до залишкової похибки навченої мережі. Геометрично це відповідає тому, що лінійна траєкторія $x_{\tau}^{\text{лін}} = x_0 + \tau(x_1 - x_0)$ і траєкторія під профілем $x_t = x_0 + \alpha(t)(x_1 - x_0)$ проходять той самий прямолінійний відрізок і відрізняються лише швидкістю: прирівнювання $x_t = x_{\tau}^{\text{лін}}$ дає $\tau = \alpha(t)$, тож просторова точка, відвідувана в момент t під профілем, збігається з точкою, яку лінійна траєкторія відвідує в момент $\alpha(t)$, і аргументи мережі в (3.14) спільно належать її навчальному розподілу.

Заміна змінної $\tau = \alpha(t)$ з диференціалом $d\tau = v(t) dt$ перетворює адаптоване рівняння $dx/dt = v(t) v_{\theta}(x, \alpha(t))$ на звичайне диференціальне рівняння у τ -просторі

$$\frac{dx}{d\tau} = v_{\theta}(x, \tau). \quad (3.15)$$

Це власне рівняння навченої моделі, параметризоване змінною τ .

Рівняння (3.15) подає $v_{\theta}(x, \tau)$ як предиктор зміщення: мережа видає умовне сподівання повного зміщення $x_1 - x_0$ за поточною позицією, а не миттєву швидкість за часом t . Із цього подання випливають як процедура адаптації, так і функція втрат (3.11). На етапі інференсу предиктор обчислюється в точці $\tau = \alpha(t)$ і просувається на крок $\Delta\tau_i$, застосовуючи відповідну частку передбаченого зміщення; під час навчання предиктор регресують безпосередньо на $x_1 - x_0$. Профіль тому входить лише через позиції під час навчання і через розміри кроків на етапі інференсу.

Нехай $t_i = i/N$, $i = 0, 1, \dots, N$ – рівномірна сітка з N кроків за часом t із кроком $\Delta t = 1/N$. Оскільки профіль руху α для кожного з розглянутих

поліноміальних профілів доступний у замкненій формі, частку зміщення, накопичену від t_i до t_{i+1} , можна обчислити точно, без квадратурного наближення:

$$\Delta\tau_i = \alpha(t_{i+1}) - \alpha(t_i) = \int_{t_i}^{t_{i+1}} v(t) dt. \quad (3.16)$$

Оскільки мережа v_θ передбачає повне зміщення $x_1 - x_0$, множення на $\Delta\tau_i$ дає координатний приріст для відповідного інтервалу. Повне оновлення на етапі інференсу має вигляд

$$x_{i+1} = x_i + v_\theta(x_i, \tau_i) \Delta\tau_i, \quad \tau_i = \alpha\left(\frac{i}{N}\right), \quad \Delta\tau_i = \alpha\left(\frac{i+1}{N}\right) - \alpha\left(\frac{i}{N}\right). \quad (3.17)$$

Повну процедуру подано в алгоритмі 2.

Algorithm 2 Інференс VSFM (точний метод Ейлера у τ -просторі)

Require: навчена мережа v_θ ; профіль руху α ; кількість кроків N

- 1: Вибрати $x_0 \sim \mathcal{N}(0, I)$
 - 2: **for** $i = 0, 1, \dots, N - 1$ **do**
 - 3: $\tau_i \leftarrow \alpha(i/N)$; $\Delta\tau_i \leftarrow \alpha((i+1)/N) - \alpha(i/N)$
 - 4: $x_{i+1} \leftarrow x_i + v_\theta(x_i, \tau_i) \cdot \Delta\tau_i$
 - 5: **end for**
 - 6: **return** x_N
-

Алгоритм 2 сформульовано для мережі, навченої на лінійному інтерполянті, для якої часовим входом є $\tau_i = \alpha(t_i)$. Для мережі, навченої з нуля за алгоритмом 1, часовим входом служить t_i , оскільки саме його мережа отримувала під час навчання; розмір кроку $\Delta\tau_i$ при цьому лишається тим самим, що відповідає інтегруванню рівняння $dx/dt = v(t) v_\theta(x, t)$ з підрозділу 3.3.

Оновлення (3.17) інтегрує рівняння навченої моделі у τ -просторі на нерівномірній сітці $\tau_i = \alpha(i/N)$, що операційно збігається із запуском стандартного семплера Ейлера на цій сітці. Розмір кроку $\Delta\tau_i \approx v(t_i) \Delta t$ змінюється вздовж траєкторії, і його розподіл за кожним профілем наведено в

таблиці 3.2. Гальмівні профілі беруть великі кроки τ поблизу $t = 0$, де поле порівняно гладке, і малі кроки поблизу $t = 1$, де навчене поле, як обговорено в підрозділі 3.7, зазвичай змінюється найшвидше; симетричні профілі натомість концентрують найбільші кроки поблизу середини траєкторії.

Таблиця 3.2

Наближений розмір кроку $\Delta\tau_i \approx v(t_i) \Delta t$ (в одиницях Δt) для кожного профілю на рівномірній часовій сітці.

Профіль	поблизу $t = 0$	поблизу $t = 1/2$	поблизу $t = 1$
лінійний	≈ 1	≈ 1	≈ 1
рівномірне сповільнення	≈ 2	≈ 1	≈ 0
накат	≈ 1.5	≈ 1.125	≈ 0
плавне приземлення	≈ 3	≈ 0.75	≈ 0
плавний вхід-вихід	≈ 0	≈ 1.5	≈ 0
мінімальний ривок	≈ 0	≈ 1.875	≈ 0

3.7. Аналіз похибки дискретизації

Профіль швидкості впливає на якість генерації через локальну похибку усічення методу Ейлера, застосованого до рівняння у τ -просторі $dx/d\tau = v_\theta(x, \tau)$ з нерівномірними кроками $\Delta\tau_i = \alpha(t_{i+1}) - \alpha(t_i)$. Нижче розглянуто всі шість профілів, починаючи з лінійного базового та рівномірного сповільнення.

Для рівняння $dx/d\tau = g(x, \tau)$, інтегрованого явною схемою Ейлера $x_{i+1} = x_i + g(x_i, \tau_i) \Delta\tau_i$, локальна похибка усічення на кроці i за стандартним результатом, отриманим розкладом точного розв'язку в ряд Тейлора

навколо τ_i [154], дорівнює

$$e_i = \frac{(\Delta\tau_i)^2}{2} \left. \frac{d^2x}{d\tau^2} \right|_{\tau=\tau_i} + \mathcal{O}((\Delta\tau_i)^3). \quad (3.18)$$

За $g(x, \tau) = v_\theta(x, \tau)$ друга похідна є повною похідною поля вздовж траєкторії, тож за правилом диференціювання складеної функції зводиться до варіації поля

$$\frac{d^2x}{d\tau^2} = \frac{\partial v_\theta}{\partial x} v_\theta + \frac{\partial v_\theta}{\partial \tau}. \quad (3.19)$$

Доданок із кривиною профілю тут відсутній, оскільки рівняння у τ -просторі не містить явного множника $v(t)$; єдиним джерелом похибки усічення залишається нелінійність навченого поля v_θ . На рівномірній часовій сітці розмір кроку задовольняє $\Delta\tau_i \approx v(t_i) \Delta t$, звідки $(\Delta\tau_i)^2 \approx v(t_i)^2 (\Delta t)^2$, тож локальна похибка масштабується як $v(t_i)^2$, помножене на варіацію поля в точці τ_i .

Локальні похибки усічення окремих кроків накопичуються у глобальну похибку згенерованого зразка. За стандартним результатом для однокрокових методів інтегрування глобальна похибка в кінцевій точці обмежена сумою локальних похибок (3.18) з точністю до множника, що залежить від константи Ліпшиця поля v_θ і не залежить від профілю [154]. Тому профіль, який зменшує локальну похибку на ділянках найшвидшої зміни поля, зменшує й накопичену похибку кінцевого зразка за фіксованого бюджету кроків.

Для лінійного профілю $v(t) \equiv 1$ крок $\Delta\tau_i = \Delta t$ сталий і $(\Delta\tau_i)^2 = (\Delta t)^2$. Доданок варіації поля у рівнянні (3.19) не пригнічується на жодній ділянці відрізка $[0, 1]$. Поблизу $\tau = 1$ траєкторія остаточно прив'язується до конкретного зразка даних, і аналіз кривини траєкторій ймовірного потоку в дифузійних моделях показує, що саме на цій ділянці навчені поля зазвичай змінюються найшвидше [11]; для узгодження потоків це припущення перевіряється експериментально в розділі 4. За лінійного профілю похибка усічення на цій ділянці зберігає повну величину.

Для рівномірного сповільнення $v(t) = 2 - 2t$ крок $\Delta\tau_i \approx 2(1 - t_i) \Delta t$ зменшується до нуля при $t \rightarrow 1$, тож $(\Delta\tau_i)^2 \approx 4(1 - t_i)^2 (\Delta t)^2 \rightarrow 0$. Похибка варіації поля тому пригнічується саме там, де навчене поле змінюється найбільше, а відсутність доданка з кривиною профілю означає, що компенсувального доданка похибки не виникає.

Для нахату $v(t) = \frac{3}{2}(1 - t^2)$ початкове значення $v(0) = 3/2$ дає $(\Delta\tau_i)^2 \approx (9/4)(\Delta t)^2$, що м'якше за $4(\Delta t)^2$ рівномірного сповільнення. Поблизу $t = 1$ розклад $(1 - t^2) = (1 - t)(1 + t) \rightarrow 2(1 - t)$ дає $(\Delta\tau_i)^2 \approx 9(1 - t_i)^2 (\Delta t)^2 \rightarrow 0$ – те саме пригнічення поблизу розподілу даних, що й у рівномірного сповільнення, але з меншим початковим кроком.

Для плавного приземлення $v(t) = 3(1 - t)^2$ початкове значення $v(0) = 3$ дає $(\Delta\tau_i)^2 \approx 9(\Delta t)^2$ – найбільший початковий квадрат кроку серед шести профілів. Поблизу $t = 1$ маємо $(\Delta\tau_i)^2 \approx 9(1 - t_i)^4 (\Delta t)^2 \rightarrow 0$, тобто найшвидше пригнічення серед гальмівних профілів. Точність поблизу даних дістається ціною великого початкового кроку, що вносить суттєву похибку усічення на ранній ділянці.

Для плавного входу-виходу $v(t) = 6t(1 - t)$ похибка пригнічується на обох кінцях, оскільки $v \rightarrow 0$ лінійно при $t \rightarrow 0$ і $t \rightarrow 1$, тоді як у середині $v(1/2) = 3/2$ дає $(\Delta\tau_i)^2 \approx (9/4)(\Delta t)^2$. Найбільші кроки припадають на середину траєкторії, де нелінійність поля нижча, ніж поблизу $\tau = 1$, тож для генерації від шуму до даних профіль дає рівномірне, але незначне покращення відносно базового.

Для мінімального ривка $v(t) = 30t^2(1 - t)^2$ швидкість занулюється квадратично на обох кінцях – швидше, ніж у плавного входу-виходу, – тоді як у середині $v(1/2) \approx 1.875$ дає найбільший серединний крок серед шести профілів, $(\Delta\tau_i)^2 \approx 3.52(\Delta t)^2$. Загальна картина повторює плавний вхід-вихід, лише виразніше в кожній точці, а за дуже малої кількості кроків серединний крок домінує в балансі похибки.

Профіль швидкості, що розміщує малі значення $v(t_i)^2$, а отже й малі

$(\Delta\tau_i)^2$, там, де навчене поле v_θ змінюється найшвидше, знижує локальну похибку усічення Ейлера відносно лінійного базового. Серед шести профілів таку структуру відносно $\tau = 1$ має гальмівне підсімейство, тоді як симетричне – ні. Звідси випливає прогноз: гальмівні профілі покращують результат відносно базового з найбільшим виграшем за малої кількості кроків, де домінує похибка усічення, і зі спадним виграшем за великої кількості кроків, де вона несуттєва. Експерименти розділу 4 підтверджують цю закономірність на CIFAR-10 – як за навчання з нуля, так і за адаптації переднавченої моделі, – де гальмівні профілі дають найбільший виграш саме за малої кількості кроків. На інших модальностях, Stable Diffusion 3.5 та синтезі ЕКГ, перевагу натомість мають симетричні профілі, проте сам метод переналаштування розкладу й там покращує якість відносно лінійного базового. Який саме профіль є оптимальним, залежить від модальності даних – від того, де поле v_θ змінюється найшвидше, – тоді як користь від переналаштування розкладу зберігається.

3.8. Умови застосовності та межі методу

Означення 3.1 та 3.2 задають метод, а твердження 3.1 його коректність. Нижче встановлено межі його застосовності: наслідки послаблення умов допустимості профілю, характер впливу профілю на результат і умови, за яких цей вплив зникає.

Обидві умови означення 3.1 відповідають за різні властивості побудови, тож і порушення кожної з них має власні наслідки.

Твердження 3.2. Нехай $v : [0, 1] \rightarrow \mathbb{R}$ – інтегровна функція, $\alpha(t) = \int_0^t v(s) ds$, а $X_t^v = X_0 + \alpha(t)(X_1 - X_0)$.

1. Якщо $\int_0^1 v(t) dt \neq 1$, то $\alpha(1) \neq 1$ і кінцевий розподіл процесу дорівнює $p_{\alpha(1)} \neq p_1$: рух не транспортує базовий розподіл у розподіл даних.
2. Якщо v набуває від’ємних значень на множині додатної міри, то α

спадає на деякій ділянці й заміна часу $t \mapsto \alpha(t)$ перестає бути бієкцією $[0, 1] \rightarrow [0, 1]$: окремі значення τ відвідуються двічі, тож частину бюджету кроків витрачено на вже пройдені ділянки траєкторії; крім того, за $\alpha(t) \notin [0, 1]$ часовий аргумент мережі виходить за межі відрізка, на якому її навчено.

Доведення. За пунктом 1 твердження 3.1 кінцевий розподіл процесу дорівнює $p_{\alpha(1)}$, а $\alpha(1) = \int_0^1 v(t) dt$ за побудовою; у загальному випадку рівність $p_{\alpha(1)} = p_1$ виконується лише за $\alpha(1) = 1$, звідки й впливає перший пункт. Якщо ж $v < 0$ на множині додатної міри, то α спадає на відповідній ділянці, тож за теоремою про проміжне значення існують $t_1 \neq t_2$ з $\alpha(t_1) = \alpha(t_2)$, і α не є ін'єктивною. $\square$

У неперервному часі профіль швидкості взагалі не впливає на результат генерації – саме це й задає головну межу методу.

Твердження 3.3. Нехай поле $v_\theta(x, \tau)$ неперервне за τ і задовольняє умову Ліпшиця за x рівномірно за $\tau \in [0, 1]$. Тоді для будь-яких двох допустимих профілів v_1, v_2 із профілями руху α_1, α_2 точні розв'язки рівнянь $dx/dt = v_j(t) v_\theta(x, \alpha_j(t))$, $j = 1, 2$, зі спільною початковою умовою $x(0) = x_0$ збігаються при $t = 1$.

Доведення. Заміна змінної $\tau = \alpha_j(t)$ з диференціалом $d\tau = v_j(t) dt$ зводить обидва рівняння до одного й того самого рівняння (3.15) на відрізок $\tau \in [0, 1]$ з тією самою початковою умовою $x(0) = x_0$. За умовою Ліпшиця задача Коші для нього має єдиний розв'язок (теорема Пікара-Ліндельофа), тож $x(1)$ в обох випадках дорівнює значенню цього спільного розв'язку при $\tau = 1$. $\square$

Наслідок 3.2. Профіль швидкості не змінює відображення $x_0 \mapsto x_1$, яке метод реалізує в неперервному часі. Його вплив цілком зосереджено в похибці дискретизації (3.18) і прямує до нуля при $N \rightarrow \infty$.

Звідси впливає закономірність, установлена в підрозділі 3.7 і підтвер-

джена експериментально в розділі 4: виграш від профілю найбільший за малої кількості кроків і поступово зникає, коли кроків стає більше. Водночас це і є межа застосовності методу – за великого бюджету кроків він не дає переваги в принципі, а не через невдалий вибір профілю.

З установлених результатів впливає й решта обмежень методу.

По-перше, метод не розширює клас досяжних розподілів. За пунктом 1 твердження 3.1 множини ймовірнісних шляхів збігаються, $\{p_t^v : t \in [0, 1]\} = \{p_\tau : \tau \in [0, 1]\}$, а спряження π , крайові розподіли й прямолінійна геометрія траєкторій лишаються незмінними. Профіль перерозподіляє лише обчислювальний бюджет уздовж траєкторії й не компенсує похибку самого вивченого поля v_θ , яка й обмежує досяжну якість генерації.

По-друге, аналіз похибки підрозділу 3.7 виконано для явного методу Ейлера, що має перший порядок точності. Для однокрокової схеми порядку p локальна похибка усічення має порядок $\mathcal{O}((\Delta\tau_i)^{p+1})$ замість $\mathcal{O}((\Delta\tau_i)^2)$ у (3.18), тож абсолютна величина похибки, якою керує профіль, зменшується зі зростанням p , а з нею й відносний виграш від переналаштування розкладу.

По-третє, універсально оптимального профілю не існує. Мінімізація сумарної локальної похибки $\sum_i \frac{(\Delta\tau_i)^2}{2} \|d^2x/d\tau^2\|_{\tau=\tau_i}$ за профілем руху α потребує знання варіації вивченого поля вздовж траєкторії, яка залежить від модальності даних і від самої навченої моделі; замкненого розв’язку ця задача не має. Тому профіль лишається параметром, який належить налаштовувати під конкретну модальність, що й підтверджують розбіжні оптимальні профілі для CIFAR-10, Stable Diffusion 3.5 та синтезу ЕКГ у розділі 4.

По-четверте, при навчанні з нуля неконстантний профіль розподіляє навчальні позиції вздовж відрізка нерівномірно. За густиною (3.12) ділянки з великим v отримують пропорційно менше прикладів: для плавного приземлення ($v(0) = 3$) у чверть $\tau \in [0, 0.25)$ потрапляє близько 9% прикладів

проти 25% у лінійного профілю. За фіксованого бюджету кроків навчання це послаблює апроксимацію поля на розріджених ділянках, через що надто нерівномірні профілі з великим $v_{\max}$ втрачають перевагу.

Останнє обмеження стосується не профілю, а самої процедури переналаштування розкладу (підрозділ 3.6). Вона потребує моделі з неперервним часовим входом $\tau \in [0, 1]$ у параметризації передбачення зміщення або швидкості, для якої справджується наближення (3.13). Для дифузійних моделей із дискретним розкладом кроків або з параметризацією передбачення шуму потрібен попередній перехід до неперервної форми узгодження потоків.

3.9. Висновки до розділу 3

У третьому розділі дисертації розроблено метод узгодження потоків із профілем швидкості, що узагальнює стандартну побудову узгодження потоків заміною сталого цільового поля $x_1 - x_0$ часозалежним полем $v(t)(x_1 - x_0)$ для довільного допустимого профілю швидкості. Метод формалізовано двома означеннями: перше задає клас допустимих профілів швидкості та відповідні їм профілі руху, друге подає метод у замкненій формі – як трійку $\langle u^v, \mathcal{L}_{\text{VSFM}}, \Phi_{\alpha, N} \rangle$, що складається з цільового векторного поля, функції втрат та оператора генерації й має профіль своїм єдиним параметром. Запропоноване узагальнення розширює клас цільових векторних полів: при сталому профілі $v(t) \equiv 1$ воно відтворює звичайне узгодження потоків, тоді як інші профілі перерозподіляють той самий рух у часі, не змінюючи ні архітектури мережі, ні напрямку транспортування, ні характеру спряження розподілів. Це формально встановлено як збереження ймовірнісного шляху за монотонної заміни часу $t \mapsto \alpha(t)$, за якої граничні розподіли p_τ та маргінальне векторне поле репараметризуються в часі, а крайові розподіли лишаються незмінними.

Сформульовано узагальнену цільову функцію навчання, у якій профіль швидкості входить через розподіл навчальних позицій, а не через величину цілі регресії, та обґрунтовано вибір незваженої форми функції втрат як чисельно стійкішої за гальмівних профілів. На основі задач планування руху побудовано шість поліноміальних профілів швидкості – базовий лінійний, три гальмівні та два симетричні, – кожен із яких задається крайовими умовами на профіль руху та його похідні.

Показано, що метод допускає два застосування. Перше – переналаштування розкладу інтегрування на етапі інференсу: профіль швидкості застосовується до вже навченої лінійної моделі через інтегрування її рівняння у τ -просторі на нерівномірній сітці $\tau_i = \alpha(i/N)$ без перенавчання та без додаткових обчислень. Друге – навчання з нуля під неконстантним профілем, що зводиться до зміни розподілу навчальних позицій у функції втрат.

Аналіз локальної похибки усічення методу Ейлера виявив єдиний механізм, що пояснює обидва застосування: гальмівні профілі пригнічують доданок варіації поля $(\Delta\tau_i)^2 d^2x/d\tau^2$ поблизу $\tau = 1$, де навчене векторне поле змінюється найшвидше. Звідси випливає прогноз про перевагу гальмівних профілів із найбільшим виграшем за малої кількості кроків інтегрування. Експериментальну перевірку цього прогнозу на задачах генерації зображень і синтезу електрокардіограм наведено в розділі 4, де він підтверджується для зображень CIFAR-10, тоді як для інших модальностей оптимальним виявляється симетричний профіль, а сам метод переналаштування розкладу залишається корисним.

Формально встановлено й межі застосовності методу. Обидві умови означення допустимого профілю виявилися необхідними: послаблення будь-якої з них або позбавляє рух властивості транспортувати базовий розподіл у розподіл даних, або виводить часовий аргумент мережі за межі відрізка, на якому її навчено. Доведено також, що в неперервному часі профіль швидкості взагалі не змінює відображення, яке реалізує метод,

тож увесь його вплив зосереджено в похибці дискретизації та зникає зі зростанням кількості кроків інтегрування. Звідси ж випливають і подальші обмеження: метод не розширює клас досяжних розподілів, його виграш зменшується для схем інтегрування вищого порядку, а універсально оптимального профілю не існує.

Таким чином, профіль швидкості стає окремим елементом проєктного вибору при розробці генеративної моделі – незалежним від архітектури, напрямку транспортування та характеру спряження розподілів параметром, через який можна керувати якістю генерації й обчислювальною вартістю як на етапі навчання, так і на етапі інференсу. При цьому оптимальний профіль не є універсальним, а залежить від модальності даних і бюджету кроків інтегрування, тож його належить налаштовувати під конкретну задачу.

РОЗДІЛ 4

ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ВАЛІДАЦІЯ УДОСКОНАЛЕНОГО МЕТОДУ УЗГОДЖЕННЯ ПОТОКІВ

4.1. Набори даних та критерії оцінювання генеративних моделей

Метрики, описані нижче, конкретизують для окремих задач ті аспекти оцінювання, які розглянуто в підрозділі 1.6: відповідність згенерованих даних цільовому розподілу та заданій умові, а також прикладну користь синтезованих зразків. Обчислювальну вартість генерації в усіх експериментах виражено кількістю кроків інтегрування (NFE), тож якість моделей зіставляється за однакового обчислювального бюджету.

4.1.1. Набори даних

CIFAR-10. CIFAR-10 [155] – набір зображень, що став стандартним орієнтиром в оцінюванні генеративних і класифікаційних моделей. До нього входять 60 000 кольорових зображень розміром 32×32 пікселі, порівну розподілених між десятьма класами: літак, автомобіль, птах, кіт, олень, собака, жаба, кінь, корабель і вантажівка. Стандартне розбиття передбачає 50 000 навчальних і 10 000 тестових прикладів – по 6 000 зображень на клас.

Компактна роздільність і рівномірний розподіл класів роблять CIFAR-10 зручною базою для порівняння генеративних підходів: малий розмір зображень знижує обчислювальні вимоги, тоді як різноманітність класів перешкоджає зведенню задачі до запам'ятовування одноманітних текстур.

DrawBench. DrawBench – бенчмарк для оцінювання моделей генерації зображень за текстовими запитами (text-to-image), запропонований разом із моделлю Imagen [156]. Набір містить 200 текстових запитів (промптів),

розподілених за 11 категоріями складності: композиційні сцени, кількісні відношення між об'єктами, просторові та причинно-наслідкові зв'язки, написи у зображенні, рідкісні та маловживані слова, динамічні сцени, а також навмисно неправдоподібні комбінації.

PartiPrompts. PartiPrompts (Parti) – бенчмарк для оцінювання моделей генерації зображень за текстовими запитами, запропонований разом із авторегресивною моделлю Parti [157]. Набір містить понад 1600 текстових запитів англійською мовою, що охоплюють широкий спектр категорій і рівнів складності: від простих описів окремих об'єктів до складних композиційних сцен із взаємодіями, абстрактними поняттями та нетривіальними просторовими відношеннями. На відміну від DrawBench, що акцентує на крайніх випадках і навмисно складних запитах, PartiPrompts охоплює типові сценарії використання та дає змогу комплексно оцінити якість і різноманітність генерації.

РТВ-XL. РТВ-XL [158] – одна з найбільших публічно доступних колекцій клінічних записів електрокардіограм (ЕКГ). Колекція охоплює 21 799 десятисекундних 12-канальних записів від 18 869 пацієнтів. Демографічний склад збалансований: 52% чоловіків і 48% жінок; вік варіюється від 0 до 95 років, медіана – 62 роки.

Кожен запис розмічено клінічними кодами системи РТВ-XL, що утворюють 71 бінарний діагностичний клас. В умовній постановці задачі ці коди подаються моделі як вектор умов, що дає змогу генерувати ЕКГ із морфологічними характеристиками, притаманними конкретному клінічному стану.

Набір розбито на навчальну та тестову частини згідно зі стандартним протоколом: навчальна частина використовувалась для оптимізації параметрів моделі, тоді як тестова вибірка залучалась виключно на етапі фінального оцінювання і зіставлення методів.

4.1.2. Метрики якості генерації

Відстань Фреше між розподілами ознак Inception (FID). Fréchet Inception Distance (FID) [121] кількісно виражає розбіжність між розподілами глибоких ознак реальних і згенерованих зображень. Ознаки отримуються з передостаннього шару мережі Inception v3 [159] – дискримінативної моделі, навченої на ImageNet [160]. Обидва розподіли апроксимуються нормальними розподілами, і відстань між розподілами обчислюється за формулою Фреше:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}\right), \quad (4.1)$$

де μ_r, Σ_r і μ_g, Σ_g – вектори середніх і коваріаційні матриці ознак реальних і згенерованих зображень відповідно. Менший FID відповідає меншій розбіжності між розподілами, тобто кращій якості генерації; для ідентичних розподілів $\text{FID} = 0$.

CLIPScore. CLIPScore [161] – безреференсна метрика відповідності між зображенням і текстовим запитом, що не потребує еталонних прикладів. Вона спирається на модель CLIP [162], навчену контрастним методом на 400 млн пар зображення-текст і здатну зіставляти зображення і текст в єдиному семантичному просторі. Метрика обчислюється як косинусна схожість між CLIP-ембедингами зображення E_I і текстового запиту E_c :

$$\text{CLIPScore}(I, c) = \max(100 \cdot \cos(E_I, E_c), 0). \quad (4.2)$$

Вищі значення CLIPScore свідчать про кращу семантичну узгодженість зображення з текстом. На відміну від FID, що оцінює якість розподілу в цілому, CLIPScore дає поекземплярну оцінку – наскільки конкретне зображення відповідає заданому текстовому запиту. Обидві метрики доповнюють одна одну: FID характеризує глобальну відповідність розподілу, CLIPScore – семантичну точність умовної генерації.

Метрики оцінювання якості синтезованих ЕКГ. Для кількісного порівняння реальних та синтетичних ЕКГ-сигналів застосовувались чотири взаємодоповнювальні метрики, прийняті в роботах із глибокої генерації ЕКГ [46]. Значення кожної метрики обчислювалось окремо для кожного відведення (каналу), після чого результати усереднювались по всіх 12 відведеннях.

Динамічне скривлення часу (DTW). Метод DTW вирівнює два часові ряди через нелінійне деформування часової осі, мінімізуючи накопичену поточкову відстань між ними. Для послідовностей $x = (x_1, \dots, x_n)$ та $y = (y_1, \dots, y_m)$ матриця накопиченої вартості задається рекурентно:

$$D(i, j) = \min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\} + d(x_i, y_j), \quad (4.3)$$

з граничною умовою $D(1, 1) = d(x_1, y_1)$; підсумкова відстань $\text{DTW}(x, y) = D(n, m)$. Як поточкову відстань d зазвичай обирають абсолютне або квадратичне відхилення.

Відстань Вассерштейна. На відміну від DTW, що порівнює окремі пари сигналів, p -відстань Вассерштейна оцінює розбіжність між розподілами цілих вибірок:

$$W_p(\alpha, \beta) = \left(\inf_{\gamma \in \Gamma(\alpha, \beta)} \mathbb{E}_{(x, y) \sim \gamma} [\|x - y\|^p] \right)^{1/p}, \quad (4.4)$$

де $\Gamma(\alpha, \beta)$ – множина всіх спільних розподілів (планів транспортування) з маргіналами α і β . У реалізації для кожного відведення формується вектор ознак, що включає статистичні моменти, часові та спектральні характеристики сигналу; між реальними та згенерованими наборами ознак обчислюється 1-відстань Вассерштейна.

Максимальне середнє розходження (MMD). MMD [163] – ядерна метрика, що вимірює різницю між двома розподілами через порівняння їхніх середніх образів у гільбертовому просторі з відтворювальним ядром. Для позитивно-визначеного ядра k квадрат MMD між розподілами P і Q

визначається як:

$$\text{MMD}^2(P, Q) = \|\mathbb{E}_{x \sim P}[\phi(x)] - \mathbb{E}_{y \sim Q}[\phi(y)]\|_{\mathcal{H}}^2. \quad (4.5)$$

Емпірична оцінка за вибірками $\{x_i\}_{i=1}^m \sim P$ і $\{y_j\}_{j=1}^n \sim Q$ набуває вигляду:

$$\text{MMD}^2(P, Q) = \frac{1}{m^2} \sum_{i,i'} k(x_i, x_{i'}) + \frac{1}{n^2} \sum_{j,j'} k(y_j, y_{j'}) - \frac{2}{mn} \sum_{i,j} k(x_i, y_j). \quad (4.6)$$

Спектральна схожість (Spectral Similarity). Метрика перевіряє збереження частотного вмісту між реальними та синтетичними ЕКГ – зокрема, пікових характеристик, пов'язаних із QRS-комплексами і Т-хвилями. Розраховується як нормована обернена відстань Вассерштейна між щільностями спектральної потужності:

$$\text{SpecSim} = \frac{1}{1 + \frac{1}{n} \sum W(\text{PSD}_{\text{real}}, \text{PSD}_{\text{gen}})}, \quad (4.7)$$

де підсумовування ведеться по всіх відведеннях [164]. На відміну від трьох попередніх метрик, вищі значення SpecSim відповідають кращій якості генерації.

Макроусереднена площа під ROC-кривою (macro-AUC). Чотири попередні метрики оцінюють якість синтезованих сигналів безпосередньо. Для оцінювання практичної користі синтетичних ЕКГ як додаткових навчальних даних застосовано окрему метрику – макроусереднену площу під робочою характеристичною кривою (macro-AUC), стандартну для багатоміткової класифікації ЕКГ за протоколом РТВ-XL [126]. Для кожного діагностичного класу обчислюється площа під ROC-кривою в режимі „один проти решти“, після чого значення усереднюються по всіх класах задачі:

$$\text{macro-AUC} = \frac{1}{C} \sum_{c=1}^C \text{AUC}_c, \quad (4.8)$$

де C – кількість класів. Вищі значення відповідають кращій роздільній здатності класифікатора; макроусереднення надає рівну вагу рідкісним і

поширеним класам, що робить метрику чутливою до якості розпізнавання саме рідкісних діагностичних станів.

Статистичну значущість відмінностей оцінювали методом парного бутстрепа: з тестової вибірки формували 1000 повторних наборів індексів, спільних для базової та порівнюваної умов, тож поекземплярна різниця маско-AUC утворює коректну парну вибірку. За 5-м і 95-м процентилями її розподілу визначали 90% довірчий інтервал.

4.2. Генерація зображень

Запропонований метод узгодження потоків із профілем швидкості перевірено на задачах генерації зображень у двох режимах: навчання моделі з нуля під заданим профілем та адаптація вже навченої моделі без перенавчання. Експерименти на наборі CIFAR-10 охоплюють усі шість профілів швидкості розділу 3, а результати для переднавченої моделі Stable Diffusion 3.5 показують, що метод переноситься на масштабнішу латентну генерацію за текстовим запитом, де дає вигреш за малої кількості кроків.

4.2.1. Експериментальне середовище

Усі експерименти на CIFAR-10 проведено для зображень роздільністю 32×32 із семплюванням за фіксованим методом Ейлера на рівномірній часовій сітці; значення FID обчислювалось відносно 10 000 відкладених тестових зображень. Такий протокол забезпечує зіставність результатів для всіх шести профілів.

Для навчання з нуля використано класо-умовну архітектуру UNet, адаптовану з коду guided-diffusion [10] і налаштовану для зображень 32×32 з тими ж характеристиками, що й TorchCFM [87]: базова ширина 128 каналів, множники каналів (1, 2, 2, 2), два резидуальні блоки на кожну роздільність і блок самоуваги на найнижчій роздільності. Шість профілів навчались за

однаковим рецептом оптимізації – оптимізатор Adam зі швидкістю навчання $2 \cdot 10^{-4}$, розмір пакета 128, експоненційне усереднення ваг із коефіцієнтом 0.9999 і косинусний розклад швидкості навчання, – упродовж 187 500 кроків градієнтного спуску з незалежними спряженнями (x_0, x_1) .

Для адаптації на етапі інференсу кожен профіль швидкості застосовувався до публічно доступного переднавченого чекпойнта TorchCFM [87] без перенавчання згідно з алгоритмом точного методу Ейлера у τ -просторі. Цей чекпойнт навчено на лінійному інтерполянті. Значення FID, опубліковані авторами TorchCFM для нього, отримано адаптивним солвером Дорманда–Прінса (dopri5), тоді як у цій роботі всі профілі, зокрема лінійний базовий, оцінено тим самим фіксованим методом Ейлера. Через це метрики лінійного базового профілю відрізняються від опублікованих, проте порівняння профілів між собою є коректним, оскільки солвер для всіх них спільний.

4.2.2. Навчання з профілями швидкості на CIFAR-10

Результати навчання з нуля для шести профілів швидкості наведено в таблиці 4.1.

Профіль накату досягає найкращого FID на всіх бюджетах кроків: 35.39 при 4 NFE, що на 17.4% нижче за лінійний базовий, 17.12 при 8 NFE (–15.9%) та 8.29 при 50 NFE (–10.4%). Рівномірне сповільнення є другим серед гальмівних профілів, і обидва покращують результат відносно лінійного базового на кожному бюджеті. Виграш найбільший за малої кількості кроків і скорочується зі зростанням NFE, що узгоджується з аналізом похибки дискретизації підрозділу 3.7: за великої кількості кроків розміри кроків достатньо малі, щоб вибір профілю мав обмежений вплив.

Плавне приземлення – єдиний гальмівний профіль, що поступається базовому на всіх трьох бюджетах (53.02 проти 42.86 при 4 NFE; 9.36 проти 9.25 при 50 NFE). Його найбільша серед шести профілів початкова швидкість

Таблиця 4.1

Порівняння якості генерації зображень натренованими моделями, використовуючи різні профілі швидкості, для датасету CIFAR-10 при різній кількості кроків інтегрування

Профіль	FID(4 NFE) ↓	FID(8 NFE) ↓	FID(50 NFE) ↓
<i>Базовий</i>			
лінійний	42.86	20.36	9.25
<i>Гальмівні</i>			
рівномірне сповільнення	37.13	17.74	8.44
накат	35.39	17.12	8.29
плавне приземлення	53.02	22.38	9.36
<i>Симетричні</i>			
плавний вхід-вихід	39.73	18.76	8.62
мінімальний ривок	45.34	21.62	9.02

$v(0) = 3$ відводить найбільші кроки на ділянку поблизу шуму, де поле порівняно гладке, і залишає лише кілька малих кроків для околу $t = 1$, де точність найважливіша. За малої кількості кроків бюджет, витрачений на великі початкові кроки, вносить похибку усічення, що переважає вигравш від дрібних кроків поблизу даних.

Симетричні профілі дають незначне покращення відносно базового при 50 NFE, але поступаються йому при 4 NFE. Їхня дзвоноподібна швидкість концентрує зусилля інтегрування на середині траєкторії, а не поблизу розподілу даних, тож пригнічення похибки припадає на ділянку з нижчою нелінійністю поля.

Окремим практично значущим результатом є пришвидшення збіжності

навчання під гальмівними профілями. На рисунку 4.1 наведено залежність FID від кількості кроків навчання при 4 NFE (верхня панель) і 50 NFE (нижня панель) у логарифмічній шкалі; пунктирною лінією позначено фінальний FID лінійного базового профілю. В обох випадках гальмівні профілі (накат, рівномірне сповільнення) знижують FID швидше за лінійний на ранніх етапах навчання. При 4 NFE накат сягає фінального FID лінійного базового (42.86) уже на кроці $\approx 47\,000$, тобто приблизно за чверть навчального бюджету, тоді як лінійний профіль досягає цієї якості лише наприкінці навчання. При 50 NFE усі профілі збігаються до близьких фінальних значень, проте накат і рівномірне сповільнення досягають фінального FID лінійного (≈ 9.25) приблизно на 15 000 кроків раніше ($\approx 78\,000$ проти $\approx 93\,000$). Це обґрунтовує висновок про скорочення тривалості навчання.

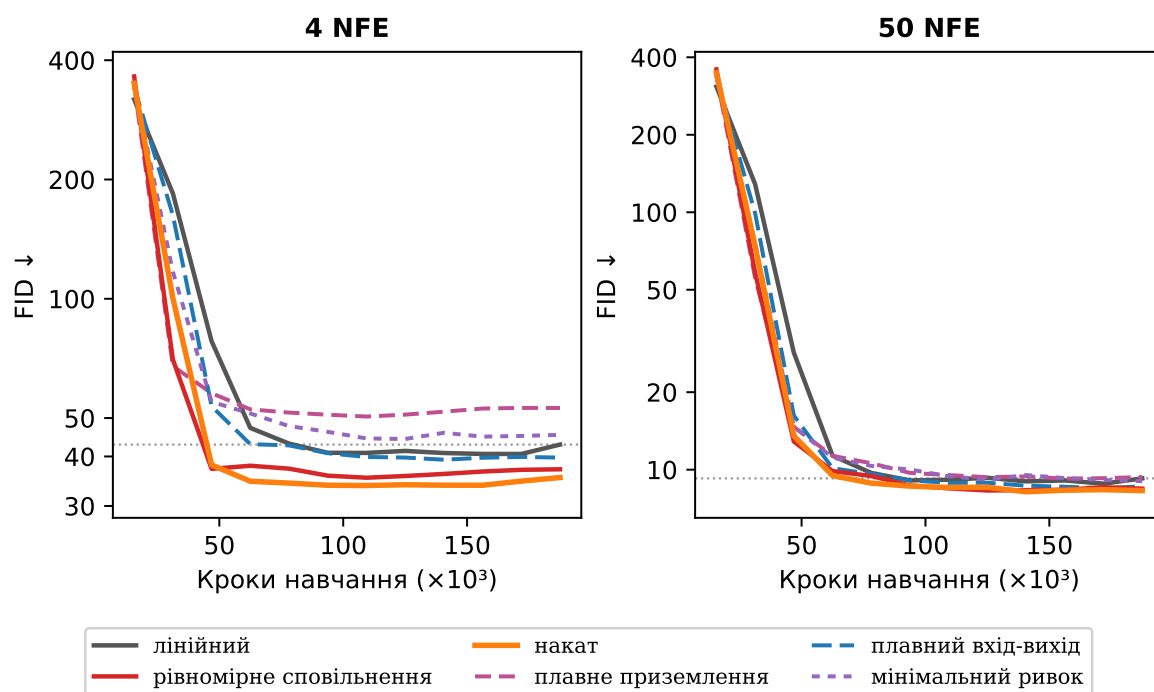


Рис. 4.1. Залежність FID від кількості кроків навчання на CIFAR-10

4.2.3. Адаптація переднавченої моделі на CIFAR-10

Профілі швидкості застосовано до переднавченого чекпойнта TorchCFM без перенавчання, що дає змогу оцінити виграш від переналаштування розкладу інтегрування окремо від ефекту навчання. Результати при 32, 50 і 100 NFE наведено в таблиці 4.2.

Таблиця 4.2

Порівняння якості генерації зображень адаптованою моделлю TorchCFM, використовуючи різні профілі швидкості, для датасету CIFAR-10 при різній кількості кроків інтегрування

Профіль	FID(32 NFE) ↓	FID(50 NFE) ↓	FID(100 NFE) ↓
<i>Базовий</i>			
лінійний	6.43	5.47	4.64
<i>Гальмівні</i>			
рівномірне сповільнення	5.61	4.51	3.89
накат	5.23	4.38	3.90
плавне приземлення	7.43	5.24	4.06
<i>Симетричні</i>			
плавний вхід-вихід	5.46	4.57	4.04
мінімальний ривок	5.64	4.61	4.12

Усі гальмівні профілі, крім плавного приземлення, перевершують лінійний базовий на кожному бюджеті кроків. Накат досягає найкращого FID при 32 і 50 NFE: 5.23 (−18.6% відносно лінійного) і 4.38 (−19.8%), тоді як рівномірне сповільнення лідирує при 100 NFE зі значенням 3.89 (−16.0%). Симетричні профілі дають менше зниження FID, але стабільно покращують

значення відносно лінійного профілю. Висновки повністю повторюють режим навчання з нуля, проте досягаються без перенавчання, що й становить головний практичний результат: переналаштування розкладу інтегрування покращує якість генерації переднавченої моделі без додаткових обчислень на навчання.

Плавне приземлення лишається винятком за найменшого бюджету: його FID при 32 NFE дорівнює 7.43 і перевищує 6.43 лінійного базового. Розподіл розмірів кроку з таблиці 3.2 пояснює цю розбіжність. За найбільшої серед шести профілів початкової швидкості $v(0) = 3$ перші кроки Ейлера покривають утричі більший інтервал τ , ніж у лінійного базового, і за 32 NFE похибка цих великих початкових кроків переважає виграш від дрібних кроків поблизу $t = 1$. Зі зростанням бюджету покрокова похибка зменшується, і перевага малих кроків поблизу розподілу даних починає переважати: при 50 і 100 NFE плавне приземлення досягає 5.24 і 4.06 відповідно. Залежність FID від кількості кроків інтегрування для лінійного базового, трьох гальмівних профілів і найкращого симетричного профілю (плавний вхід-вихід) наведено на рисунку 4.2.

4.2.4. Генерація зображень за текстовим запитом

Перенесення профілів швидкості на масштабнішу задачу перевірено на переднавченій моделі Stable Diffusion 3.5-medium у режимі адаптації без перенавчання. На відміну від CIFAR-10, це латентна модель значно більшого масштабу, навчена генерувати зображення за текстовим запитом, тож експеримент перевіряє, наскільки широко переносяться спостережені переваги.

Профіль швидкості застосовувався тим самим алгоритмом точного методу Ейлера у τ -просторі, що й для адаптації переднавченого чекпойнта на CIFAR-10, без жодного перенавчання. Порівнювалися рідний розклад

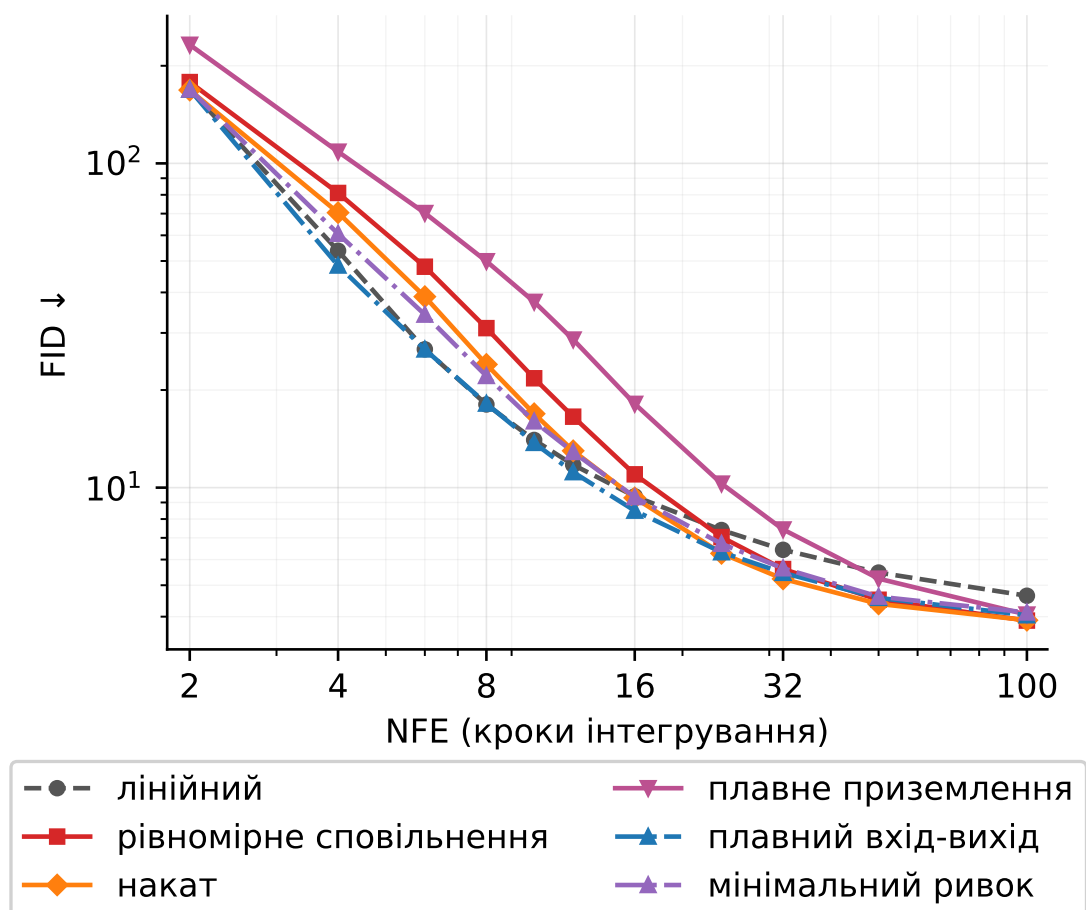


Рис. 4.2. Залежність FID від кількості кроків інтегрування на CIFAR-10 для переднавченої моделі TorchCFM

семплування моделі (позначений *оригінальний* – розклад зі зсувом часових кроків, що постачається з SD3.5), лінійний базовий (рівномірна сітка у τ -просторі, що відповідає лінійному інтерполянту, на якому навчено модель) і нелінійні профілі з розділу 3. Для кожного текстового запиту генерувалось одне зображення, а CLIPScore усереднювалось по всіх запитах набору (1632 для Parti і 200 для DrawBench). Щоб відокремити вплив розкладу інтегрування від ефекту спрямування без класифікатора, основні вимірювання проведено з вимкненим спрямуванням. Оцінювання охоплювало широкий діапазон бюджетів від 2 до 100 NFE; результати при малому бюджеті кроків, де профілі найбільше різняться, наведено в таблиці 4.3.

Таблиця 4.3

Порівняння якості генерації зображень, обумовлених текстом, для переднавченої моделі Stable Diffusion 3.5-medium на наборах Parti та DrawBench за метрикою CLIPScore ($\uparrow$) при малій кількості кроків інтегрування (CFG=1).

Профіль	Parti 4 NFE	Parti 8 NFE	DrawBench 4 NFE	DrawBench 8 NFE
оригінальний (SD3.5)	21.74	27.44	21.67	25.98
<i>Базовий</i>				
лінійний	24.17	27.82	23.60	26.32
<i>Гальмівні</i>				
рівномірне сповільнення	20.46	25.36	21.03	24.34
накат	21.46	26.57	21.51	25.39
плавне приземлення	19.07	22.83	20.09	22.40
<i>Симетричні</i>				
плавний вхід-вихід	25.69	27.86	24.54	26.44
мінімальний ривок	25.25	27.18	24.15	25.90

При малій кількості кроків інтегрування симетричні профілі швидкості покращують CLIPScore відносно лінійного та оригінального на обох наборах. При 4–8 NFE профіль плавного входу-виходу досягає найвищого CLIPScore: на Parti – 25.69 при 4 NFE проти 24.17 у лінійного базового (+1.52) і 21.74 у рідного розкладу SD3.5 (+3.95); на DrawBench – 24.54 проти 23.60 (+0.94) і 21.67 (+2.87). Мінімальний ривок – другий за цим показником (+1.09 на Parti, +0.55 на DrawBench при 4 NFE). Натомість усі гальмівні профілі за малого NFE поступаються і лінійному базовому, і симетричним профілям,

а також оригінальному розкладу на обох наборах: при 4 NFE на Parti накат дає 21.46, рівномірне сповільнення – 20.46, плавне приземлення – лише 19.07 (проти 24.17 у лінійного), а на DrawBench – відповідно 21.51, 21.03 і 20.09 (проти 23.60). Порядок усередині гальмівних профілів узгоджується з їхньою початковою швидкістю: що більша $v(0)$, то гірший результат (накат $v(0) = 1.5$, рівномірне сповільнення 2, плавне приземлення 3). Виграш симетричних профілів зосереджений у режимі малого NFE і зменшується зі зростанням бюджету: за великої кількості кроків профілі збігаються, а при 50 і 100 NFE відмінності незначущі, причому гальмівне плавне приземлення наближається до решти найповільніше. Повну залежність CLIPScore від кількості кроків інтегрування в логарифмічній шкалі наведено на рисунку 4.3: ліворуч – для набору Parti, праворуч – для DrawBench; затінено область малого NFE, де відмінності між профілями найбільші. Це повторює картину похибки дискретизації з розділу 3 та результат на CIFAR-10: за малої кількості кроків розподіл кроків інтегрування суттєвий, за великої – ні.

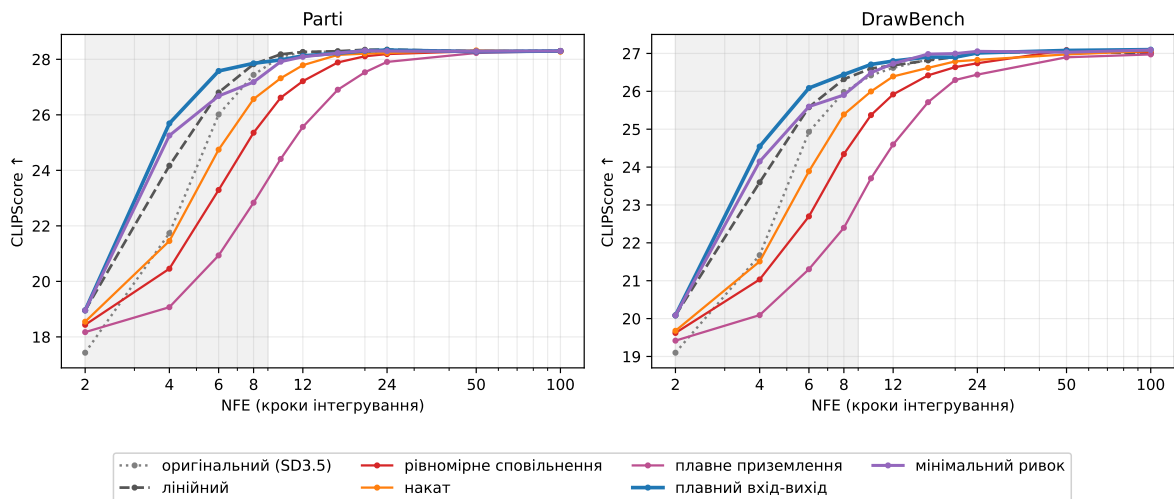


Рис. 4.3. Залежність CLIPScore від кількості кроків інтегрування для моделі Stable Diffusion 3.5-medium на наборах Parti та DrawBench

Слід зазначити, що оптимальний профіль залежить від моделі та задачі.

На CIFAR-10 найкращими були гальмівні профілі, тоді як для генерації за текстовим запитом на SD3.5 виграють симетричні, а гальмівні навпаки шкодять. Спільним для обох задач залишається те, що велика початкова швидкість $v(0)$ – великий перший крок поблизу шуму – шкодить за малого NFE, а симетричні профілі з $v(0) = v(1) = 0$ подрібнюють кроки біля обох кінців траєкторії, зокрема поблизу розподілу даних ($t \rightarrow 1$), і згрубляють лише гладку середину, чим мінімізують похибку інтегрування.

Метрика CLIPScore насичується рано: щойно зображення семантично відповідає запиту (приблизно від 10–16 NFE), її значення перестає зростати й подальше уточнення дрібних деталей майже не відображає. Тому додатково проведено візуальне порівняння згенерованих зображень при 16 і 24 NFE (рис. 4.4; ширшу добірку прикладів для різних запитів наведено в Додатку А). За однакового насиченого CLIPScore симетричні профілі (плавний вхід-вихід і мінімальний ривок) дають помітно чіткіші й деталізованіші зображення, тоді як лінійний і гальмівні профілі видимого покращення не показують. Це узгоджується з розподілом кроків: дрібні кроки поблизу $t \rightarrow 1$ точніше інтегрують кінцеву ділянку траєкторії, де формується високочастотна структура зображення – ефект, який CLIPScore не вловлює, тоді як візуальний огляд виявляє. За ввімкненого спрямування без класифікатора той самий порядок профілів зберігається, проте відмінності між ними помітно зменшуються.

4.3. Синтез електрокардіограм методом узгодження потоків

Перенесення методу за межі генерації зображень перевірено на задачі умовного синтезу електрокардіограм за діагностичними мітками набору РТВ-XL. Розглянуті в підрозділі 1.2.2 дифузійні моделі задають поточний рівень якості в цій задачі, проте потребують сотень прямих проходів через нейронну мережу для породження одного запису. Узгодження потоків



Рис. 4.4. Візуальне порівняння зображень, згенерованих моделлю Stable Diffusion 3.5-medium за різних профілів швидкості

транспортують розподіл шуму до розподілу даних через розв'язання звичайних диференціальних рівнянь [12] і дозволяє скоротити кількість кроків семплювання. Нижче дифузійний ланцюг базової моделі замінено на узгодження потоків за незмінних архітектури та умов навчання, після чого перевірено дію профілів швидкості на цій модальності.

4.3.1. FlowECG: адаптація архітектури SSSD-ECG для узгодження потоків

За основу взято архітектуру SSSD-ECG [45] – метод умовної генерації ЕКГ, що поєднує структуровані моделі простору станів із дифузійним процесом знешумлення. Мережа складається з 36 послідовно об'єднаних резидуальних шарів із двонаправленими S4-блоками: вони замінили розрізнені згортки з іншої архітектури DiffWave [165] і забезпечують ефективне моделювання довготривалих залежностей у сигналах ЕКГ. Кожен резидуальний шар містить 256 каналів. Обумовлення за 71 бінарним діагностичним

класом набору даних РТВ-XL реалізується за допомогою вагових матриць, параметри яких оптимізуються під час навчання: вони відображають вектор міток у неперервний простір ознак і подають отримане представлення до кількох шарів нейронної мережі. Замість безпосередньої генерації всіх 12 відведень модель синтезує лише вісім сигналів (шість прекордіальних, а також відведення I та II), тоді як решта відновлюються на основі електрофізіологічних співвідношень: $III = II - I$, $aVL = (I - III)/2$, $aVF = (II + III)/2$ і $-aVR = (I + II)/2$.

Метод FlowECG [25] зберігає повну архітектуру SSSD-ECG, замінюючи дискретний дифузійний ланцюг на неперервне узгодження потоків. Упродовж навчання час семплюється рівномірно $t \sim \mathcal{U}(0, 1)$, проміжний стан x_t формується лінійною інтерполяцією (1.29) між цільовим ЕКГ-сигналом x_1 і гауссівським шумом x_0 , а мережа навчається апроксимувати константне векторне поле $v_t = x_1 - x_0$, що задає пряме транспортування від шуму до даних. Цільовою функцією є умовний варіант (1.32), у якому векторне поле додатково залежить від вектора діагностичних міток:

$$\mathcal{L} = \mathbb{E} \|v_\theta(x_t, c, t) - (x_1 - x_0)\|^2, \quad (4.9)$$

де v_θ – нейронна мережа, c – вектор діагностичних міток, а математичне сподівання береться по зразках даних, шуму, часових кроках і умовах кондиціювання. Порівняно зі зваженою варіаційною нижньою оцінкою дифузійних моделей, цей функціонал простіший і не потребує апроксимації скор-функції.

Генерація нового сигналу зводиться до розв'язання рівняння (1.33) з умовним полем $v_\theta(x_t, c, t)$ – від гауссівського початкового стану до цільового розподілу методом Ейлера. Кількість кроків інтегрування (NFE) залишається гіперпараметром: менше кроків – нижча обчислювальна вартість генерації. Параметри оптимізації збігаються з базовою моделлю – оптимізатор Adam із швидкістю навчання $2 \cdot 10^{-4}$ і розміром пакета 6, – що

забезпечує коректне порівняння методів за однакових умов навчання.

4.3.2. Результати та аналіз ефективності

Порівняльне оцінювання проводилось між FlowECG і перенавченою моделлю SSSD-ECG за ідентичних архітектурних і тренувальних конфігурацій. Оцінювалась здатність обох моделей умовно генерувати якісні ЕКГ у 12 відведеннях за діагностичними мітками PTB-XL. Аналіз охоплював візуальну оцінку, кількісне порівняння за чотирма метриками та дослідження залежності якості від кількості NFE.

Візуальне порівняння реальних записів (ліва частина рис. 4.5) із згенерованими (права частина) підтверджує, що FlowECG відтворює фізіологічно обґрунтовані морфологічні характеристики в усіх 12 відведеннях: ритмічну структуру, амплітудні співвідношення між відведеннями і характерні форми хвиль. Часові залежності і міжвідведенні кореляції збережено, що свідчить про здатність моделі відтворювати складну структуру розподілу багатовимірних ЕКГ-сигналів.

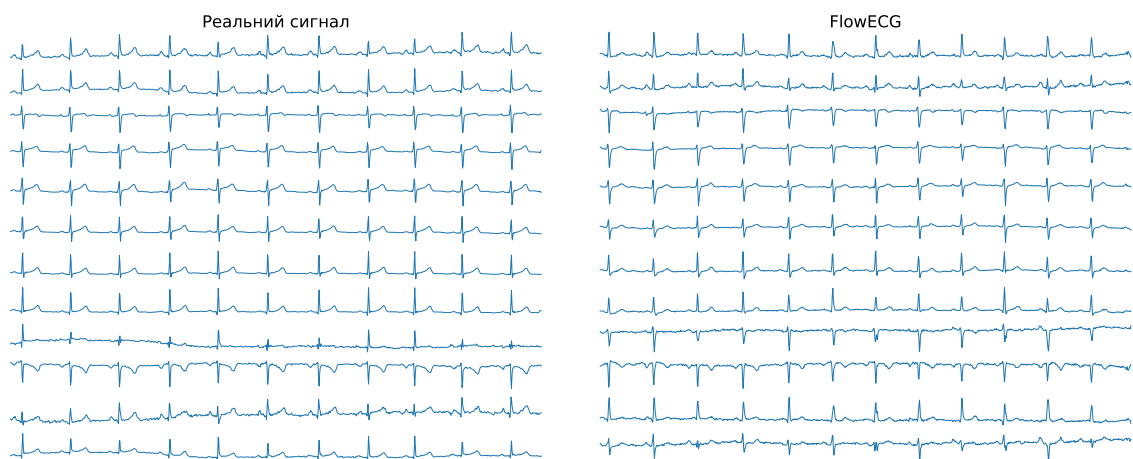


Рис. 4.5. Порівняння реальних і згенерованих FlowECG сигналів ЕКГ у 12 відведеннях

Кількісне порівняння при 200 NFE наведено в таблиці 4.4. FlowECG перевершує базову модель за трьома з чотирьох показників: MMD знизилося

з 80.70 до 21.09, відстань Вассерштейна – з 1.03 до 0.62, а спектральна схожість зросла з 0.30 до 0.37. За метрикою DTW FlowECG поступається базовій моделі: 83.96 проти 73.49 – що відповідає відмінності між покроковим дифузійним і неперервним потоковим підходами до оптимізації траєкторій.

Таблиця 4.4

Порівняння метрик якості SSSD-ECG і FlowECG при 200 NFE

	MMD ↓	DTW ↓	Wasserstein ↓	SpecSim ↑
SSSD-ECG	80.70	73.49	1.0308	0.3000
FlowECG	21.09	83.96	0.6204	0.3687

Найбільш практично значущий результат від переходу до методу узгодження потоків отримано при аналізі залежності якості від кількості кроків інтегрування (рис. 4.6). Для SSSD-ECG скорочення NFE призводить до різкої деградації якості: при 2 кроках DTW перевищує 700, а до базового рівня (близько 73) метрика наближається лише за 50 і більше кроків; аналогічну тенденцію простежуємо й за іншими метриками. FlowECG натомість зберігає стабільну якість при суттєво меншому NFE: 10-25 кроків достатньо для результатів, порівнянних із SSSD-ECG при 200 кроках.

Це пояснюється різницею у природі двох генеративних процесів. Дифузійні моделі прив'язані до розкладу зашумлення і потребують великої кількості кроків для точного обернення процесу. Векторні поля узгодження потоків – прямолінійніші і гладші, тому ефективно апроксимуються методом Ейлера за малим числом кроків. Десятикратне скорочення обчислювальних витрат робить FlowECG придатним для пакетної генерації синтетичних наборів даних і ресурсообмежених систем підтримки клінічних рішень.

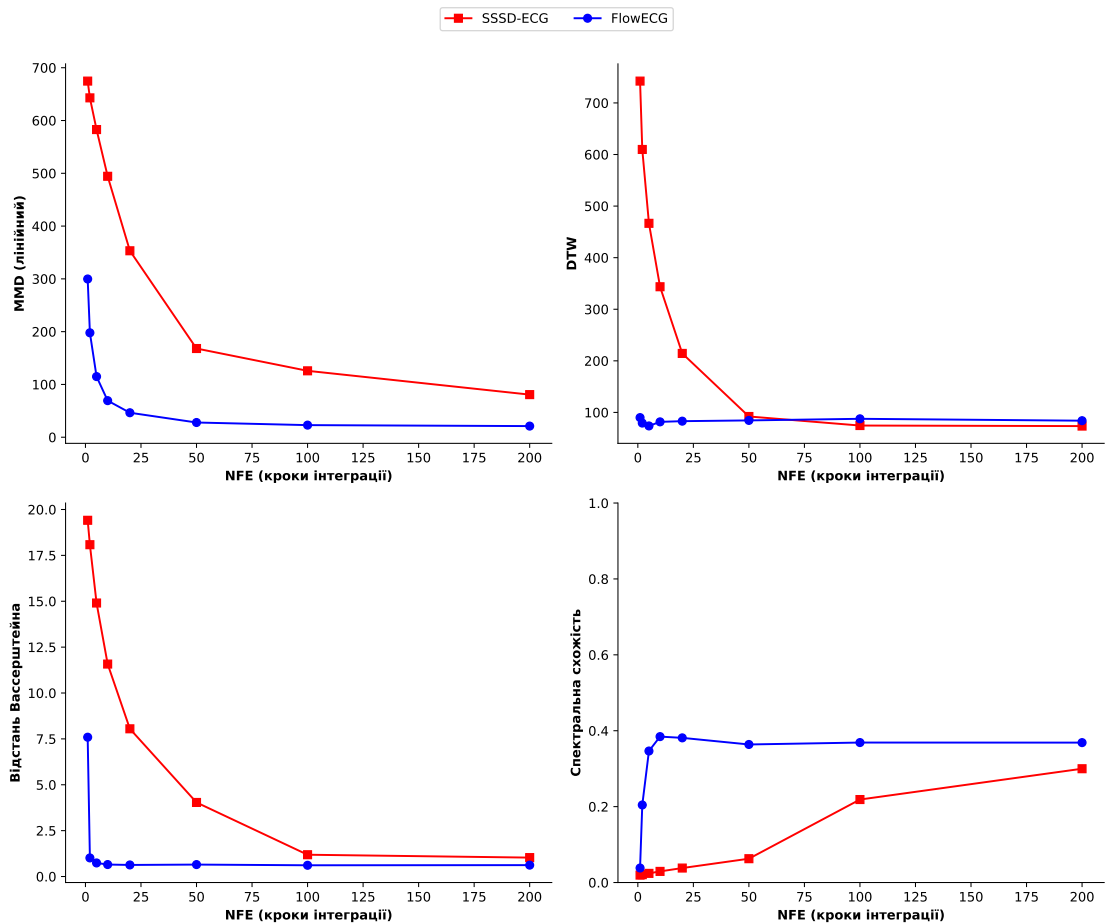


Рис. 4.6. Залежність метрик якості від кількості кроків інтегрування для SSSD-ECG і FlowECG

4.3.3. Профілі швидкості у FlowECG

Профілі швидкості, розроблені для генерації зображень, переносяться й на синтез електрокардіограм у межах методу FlowECG. Базова модель навчається на лінійному інтерполянті, а нелінійні профілі застосовуються на етапі інференсу як переналаштування розкладу інтегрування без перенавчання, аналогічно до адаптації переднавчених моделей на CIFAR-10. Метрики якості при 200 NFE для лінійного та п'яти нелінійних профілів наведено в таблиці 4.5.

За результатами таблиці 4.5 перевага при 200 NFE належить симетричному профілю плавного входу-виходу: він досягає найкращих значень MMD

Таблиця 4.5

Порівняння метрик якості FlowECG при 200 NFE та різних профілях швидкості

	MMD ↓	DTW ↓	Wasserstein ↓	SpecSim ↑
FlowECG (лінійний)	21.09	83.96	0.6204	0.3687
FlowECG (рівномірне сповільнення)	21.17	79.21	0.5903	0.3772
FlowECG (накат)	21.84	79.58	0.6405	0.3623
FlowECG (плавне приземлення)	21.37	80.53	0.6414	0.3515
FlowECG (плавний вхід-вихід)	19.37	82.14	0.5880	0.3782
FlowECG (мінімальний ривок)	21.43	90.71	0.6045	0.3677

(19.37 проти 21.09 у лінійного), відстані Вассерштейна (0.5880 проти 0.6204) та спектральної схожості (0.3782 проти 0.3687), а рівномірне сповільнення дає найкращий показник DTW (79.21 проти 83.96) і переважає лінійний за рештою метрик. Гальмівні профілі накату та плавного приземлення покращують лише DTW, тоді як за відстанню Вассерштейна та спектральною схожістю поступаються лінійному базовому (0.6405 і 0.6414 проти 0.6204 за Вассерштейном). Отже, на відміну від CIFAR-10, де перевагу мали гальмівні профілі, для синтезу ЕКГ при 200 NFE виграє симетричний профіль, а гальмівні профілі дають змішаний результат.

Окремо перевірено прогноз аналізу похибки дискретизації розділу 3 щодо переваги гальмівних профілів за малої кількості кроків. На рисунку 4.7 наведено залежність чотирьох метрик якості від кількості кроків інтегрування у діапазоні від 2 до 200 NFE (логарифмічна шкала) для всіх шести профілів. Прогнозованої для зображень переваги гальмівних профілів за

малого NFE на ЕКГ не виявлено: за найменших бюджетів (2 і 5 NFE) лінійний базовий профіль є найкращим за всіма чотирма метриками, а кожен гальмівний профіль поступається йому. Єдиний виняток – метрика DTW, за якою гальмівний профіль випереджає лінійний лише на середніх бюджетах (10–100 NFE), але не за 2–5 NFE, де очікувався найбільший виграш.

Метрики якості генерації залежно від кількості кроків (NFE) за профілем

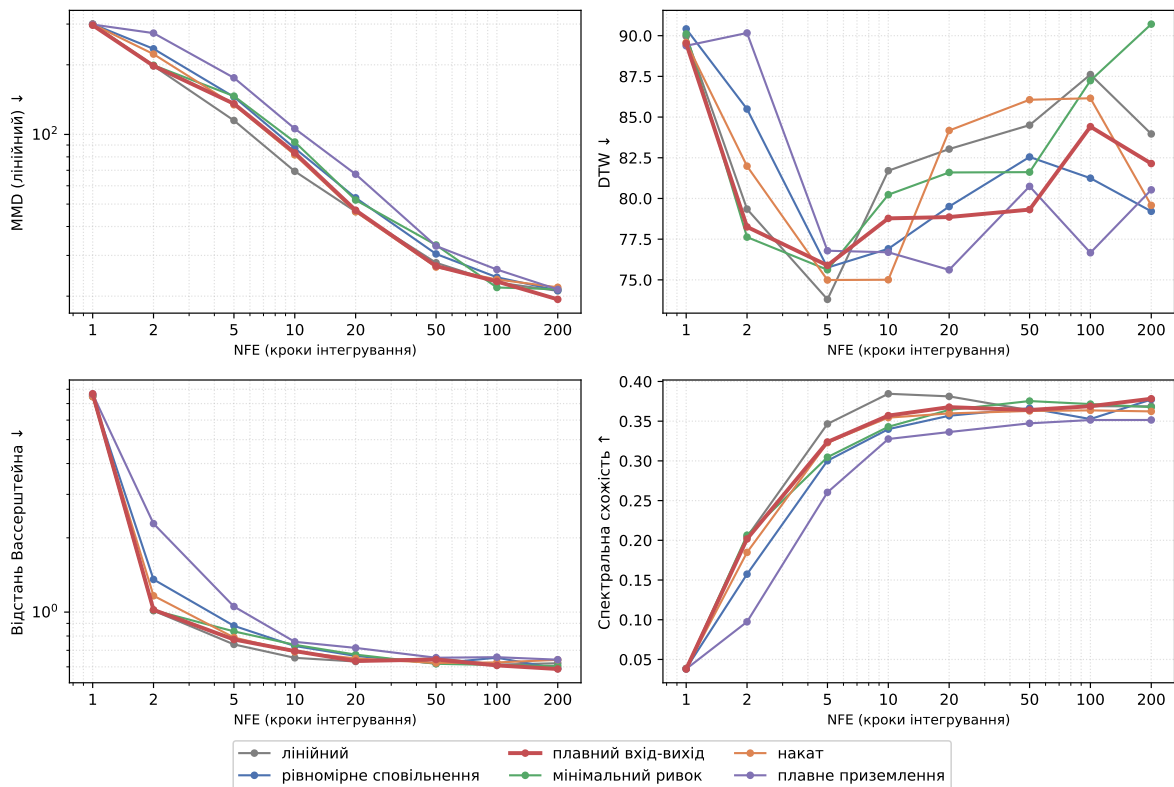


Рис. 4.7. Залежність метрик якості від кількості кроків інтегрування для FlowECG за шести профілів швидкості

Цей результат уточнює межі застосовності методу. Сама процедура переналаштування розкладу інтегрування переноситься на синтез ЕКГ: профіль швидкості застосовано до вже навченої моделі FlowECG без перенавчання, і симетричний профіль дає приріст якості за рахунок самого лише перерозподілу кроків інтегрування. Проте оптимальний профіль і величина виграшу залежать від модальності даних: для зображень за малого NFE виграють гальмівні профілі, для ЕКГ – симетричний за повного бюджету, тоді як

низькокроковий виграш гальмівних профілів, показаний на CIFAR-10, на датасеті PTB-XL не відтворюється.

4.3.4. Розширення навчальних даних синтетичними ЕКГ

Здатність FlowECG генерувати фізіологічно обґрунтовані сигнали створює підстави для практичного застосування – розширення навчальних вибірок синтетичними записами. Набір PTB-XL має різко незбалансований розподіл класів: синусовий ритм присутній у 76.98% навчальних записів, тоді як окремі патологічні стани, як-от задній інфаркт міокарда, трапляються менш ніж у п'ятнадцяти записах. Класифікатори, навчені на таких даних, гірше розпізнають рідкісні, але клінічно важливі стани. Стандартним засобом проти дисбалансу є передискретизація – дублювання або підвищення ваги записів рідкісних класів [166]. У цьому підрозділі перевірено альтернативу: доповнення навчальної вибірки синтетичними ЕКГ, згенерованими FlowECG, з акцентом на рідкісні класи на етапі семплювання міток генератора, а не дублюванням реальних записів.

Семплювання міток за оберненою частотою. Для кожного синтетичного зразка вектор діагностичних міток вибирається з пулу реальних навчальних записів PTB-XL, що зберігає клінічно достовірні комбінації міток. Нехай f_c – емпірична частота класу c , а $y_{i,c} \in \{0, 1\}$ позначає активність класу c у записі i . Вага семплювання запису визначається оберненою частотою найрідкіснішого з активних у ньому класів:

$$w_i = \max_c \left(y_{i,c} \cdot \frac{1}{\max(f_c, \varepsilon)} \right), \quad (4.10)$$

а записи добираються з імовірністю, пропорційною w_i . За такого зважування поширеність синусового ритму у вибраних мітках знижується з 76.98% до 60.30%, поширеність нормальної ЕКГ (клас NORM) – з 43.62% до 12.35%, а частка десяти найрідкісніших класів зростає з медіанних 0.08% до приблизно 2%, тобто в середньому у 25 разів. Оскільки мітки беруться з реальних

рядків сумісної появи, а не складаються незалежно, клінічні патерни зберігаються.

Протокол оцінювання. Згенеровано 20 000 збалансованих синтетичних записів і долучено їх до 17 418 реальних навчальних записів PTB-XL (фолди 1–8). Перевірний фолд 9 і тестовий фолд 10 залишалися суто реальними на всіх етапах: жоден синтетичний запис не потрапляв до них. Класифікація проводилась у межах стандартного бенчмарку PTB-XL [126] для трьох еталонних згорткових архітектур – XResNet-1D-101, ResNet-1D та Inception-1D – і їхнього ансамблю за м'яким голосуванням. Кожну мережу навчали впродовж 25 епох із косинусним розкладом швидкості навчання та розміром пакета 128. Якість оцінювали за макроусередненою AUC у режимі „один проти решти“ на фолді 10 для шести задач бенчмарку: усі 71 код SCP (exp0), діагностична (44 коди, exp1), піддіагностична (23 коди, exp1.1), наддіагностична (5 кодів, exp1.1.1), форма (19 кодів, exp2) і ритм (12 кодів, exp3).

Результати розширення вибірки. Результати ансамблю для основного варіанту з 20 000 збалансованих синтетичних записів наведено в таблиці 4.6. Долучення синтетичних даних статистично значуще покращує якість на трьох діагностичних задачах: діагностична AUC зростає з 0.9387 до 0.9502 ($\Delta = +0.0115$, $p < 0.001$), задача всіх кодів – з 0.9317 до 0.9422 ($\Delta = +0.0105$, $p = 0.001$), піддіагностична – з 0.9308 до 0.9395 ($\Delta = +0.0086$, $p = 0.001$). Наддіагностична задача дає граничний результат ($\Delta = +0.0017$, $p = 0.057$), а на задачах форми та ритму значущих змін немає.

Щоб відокремити внесок перерозподілу міток від внеску варіативності самих сигналів, проведено два абляційні експерименти, у кожному з яких до реальних фолдів долучали 20 000 записів (таблиця 4.7). Перший замінює семплювання за оберненою частотою на емпіричний розподіл PTB-XL за того самого генератора. На задачі всіх кодів емпіричний варіант не дає

Таблиця 4.6

Макроусереднена AUC $\uparrow$ ансамблю на тестовому фолді 10 РТВ-XL для основного варіанту розширення з 20 000 збалансованих синтетичних записів

Задача	Класів	Базова AUC	Розширена AUC	Δ AUC
усі коди SCP	71	0.9317	0.9422	+0.0105
діагностична	44	0.9387	0.9502	+0.0115
піддіагностична	23	0.9308	0.9395	+0.0086
наддіагностична	5	0.9338	0.9355	+0.0017
форма	19	0.9094	0.9110	+0.0016
ритм	12	0.9611	0.9602	-0.0009

значущого покращення ($\Delta = +0.0017$, $p = 0.262$), тоді як збалансований дає $\Delta = +0.0105$ ($p = 0.001$); парний тест різниці приростів підтверджує перевагу збалансованого варіанта ($+0.0088$, $p < 0.001$). Отже, семплювання за оберненою частотою є визначальним чинником приросту на рівні всіх кодів.

Другий експеримент замінює синтетичний генератор контролем із передискретизацією, що дублює 20 000 реальних записів за того самого зважування за оберненою частотою. Цей контроль знижує AUC на кожній задачі – від -0.0149 до -0.0367 . Оскільки 20 000 дублікатів походять приблизно з 6 700 унікальних записів, найрідкісніші класи повторюються 40-100 разів, класифікатор перенавчається на них, і тестова AUC падає. Розширення синтетичними даними цього не спричиняє, оскільки кожен згенерований зразок поділяє мітки з реальним записом, але відрізняється від нього морфологією сигналу. Парна різниця між збалансованим розширенням і контролем додатна та значуща на всіх задачах ($p \leq 0.001$). Це показує, що приріст залежить від морфологічної варіативності новосинтезованих

сигналів, а не лише від перебалансування міток.

Таблиця 4.7

Приріст $\Delta \uparrow$ макроусередненої AUC ансамблю для трьох варіантів розширення відносно базової моделі на реальних даних (20 000 записів у кожному).

Задача	синтез (збалансований)	синтез (емпіричний)	передискретизація
усі коди SCP	+0.0105	+0.0017	−0.0181
діагностична	+0.0115	+0.0082	−0.0205
піддіагностична	+0.0086	+0.0087	−0.0149
наддіагностична	+0.0017	+0.0019	−0.0158
форма	+0.0016	−0.0055	−0.0260
ритм	−0.0009	0.0000	−0.0367

Залежність приросту від обсягу синтетичної вибірки оцінено для наборів у 5 000, 10 000, 20 000 і 40 000 записів (рис. 4.8). Для чотирьох задач бенчмарку РТВ-XL збалансоване розширення порівнюється з базовою моделлю на реальних даних, варіантом з емпіричним розподілом міток і контролем із передискретизацією (обидва при 20 000 записів). На діагностичній задачі приріст монотонно зростає на всьому діапазоні – +0.0075 при 5 000, +0.0109 при 10 000, +0.0115 при 20 000 і +0.0137 при 40 000 записів – і не виходить на насичення навіть за обсягу, удвічі більшого за реальну навчальну вибірку. На піддіагностичній задачі приріст приблизно сталий (близько +0.009 на всьому діапазоні), тож користь досягається вже за найменшого обсягу.

Профілі швидкості у розширенні вибірки. Метод узгодження потоків із профілем швидкості перевірено й на задачі розширення. З єдиного навченого на лінійному інтерполянті чекпойнта FlowECG згенеровано шість збалансованих наборів по 20 000 записів – по одному на кожен профіль швидкості розділу 3 – з ідентичного набору міток зі спільним зерном

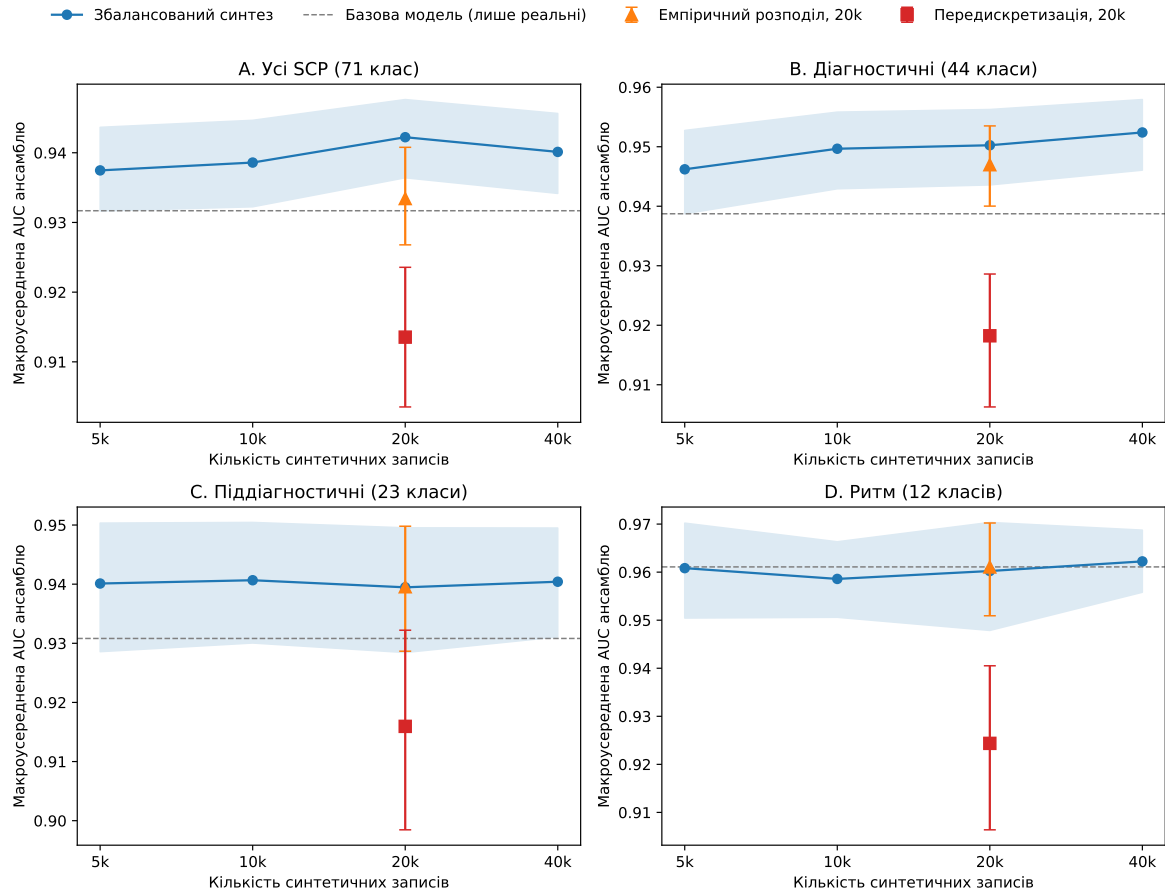


Рис. 4.8. Залежність макроусередненої AUC ансамблю від обсягу синтетичної навчальної вибірки

генератора, тож набори відрізняються лише реалізацією сигналів, а кожне порівняння є парним. Усі шість профілів дають значущий приріст на трьох діагностичних задачах (від $+0.009$ до $+0.015$ за діагностичною AUC), що відтворює й узагальнює результат лінійного розкладу: користь розширення є властивістю даних і стійка до вибору профілю (рис. 4.9); зірочкою на рисунку позначено прирости, 90% довірчий інтервал яких виключає нуль.

Водночас за повного бюджету 200 NFE, на якому генеровано набори, профіль швидкості є щонайбільше другорядним чинником: жоден профіль не перевершує лінійний базовий надійно, а відмінності між профілями порівнянні з варіацією самого конвеєра генерації та навчання. Два незалежні запуски за лінійного профілю в однаковій конфігурації різняться

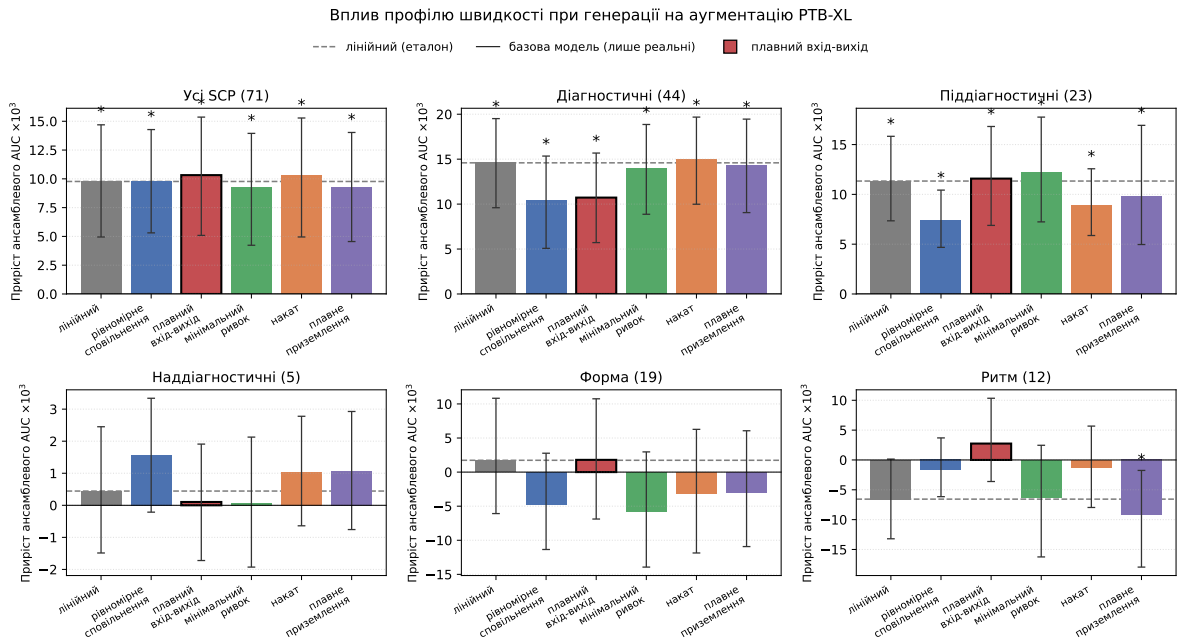


Рис. 4.9. Приріст макросередньої AUC ансамблю за шести профілів швидкості для задач бенчмарку PTB-XL

приростом на $+3.1$, $+2.7$ і -5.7×10^{-3} за діагностичною, піддіагностичною та ритмічною задачами, тоді як розкид між різними профілями на тих самих задачах становить лише 4.6 , 4.9 і 12.0×10^{-3} . Отже, варіація між двома запусками одного профілю має той самий порядок, що й варіація між різними профілями, тож ранжування профілів за цим показником не є надійним.

Варто відзначити, що метрики якості генерації не передбачають корисності профілю для розширення. Рангові кореляції Спірмена між чотирма метриками таблиці 4.5 та приростом діагностичної AUC за шістьма профілями слабкі й непослідовні за знаком (рис. 4.10): профіль із найкращою якістю генерації (плавний вхід-вихід) посідає середню позицію за корисністю для розширення, а профілі з гіршою якістю розширюють діагностичні задачі не гірше. Звідси випливає практичний висновок: профіль для розширення не можна обрати за метриками MMD, DTW, Вассерштейна чи спектральної схожості – його належить перевіряти безпосередньо на цільовій задачі.

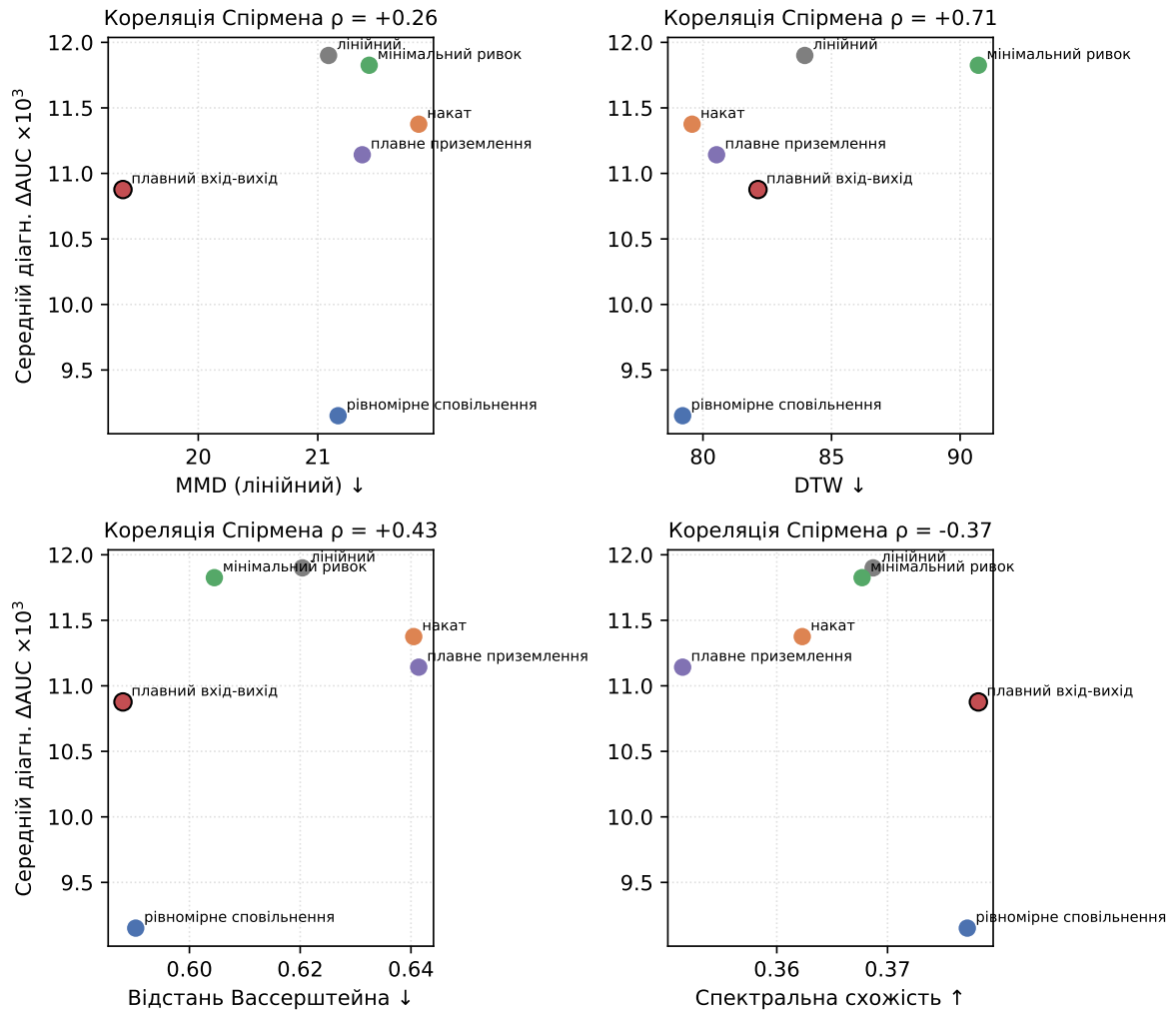


Рис. 4.10. Метрики якості генерації та приріст діагностичної AUC за шести профілів швидкості

4.4. Висновки до розділу 4

У четвертому розділі дисертації проведено експериментальну перевірку методу узгодження потоків із профілем швидкості на задачах генерації зображень і синтезу електрокардіограм. Перевірку виконано у двох режимах – навчання моделі з нуля під заданим профілем та переналаштування розкладу інтегрування для вже навчених моделей без перенавчання, – із застосуванням метрик FID, CLIPScore та чотирьох метрик відстані між розподілами для ЕКГ.

На наборі CIFAR-10 підтверджено прогноз аналізу похибки дискретиза-

ції з розділу 3. За навчання з нуля гальмівні профілі покращують FID на всіх бюджетах кроків, причому профіль накату дає найбільший виграш – до 17.4% при 4 NFE, – а виграш найбільший за малої кількості кроків. Гальмівні профілі також пришвидшують збіжність навчання: профіль накату сягає фінальної якості лінійного базового приблизно за чверть навчального бюджету при 4 NFE, що обґрунтовує скорочення тривалості навчання.

Показано, що переналаштування розкладу інтегрування покращує якість генерації переднавчених моделей без перенавчання. Для чекпойнта TorchCFM на CIFAR-10 профіль накату знижує FID до 19.8% відносно лінійного базового без жодних витрат на навчання. Метод переноситься й на масштабну латентну модель Stable Diffusion 3.5: за малої кількості кроків симетричний профіль плавного входу-виходу підвищує CLIPScore на обох наборах текстових запитів (до +1.5 при 4 NFE), а візуальне порівняння виявляє виграш у деталізації навіть там, де CLIPScore вже насичена. Оптимальний профіль при цьому залежить від моделі – симетричний для SD3.5 проти гальмівного для CIFAR-10, – що підкреслює гнучкість методу переналаштування розкладу.

Підтверджено перенесення методу поза генерацію зображень, проте з уточненням меж застосовності. У межах FlowECG сама процедура переналаштування розкладу переноситься на синтез ЕКГ, і за повного бюджету 200 NFE симетричний профіль плавного входу-виходу покращує якість за метриками MMD, відстані Вассерштейна та спектральної схожості відносно лінійного базового. Водночас оптимальний профіль і величина виграшу залежать від модальності даних: доведена на CIFAR-10 перевага гальмівних профілів за малої кількості кроків на ЕКГ не відтворюється – за 2–5 NFE лінійний базовий профіль є найкращим за всіма чотирма метриками.

Окремим практичним результатом є застосування FlowECG до розширення навчальних вибірок. Долучення 20 000 синтетичних ЕКГ із семплюванням міток за оберненою частотою підвищує макроусереднену AUC ансам-

блю класифікаторів РТВ-XL на діагностичній (з 0.9387 до 0.9502), піддіагностичній і повнокодovій задачах. Абляції локалізують ефект: визначальним чинником на рівні всіх кодів є семплювання за оберненою частотою, а порівняння з контролем із передискретизацією показує, що приріст залежить від морфологічної варіативності синтезованих сигналів, а не лише від перебалансування міток. Приріст на діагностичній задачі не виходить на насичення навіть за 40 000 записів. Користь розширення стійка до вибору профілю швидкості, тоді як сам профіль за 200 NFE є щонайбільше другим чинником, порівнянним із варіацією конвеєра, а метрики якості генерації не передбачають корисності профілю для розширення.

ВИСНОВКИ

У дисертаційній роботі розв'язано актуальну науково-прикладну задачу підвищення якості нейромережевої генерації об'єктів за обмеженого обчислювального бюджету на основі моделей генеративного процесу та масштабування обчислень із розробленням удосконаленого методу узгодження потоків, що використовує нелінійні профілі швидкості вздовж траєкторій перетворення розподілів. Відповідно до поставлених завдань одержано такі основні результати.

1. За результатами аналізу наявних генеративних моделей виявлено їхні спільні ознаки та принципові відмінності й установлено, що варіаційні автокодувальники, авторегресивні системи, генеративні змагальні мережі, нормалізуючі потоки, дифузійні моделі та методи узгодження потоків є різними способами розв'язати одну задачу: апроксимувати невідомий розподіл реальних даних і забезпечити можливість семплювання з нього. При цьому відмінності між моделями зводяться переважно до способу параметризації перетворення між простим базовим і складним цільовим розподілами та до відповідних компромісів щодо точності оцінювання густини, стабільності навчання, швидкості інференсу й якості синтезованих зразків, а аналіз методу узгодження потоків виявив невирішене питання вибору профілю швидкості вздовж траєкторії, оскільки стандартний підхід із постійною швидкістю є лише найпростішим варіантом, що й стало підґрунтям для подальшого дослідження.

2. Побудовано узагальнену модель генеративного процесу, яка подає основні класи генеративних моделей як кортеж із базового розподілу, функції трансформації розподілів, цільової функції навчання та множини допоміжних відображень. Показано, що попри суттєві відмінності в архітектурах,

методах навчання та теоретичному обґрунтуванні всі розглянуті моделі здійснюють відображення елементів наперед визначеного простого розподілу в елементи складного розподілу даних, яке формалізується через класичну формулу заміни змінних або її узагальнення, причому в генерації зразка задіяно лише базовий розподіл і функцію трансформації, тоді як відмінності між класами зосереджено у цільовій функції навчання та допоміжних відображеннях. Межі моделі задано трьома умовами: фіксованістю базового розподілу, відокремленням структури процесу від якості наближення розподілу даних і застосовністю формули заміни змінних лише до оборотних трансформацій. Це дозволило порівнювати підходи за значеннями компонент кортежу, переносити напрацьовані рішення між методами та створило теоретичну основу для розроблення удосконаленого методу.

3. Побудовано структурно-топологічну модель масштабування обчислень під час генерації. Структурну складову моделі утворює оператор, заданий трійкою політик – генерації, відбору та розподілу обчислювальних ресурсів – за умови неспадання математичного сподівання якості зі зростанням обчислювального бюджету. Межі моделі виражено в математичній формі трьома співвідношеннями: щонайменше лінійним зростанням вартості, сублінійним приростом якості та обмеженням результату з боку генератора й верифікатора. Топологічну складову моделі утворюють шість базових обчислювальних топологій, виділених на основі аналізу методів із різних галузей машинного навчання: послідовне уточнення, паралельне семплювання з агрегацією, деревовидний та графовий пошук, адаптація на етапі інференсу, керована генерація й адаптивна маршрутизація обчислень. Виявлено також чотири інваріанти, що проявляються незалежно від конкретної топології: сублінійну криву „вартість – якість“, обмеження з боку верифікатора, послідовно-паралельну дуальність методів і міграцію методів між доменами. Це дозволило зіставляти в єдиних термінах методи

масштабування з різних галузей машинного навчання.

4. Розроблено удосконалений метод узгодження потоків із профілем швидкості, що узагальнює стандартну побудову заміною сталого цільового поля $x_1 - x_0$ часозалежним полем $v(t) (x_1 - x_0)$ для довільного допустимого профілю швидкості, причому формально встановлено, що таке узагальнення зберігає ймовірнісний шлях за монотонної заміни часу, а отже не змінює ні архітектури мережі, ні напрямку транспортування, ні характеру спряження розподілів, а лише перерозподіляє той самий рух у часі. На основі задач планування руху побудовано шість поліноміальних профілів швидкості (базовий лінійний, три гальмівні та два симетричні), кожен із яких задається крайовими умовами на профіль руху та його похідні, і показано, що метод допускає два застосування: переналаштування розкладу інтегрування вже навченої лінійної моделі на етапі інференсу без перенавчання та навчання з нуля під неконстантним профілем. Аналіз локальної похибки усічення методу Ейлера виявив механізм, що пояснює обидва застосування: гальмівні профілі пригнічують доданок варіації поля поблизу кінця траєкторії, де навчене векторне поле змінюється найшвидше, звідки впливає прогноз про перевагу гальмівних профілів із найбільшим виграшем за малої кількості кроків інтегрування.

5. Проведено експериментальну перевірку розробленого методу на задачах генерації зображень, яка підтвердила прогноз аналізу похибки дискретизації: на наборі CIFAR-10 за навчання з нуля гальмівні профілі покращують FID на всіх бюджетах кроків, причому профіль накату дає найбільший виграш (до 17.4% при чотирьох кроках інтегрування), а сам виграш зростає зі зменшенням кількості кроків; крім того, гальмівні профілі також пришвидшують збіжність навчання, досягаючи фінальної якості лінійного базового приблизно за чверть навчального бюджету. Переналаштування розкладу інтегрування покращує якість генерації переднавчених моделей без перенавчання: для чекпойнта TorchCFM на CIFAR-10 профіль накату

знижує FID до 19.8% відносно лінійного базового без жодних витрат на навчання, а на масштабній латентній моделі Stable Diffusion 3.5 за малої кількості кроків симетричний профіль плавного входу-виходу підвищує CLIPScore на наборах текстових запитів Parti та DrawBench, звідки випливає, що оптимальний профіль залежить від моделі та бюджету кроків, а сама процедура переналаштування розкладу залишається корисною.

6. Доведено застосовність узгодження потоків і розроблених профілів швидкості поза межами генерації зображень на прикладі синтезу електрокардіограм: метод FlowECG, який замінює дифузійний процес узгодженням потоків, зберігає якість генерації за 10–25 кроків проти близько 200 у базовій дифузійній моделі SSSD-ECG, що відповідає скороченню обчислювальних витрат приблизно в десять разів за одночасного покращення якості за метриками максимальної середньої розбіжності, відстані Вассерштейна та спектральної схожості. Окремим практичним результатом є застосування FlowECG до розширення навчальних вибірок, за якого долучення синтетичних електрокардіограм із семплюванням міток за оберненою частотою підвищує макроусереднену AUC ансамблю класифікаторів PTB-XL на діагностичній задачі з 0.9387 до 0.9502. Водночас уточнено межі застосовності: доведена на CIFAR-10 перевага гальмівних профілів за малої кількості кроків для електрокардіограм не відтворюється, тож оптимальний профіль швидкості належить налаштовувати під конкретну модальність даних і бюджет кроків інтегрування.

Практичне значення одержаних результатів визначається тим, що запропоновані моделі та методи дозволяють будувати алгоритми генерації, які знижують її обчислювальну вартість без зміни архітектури моделі та без перенавчання, а також алгоритми розширення навчальних вибірок у задачах, де збір реальних даних ускладнений. Результати дисертаційної роботи впроваджено в навчальний процес Черкаського державного технологічного університету та Черкаського національного університету імені

Богдана Хмельницького (додаток Б).

Перспективи подальших досліджень полягають у поширенні методу нелінійних профілів швидкості на інші модальності генерації (аудіо, відео, тривимірні дані та часові ряди), у розробленні процедур автоматичного добору профілю під задану модальність і бюджет кроків інтегрування, а також у поєднанні переналаштування розкладу з іншими методами масштабування обчислень під час генерації, описаними побудованою моделлю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1.D. P. Kingma and M. Welling, “Auto-encoding variational bayes”, *arXiv preprint arXiv:1312.6114*, 2013. DOI: [10.48550/arXiv.1312.6114](https://doi.org/10.48550/arXiv.1312.6114)
- 2.D. Rezende and S. Mohamed, “Variational inference with normalizing flows”, in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, PMLR, 2015, pp. 1530–1538. [Online]. Available: <https://proceedings.mlr.press/v37/rezende15.html>
- 3.G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, “Normalizing flows for probabilistic modeling and inference”, *Journal of Machine Learning Research*, vol. 22, no. 57, pp. 1–64, 2021. [Online]. Available: <http://jmlr.org/papers/v22/19-1028.html>
- 4.I. J. Goodfellow et al., “Generative adversarial nets”, in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, ser. NIPS’14, vol. 27, MIT Press, 2014, pp. 2672–2680. DOI: [10.48550/arXiv.1406.2661](https://doi.org/10.48550/arXiv.1406.2661)
- 5.M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks”, in *International conference on machine learning*, PMLR, 2017, pp. 214–223. [Online]. Available: <https://proceedings.mlr.press/v70/arjovsky17a.html>
- 6.A. van den Oord, N. Kalchbrenner, L. Espeholt, k. kavukcuoglu koray, O. Vinyals, and A. Graves, “Conditional image generation with pixcnn decoders”, in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/b1301141feffabac455e1f90a7de2054-Paper.pdf
- 7.J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models”, *Advances in neural information processing systems*, vol. 33, pp. 6840–6851,

2020. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf>
- 8.Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution”, *Advances in neural information processing systems*, vol. 32, 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf
 - 9.Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations”, in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/pdf?id=PXTIG12RRHS>
 - 10.P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis”, *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021. [Online]. Available: <https://proceedings.nips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf>
 - 11.T. Karras, M. Aittala, T. Aila, and S. Laine, “Elucidating the design space of diffusion-based generative models”, *Advances in neural information processing systems*, vol. 35, pp. 26 565–26 577, 2022. DOI: [10.52202/068431-1926](https://doi.org/10.52202/068431-1926)
 - 12.Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow matching for generative modeling”, *arXiv preprint arXiv:2210.02747*, 2022. DOI: [10.48550/arXiv.2210.02747](https://doi.org/10.48550/arXiv.2210.02747)
 - 13.X. Liu, C. Gong, and Q. Liu, “Flow straight and fast: Learning to generate and transfer data with rectified flow”, *arXiv preprint arXiv:2209.03003*, 2022. DOI: [10.48550/arXiv.2209.03003](https://doi.org/10.48550/arXiv.2209.03003)
 - 14.M. S. Albergo and E. Vanden-Eijnden, “Building normalizing flows with stochastic interpolants”, *arXiv preprint arXiv:2209.15571*, 2022. DOI: [10.48550/arXiv.2209.15571](https://doi.org/10.48550/arXiv.2209.15571)
 - 15.P. Esser et al., “Scaling rectified flow transformers for high-resolution image synthesis”, in *Proceedings of the 41st International Conference on Machine*

- Learning*, ser. ICML'24, Vienna, Austria, 2024, pp. 12 606–12 633. DOI: [10.48550/arXiv.2403.03206](https://doi.org/10.48550/arXiv.2403.03206)
- 16.S. Batifol et al., “Flux. 1 kontekst: Flow matching for in-context image generation and editing in latent space”, *arXiv preprint arXiv:2506.15742*, 2025. DOI: [10.48550/arXiv.2506.15742](https://doi.org/10.48550/arXiv.2506.15742)
 - 17.V. Bondar and V. Babenko, “Investigation of the main aspects of the generative models applications in machine learning”, *Herald of Khmelnytskyi National University. Technical sciences*, vol. 347, no. 1, pp. 69–72, 2025, ISSN: 2307-5732. DOI: [10.31891/2307-5732-2025-347-8](https://doi.org/10.31891/2307-5732-2025-347-8) [Online]. Available: <https://heraldts.khmnpu.edu.ua/index.php/heraldts/article/view/1026>
 - 18.V. Bondar and V. Babenko, “Comparative analysis of adversarial-diffusion hybrid architectures of generative models”, *Infocommunication and computer technologies*, vol. 1, no. 09, pp. 82–88, Jul. 2025, ISSN: 2788-5518. DOI: [10.36994/2788-5518-2025-01-09-10](https://doi.org/10.36994/2788-5518-2025-01-09-10) [Online]. Available: <https://visn-icct.uu.edu.ua/index.php/icct/article/view/180>
 - 19.V. Bondar, “The main types of models in machine learning”, in *Scientific Research Methodology – 2024: Ukrainian Scientific and Practical Conference*, (Cherkasy), Cherkasy: Cherkasy State Technological University, Nov. 2024, pp. 88–90. [Online]. Available: https://er.chdtu.edu.ua/bitstream/ChSTU/5470/1/Proceedings.%20SRM_2024.pdf
 - 20.V. Bondar and V. Babenko, “Analysis and survey on the specifics of application of hybrid adversarial-diffusion generative models”, in *Proceedings of XIII International Scientific and Practical Conference “Science in the Modern World: Innovations and Challenges”*, (Toronto), Toronto, Canada, Sep. 2025, pp. 78–87. [Online]. Available: <https://sci-conf.com.ua/wp-content/uploads/2025/09/SCIENCE-IN-THE-MODERN-WORLD-INNOVATIONS-AND-CHALLENGES-4-6.09.25.pdf>
 - 21.V. Bondar, V. Babenko, R. Trembovetskyi, Y. Korobeinyk, and V. Dzyuba, “Deep generative models as the probability transformation functions”, *Com-*

- munications in Computer and Information Science*, vol. 2871, pp. 97–111, 2026, ISSN: 1865-0929. DOI: [10.1007/978-3-032-16808-5_8](https://doi.org/10.1007/978-3-032-16808-5_8) [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-032-16808-5_8
22. В. В. Бондар, В. Г. Бабенко та Д. Є. Козлов, „Масштабування обчислень під час генерації як універсальний принцип для генеративних моделей“, *Телекомунікаційні та Інформаційні Технології*, № 4 (89), с. 229–234, 2025, ISSN: 2412-4338. DOI: [10.31673/2412-4338.2025.048926](https://doi.org/10.31673/2412-4338.2025.048926) url: <https://tit.dut.edu.ua/telecommunication/article/view/2683>
23. V. Bondar, “Inference-time scaling as a universal principle in machine learning: Classification and cross-domain analysis”, *Visnyk of Kherson National Technical University*, no. 1 (96), pp. 200–206, 2026, ISSN: 2078-4481. DOI: [10.35546/kntu2078-4481.2026.1.28](https://doi.org/10.35546/kntu2078-4481.2026.1.28) [Online]. Available: https://journals.kntu.kherson.ua/index.php/visnyk_kntu/article/view/1395
24. В. В. Бондар та В. Г. Бабенко, „Масштабування під час інференсу у дифузійних моделях: аналогія з міркуванням у великих мовних моделях“, в *Матеріали тез доповіді IV Міжнародної науково-практичної інтернет-конференції ”Інновації та перспективні шляхи розвитку інформаційних технологій”(ПІШПІТ-2025)*, (Черкаси), Черкаси, Україна: Черкаський державний технологічний університет, листоп. 2025, с. 95–97. url: https://drive.google.com/file/d/1vfK7HzALRZHFTE8SKi6P_c-D3X4K3YPK/view
25. V. Bondar, S. Semenov, V. Babenko, and D. Holovniak, “Flowecg: Using flow matching to create a more efficient ecg signal generator”, *Lecture Notes in Electrical Engineering*, vol. 1570, pp. 310–319, 2026, ISSN: 1876-1100. DOI: [10.1007/978-3-032-18415-3_32](https://doi.org/10.1007/978-3-032-18415-3_32) [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-032-18415-3_32
26. V. Bondar, “Velocity scheduled flow matching”, *arXiv preprint arXiv:2607.11442*, 2026. DOI: [10.48550/arXiv.2607.11442](https://doi.org/10.48550/arXiv.2607.11442)
27. A. Y. Ng and M. I. Jordan, “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes”, in *Proceedings of the 15th*

- International Conference on Neural Information Processing Systems: Natural and Synthetic*, ser. NIPS'01, Vancouver, British Columbia, Canada: MIT Press, 2001, pp. 841–848. [Online]. Available: <https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf>
- 28.K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012, p. 1096, ISBN: 978-0-262-01802-9.
 - 29.Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013. [Online]. Available: [10.1109/TPAMI.2013.50](https://doi.org/10.1109/TPAMI.2013.50)
 - 30.P. Eigenschink, T. Reutterer, S. Vamosi, R. Vamosi, C. Sun, and K. Kalcher, “Deep generative models for synthetic data: A survey”, *IEEE Access*, vol. 11, pp. 47 304–47 320, 2023. DOI: [10.1109/ACCESS.2023.3275134](https://doi.org/10.1109/ACCESS.2023.3275134)
 - 31.L. Yang et al., “Diffusion models: A comprehensive survey of methods and applications”, English, *ACM Computing Surveys*, vol. 56, no. 4, Apr. 2024, Publisher Copyright: © 2023 held by the owner/author(s). Publication rights licensed to ACM., ISSN: 0360-0300. DOI: [10.1145/3626235](https://doi.org/10.1145/3626235)
 - 32.Y. Chen et al., “Generative adversarial networks in medical image augmentation: A review”, *Computers in Biology and Medicine*, vol. 144, p. 105 382, 2022, ISSN: 0010-4825. DOI: [10.1016/j.combiomed.2022.105382](https://doi.org/10.1016/j.combiomed.2022.105382)
 - 33.C. Pang, J. Qiao, X. Zeng, Q. Zou ta L. Wei, „Deep generative models in de novo drug molecule generation“, *Journal of Chemical Information and Modeling*, т. 64, № 7, с. 2174—2194, 2023. DOI: [10.1021/acs.jcim.3c01496](https://doi.org/10.1021/acs.jcim.3c01496)
 - 34.X. Tang et al., “A survey of generative ai for de novo drug design: New frontiers in molecule and protein generation”, *Briefings in Bioinformatics*, vol. 25, no. 4, bbae338, 2024. DOI: [10.1093/bib/bbae338](https://doi.org/10.1093/bib/bbae338)
 - 35.K. Zhang et al., *Generative artificial intelligence in robotic manipulation: A survey*, 2025. DOI: [10.48550/arXiv.2503.03464](https://doi.org/10.48550/arXiv.2503.03464)

- 36.L. Ericson et al., “Deep generative modeling for financial time series with application in var: A comparative review”, *arXiv preprint arXiv:2401.10370*, 2024. DOI: [10.48550/arXiv.2401.10370](https://doi.org/10.48550/arXiv.2401.10370)
- 37.T. Brown et al., “Language models are few-shot learners”, *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf
- 38.J. Betker et al., “Improving image generation with better captions”, 2023. [Online]. Available: <https://cdn.openai.com/papers/dall-e-3.pdf>
- 39.A. van den Oord et al., “Wavenet: A generative model for raw audio”, in *Speech Synthesis Workshop*, vol. 9, 2016. [Online]. Available: https://www.isca-archive.org/ssw_2016/vandenoord16_ssw.pdf
- 40.M. Quiroz-Juárez, O. Jiménez-Ramírez, R. Vázquez-Medina, V. Breña-Medina, J. Aragón, and R. Barrio, “Generation of ecg signals from a reaction-diffusion model spatially discretized”, *Scientific reports*, vol. 9, no. 1, p. 19 000, 2019. DOI: [10.1038/s41598-019-55448-5](https://doi.org/10.1038/s41598-019-55448-5)
- 41.C. Donahue, J. McAuley, and M. Puckette, “Adversarial audio synthesis”, in *International Conference on Learning Representations*, 2018. [Online]. Available: <https://openreview.net/pdf?id=ByMVTsR5KQ>
- 42.V. Thambawita et al., “Deepfake electrocardiograms using generative adversarial networks are the beginning of the end for privacy issues in medicine”, *Scientific reports*, vol. 11, no. 1, p. 21 896, 2021, ISSN: 2045-2322. DOI: [10.1038/s41598-021-01295-2](https://doi.org/10.1038/s41598-021-01295-2)
- 43.D. Hazra and Y.-C. Byun, “Synsiggan: Generative adversarial networks for synthetic biomedical signal generation”, *Biology*, vol. 9, no. 12, p. 441, 2020, ISSN: 2079-7737. DOI: [10.3390/biology9120441](https://doi.org/10.3390/biology9120441)
- 44.Y. Xia, Y. Xu, P. Chen, J. Zhang, and Y. Zhang, “Generative adversarial network with transformer generator for boosting ecg classification”, *Biomedical Signal*

- Processing and Control*, vol. 80, p. 104 276, 2023, ISSN: 1746-8094. DOI: [10.1016/j.bspc.2022.104276](https://doi.org/10.1016/j.bspc.2022.104276)
- 45.J. M. L. Alcaraz and N. Strodthoff, “Diffusion-based conditional ecg generation with structured state space models”, *Computers in biology and medicine*, vol. 163, p. 107 115, 2023, ISSN: 0010-4825. DOI: [10.1016/j.combiomed.2023.107115](https://doi.org/10.1016/j.combiomed.2023.107115)
 - 46.N. Neifar, A. Ben-Hamadou, A. Mdhaftar, and M. Jmaiel, “Diffecg: A versatile probabilistic diffusion model for ecg signals synthesis”, in *2024 IEEE/ACIS 22nd International Conference on Software Engineering Research, Management and Applications (SERA)*, 2024, pp. 182–188. DOI: [10.1109/SERA61261.2024.10685651](https://doi.org/10.1109/SERA61261.2024.10685651)
 - 47.M. H. Zama and F. Schwenker, “Ecg synthesis via diffusion-based state space augmented transformer”, *Sensors*, vol. 23, no. 19, p. 8328, 2023, ISSN: 1424-8220. DOI: [10.3390/s23198328](https://doi.org/10.3390/s23198328)
 - 48.X. Li, M. Sakevych, G. Atkinson, and V. Metsis, “Biodiffusion: A versatile diffusion model for biomedical signal synthesis”, *Bioengineering*, vol. 11, no. 4, p. 299, 2024, ISSN: 2306-5354. DOI: [10.3390/bioengineering11040299](https://doi.org/10.3390/bioengineering11040299)
 - 49.Z. Wang, Q. She, and T. E. Ward, “Generative adversarial networks in computer vision: A survey and taxonomy”, *ACM Comput. Surv.*, vol. 54, no. 2, Feb. 2021, ISSN: 0360-0300. DOI: [10.1145/3439723](https://doi.org/10.1145/3439723)
 - 50.G. Harshvardhan, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, “A comprehensive survey and analysis of generative models in machine learning”, *Computer Science Review*, vol. 38, p. 100 285, 2020. DOI: [10.1016/j.cosrev.2020.100285](https://doi.org/10.1016/j.cosrev.2020.100285)
 - 51.D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors”, *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. DOI: [10.1038/323533a0](https://doi.org/10.1038/323533a0)
 - 52.Y. Le Cun and F. Fogelman-Soulié, “Modèles connexionnistes de l’apprentissage”, *Intellectica*, vol. 2, no. 1, pp. 114–143, 1987. [Online]. Available: https://www.persee.fr/doc/intel_0769-4113_1987_num_2_1_1804

- 53.H. Bourlard and Y. Kamp, “Auto-association by multilayer perceptrons and singular value decomposition”, *Biological cybernetics*, vol. 59, no. 4, pp. 291–294, 1988. DOI: [10.1007/BF00332918](https://doi.org/10.1007/BF00332918)
- 54.G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks”, *Science*, vol. 313, no. 5786, pp. 504–507, 2006. DOI: [10.1126/science.1127647](https://doi.org/10.1126/science.1127647)
- 55.K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009. DOI: [10.1109/CVPR52688.2022.01553](https://doi.org/10.1109/CVPR52688.2022.01553)
- 56.A. Van Den Oord, O. Vinyals, et al., “Neural discrete representation learning”, *Advances in neural information processing systems*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf
- 57.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*, 2022, pp. 10 684–10 695. DOI: <https://doi.org/10.1109/cvpr52688.2022.01042>
- 58.A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., “Improving language understanding by generative pre-training”, *OpenAI Blog*, 2018. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf
- 59.J. Achiam et al., “Gpt-4 technical report”, *arXiv preprint arXiv:2303.08774*, 2023. DOI: [10.48550/arXiv.2303.08774](https://doi.org/10.48550/arXiv.2303.08774)
- 60.G. Team et al., “Gemini: A family of highly capable multimodal models”, *arXiv preprint arXiv:2312.11805*, 2023. DOI: [10.48550/arXiv.2312.11805](https://doi.org/10.48550/arXiv.2312.11805)
- 61.A. Grattafiori et al., “The llama 3 herd of models”, *arXiv preprint arXiv:2407.21783*, 2024. DOI: [10.48550/arXiv.2407.21783](https://doi.org/10.48550/arXiv.2407.21783)

- 62.A. Yang et al., “Qwen3 technical report”, *arXiv preprint arXiv:2505.09388*, 2025. DOI: [10.48550/arXiv.2505.09388](https://doi.org/10.48550/arXiv.2505.09388)
- 63.A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel recurrent neural networks”, in *International conference on machine learning*, PMLR, 2016, pp. 1747–1756. [Online]. Available: <https://proceedings.mlr.press/v48/oord16.pdf>
- 64.C. B. Vennerød, A. Kjærran, and E. S. Bugge, “Long short-term memory rnn”, *arXiv preprint arXiv:2105.06756*, 2021. DOI: [10.48550/arXiv.2105.06756](https://doi.org/10.48550/arXiv.2105.06756)
- 65.A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al., “Conditional image generation with pixelcnn decoders”, *Advances in neural information processing systems*, vol. 29, 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/b1301141feffabac455e1f90a7de2054-Paper.pdf
- 66.T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, “Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications”, *arXiv preprint arXiv:1701.05517*, 2017. DOI: [10.48550/arXiv.1701.05517](https://doi.org/10.48550/arXiv.1701.05517)
- 67.P. Esser, R. Rombach, and B. Ommer, “Taming transformers for high-resolution image synthesis”, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 10, 2020, pp. 12 868–12 878. DOI: [10.1109/cvpr46437.2021.01268](https://doi.org/10.1109/cvpr46437.2021.01268)
- 68.K. Tian, Y. Jiang, Z. Yuan, B. PENG, and L. Wang, “Visual autoregressive modeling: Scalable image generation via next-scale prediction”, in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. DOI: [10.52202/079017-2694](https://doi.org/10.52202/079017-2694)
- 69.N. Huang, A. Gokaslan, V. Kuleshov, and J. Tompkin, “The gan is dead; long live the gan! a modern gan baseline”, *Advances in Neural Information Processing Systems*, vol. 37, pp. 44 177–44 215, 2024. DOI: [10.52202/079017-1402](https://doi.org/10.52202/079017-1402)
- 70.I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans”, *Advances in neural information processing*

- systems*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/892c3b1c6dccc52936e27cbd0ff683d6-Paper.pdf
- 71.C. Abou Akar, R. Abdel Massih, A. Yaghi, J. Khalil, M. Kamradt, and A. Makhoul, “Generative adversarial network applications in industry 4.0: A review”, *International Journal of Computer Vision*, vol. 132, no. 6, pp. 2195–2254, 2024. DOI: [10.1007/s11263-023-01966-9](https://doi.org/10.1007/s11263-023-01966-9)
- 72.P. Sharma, M. Kumar, H. K. Sharma, and S. M. Biju, “Generative adversarial networks (gans): Introduction, taxonomy, variants, limitations, and applications”, *Multimedia Tools and Applications*, pp. 1–48, 2024. DOI: [10.1007/s11042-024-18767-y](https://doi.org/10.1007/s11042-024-18767-y)
- 73.L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real nvp”, *arXiv preprint arXiv:1605.08803*, 2016. DOI: [10.48550/arXiv.1605.08803](https://doi.org/10.48550/arXiv.1605.08803)
- 74.R. T. Chen, J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen, “Residual flows for invertible generative modeling”, *Advances in neural information processing systems*, vol. 32, 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/5d0d5594d24f0f955548f0fc0ff83d10-Paper.pdf
- 75.R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations”, *Advances in neural information processing systems*, vol. 31, 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf
- 76.J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics”, in *International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, PMLR, 2015, pp. 2256–2265. [Online]. Available: <https://proceedings.mlr.press/v37/sohl-dickstein15.html>
- 77.Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution”, in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Avail-

able: https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf

- 78.E. Heitz, L. Belcour, and T. Chambon, “Iterative α -(de) blending: A minimalist deterministic diffusion model”, in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH ’23, Association for Computing Machinery, 2023, pp. 1–8, ISBN: 9798400701597. DOI: [10.1145/3588432.3591540](https://doi.org/10.1145/3588432.3591540)
- 79.J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models”, in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/pdf?id=St1giarCHLP>
- 80.C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps”, in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 5775–5787. DOI: [10.52202/068431-0418](https://doi.org/10.52202/068431-0418)
- 81.T. Salimans and J. Ho, “Progressive distillation for fast sampling of diffusion models”, *arXiv preprint arXiv:2202.00512*, 2022. DOI: [10.48550/arXiv.2202.00512](https://doi.org/10.48550/arXiv.2202.00512)
- 82.C. Meng et al., “On distillation of guided diffusion models”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 14 297–14 306. DOI: [10.1109/CVPR52729.2023.01374](https://doi.org/10.1109/CVPR52729.2023.01374)
- 83.A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach, “Adversarial diffusion distillation”, in *European Conference on Computer Vision*, Springer, 2024, pp. 87–103. DOI: [10.1007/978-3-031-73016-0_6](https://doi.org/10.1007/978-3-031-73016-0_6)
- 84.T. Chen, “On the importance of noise scheduling for diffusion models”, *arXiv preprint arXiv:2301.10972*, 2023. DOI: [10.48550/arXiv.2301.10972](https://doi.org/10.48550/arXiv.2301.10972)
- 85.A. Sabour, S. Fidler, and K. Kreis, “Align your steps: Optimizing sampling schedules in diffusion models”, in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML’24, Vienna, Austria: JMLR.org, 2024, pp. 42 947–42 975. [Online]. Available: <https://dl.acm.org/doi/10.5555/3692070.3693820>

- 86.S. Sahoo, A. Gokaslan, C. M. De Sa, and V. Kuleshov, “Diffusion models with learned adaptive noise”, *Advances in Neural Information Processing Systems*, vol. 37, pp. 105 730–105 779, 2024. DOI: [10.52202/079017-3354](https://doi.org/10.52202/079017-3354)
- 87.A. Tong et al., “Improving and generalizing flow-based generative models with minibatch optimal transport”, *Transactions on Machine Learning Research*, 2024, ISSN: 2835-8856. [Online]. Available: <https://openreview.net/pdf?id=CD9Snc73AW>
- 88.A.-A. Pooladian, H. Ben-Hamu, C. Domingo-Enrich, B. Amos, Y. Lipman, and R. T. Q. Chen, “Multisample flow matching: Straightening flows with minibatch couplings”, in *International Conference on Machine Learning*, PMLR, 2023, pp. 28 100–28 127. [Online]. Available: <https://proceedings.mlr.press/v202/pooladian23a/pooladian23a.pdf>
- 89.T. B. Ifriqi, J. Nguyen, K. Alahari, J. Verbeek, and R. T. Chen, “Flowception: Temporally expansive flow matching for video generation”, pp. 16 185–16 195, 2026. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2026/papers/Ifriqi_Flowception_Temporally_Expansive_Flow_Matching_for_Video_Generation_CVPR_2026_paper.pdf
- 90.A. Sauer, F. Boesel, T. Dockhorn, A. Blattmann, P. Esser, and R. Rombach, “Fast high-resolution image synthesis with latent adversarial diffusion distillation”, in *SIGGRAPH Asia 2024 Conference Papers*, ser. SA '24, 2024, pp. 1–11, ISBN: 9798400711312. DOI: [10.1145/3680528.3687625](https://doi.org/10.1145/3680528.3687625)
- 91.S. Lin, A. Wang, and X. Yang, “Sd-xl-lightning: Progressive adversarial diffusion distillation”, *arXiv preprint arXiv:2402.13929*, 2024. DOI: [10.48550/arXiv.2402.13929](https://doi.org/10.48550/arXiv.2402.13929)
- 92.M. Kang et al., “Distilling diffusion models into conditional gans”, in *European Conference on Computer Vision (ECCV)*, Springer Nature Switzerland, 2025, pp. 428–447, ISBN: 978-3-031-73390-1. DOI: [10.1007/978-3-031-73390-1_25](https://doi.org/10.1007/978-3-031-73390-1_25)
- 93.L. Yang, H. Qian, Z. Zhang, J. Liu, and B. Cui, “Structure-guided adversarial training of diffusion models”, in *Proceedings of the IEEE/CVF Conference on*

- Computer Vision and Pattern Recognition*, 2024, pp. 7256–7266. DOI: [10.1109/CVPR52733.2024.00693](https://doi.org/10.1109/CVPR52733.2024.00693)
- 94.D. Park, S. Lee, S. Kim, T. Lee, Y. Hong, and H. J. Kim, “Constant acceleration flow”, *Advances in Neural Information Processing Systems*, vol. 37, pp. 90 030–90 060, 2024. DOI: [10.52202/079017-2859](https://doi.org/10.52202/079017-2859)
 - 95.Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, “Diffusion-gan: Training gans with diffusion”, *arXiv preprint arXiv:2206.02262*, 2022. DOI: [10.48550/arXiv.2206.02262](https://doi.org/10.48550/arXiv.2206.02262)
 - 96.L. T. Trinh and T. Hamagami, “Latent denoising diffusion gan: Faster sampling, higher image quality”, *IEEE Access*, vol. 12, pp. 78 161–78 172, 2024. DOI: [10.1109/ACCESS.2024.3406535](https://doi.org/10.1109/ACCESS.2024.3406535)
 - 97.X. Liu et al., “Ladiffgan: Training gans with diffusion supervision in latent spaces”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2024, pp. 1115–1125, ISBN: 9798350365474. DOI: [10.1109/CVPRW63382.2024.00118](https://doi.org/10.1109/CVPRW63382.2024.00118)
 - 98.T. Hang et al., “Improved noise schedule for diffusion training”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 4796–4806, ISBN: 9798331587758. DOI: [10.1109/ICCV51701.2025.00456](https://doi.org/10.1109/ICCV51701.2025.00456)
 - 99.C. Snell, J. Lee, K. Xu, and A. Kumar, “Scaling llm test-time compute optimally can be more effective than scaling model parameters”, in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 10 131–10 165. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2025/file/1b623663fd9b874366f3ce019fdfdd44-Paper-Conference.pdf
 - 100.B. Brown et al., “Large language monkeys: Scaling inference compute with repeated sampling”, *arXiv preprint arXiv:2407.21787*, 2024. DOI: [10.48550/arXiv.2407.21787](https://doi.org/10.48550/arXiv.2407.21787)
 - 101.Y. Wu, Z. Sun, S. Li, S. Welleck, and Y. Yang, “Inference scaling laws: An empirical analysis of compute-optimal inference for llm problem-solving”,

- in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/pdf?id=VNckp7JEHn>
- 102.J. Kaplan et al., “Scaling laws for neural language models”, *arXiv preprint arXiv:2001.08361*, 2020. DOI: [10.48550/arXiv.2001.08361](https://doi.org/10.48550/arXiv.2001.08361)
 - 103.J. Hoffmann et al., “An empirical analysis of compute-optimal large language model training”, *Advances in neural information processing systems*, vol. 35, pp. 30 016–30 030, 2022. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/c1e2faff6f588870935f114ebe04a3e5-Paper-Conference.pdf
 - 104.S. Welleck et al., “From decoding to meta-generation: Inference-time algorithms for large language models”, *arXiv preprint arXiv:2406.16838*, 2024. DOI: [10.48550/arXiv.2406.16838](https://doi.org/10.48550/arXiv.2406.16838)
 - 105.N. Ma et al., “Scaling inference time compute for diffusion models”, in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 2523–2534, ISBN: 9798331543648. DOI: [10.1109/CVPR52734.2025.00241](https://doi.org/10.1109/CVPR52734.2025.00241)
 - 106.V. Balachandran et al., “Inference-time scaling for complex tasks: Where we stand and what lies ahead”, *arXiv preprint arXiv:2504.00294*, 2025. DOI: [10.48550/arXiv.2504.00294](https://doi.org/10.48550/arXiv.2504.00294)
 - 107.A. Agarwal, A. Sengupta, and T. Chakraborty, “The art of scaling test-time compute for large language models”, *arXiv preprint arXiv:2512.02008*, 2025. DOI: [10.48550/arXiv.2512.02008](https://doi.org/10.48550/arXiv.2512.02008)
 - 108.J. Wei et al., “Chain-of-thought prompting elicits reasoning in large language models”, in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS ’22, New Orleans, LA, USA: Curran Associates Inc., 2022, ISBN: 9781713871088. DOI: [10.52202/068431-1800](https://doi.org/10.52202/068431-1800)
 - 109.X. Wang, J. Wei, D. Schuurmans, Q. Le, E. H. Chi, and D. Zhou, “Self-consistency improves chain of thought reasoning in language models”, *arXiv preprint arXiv:2203.11171*, 2022. DOI: [10.48550/arXiv.2203.11171](https://doi.org/10.48550/arXiv.2203.11171)

- 110.S. Yao et al., “Tree of thoughts: Deliberate problem solving with large language models”, in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS ’23, New Orleans, LA, USA: Curran Associates Inc., 2023. DOI: [10.52202/075280-0517](https://doi.org/10.52202/075280-0517)
- 111.OpenAI Team, “Learning to reason with llms”, *OpenAI Blog*, 2024. [Online]. Available: <https://openai.com/index/learning-to-reason-with-llms/>
- 112.D. Guo et al., “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning”, *arXiv preprint arXiv:2501.12948*, 2025. DOI: [10.48550/arXiv.2501.12948](https://doi.org/10.48550/arXiv.2501.12948)
- 113.Y. Xu, M. Deng, X. Cheng, Y. Tian, Z. Liu, and T. Jaakkola, “Restart sampling for improving generative processes”, *Advances in Neural Information Processing Systems*, vol. 36, pp. 76 806–76 838, 2023. DOI: [10.52202/075280-3356](https://doi.org/10.52202/075280-3356)
- 114.J. Ho and T. Salimans, “Classifier-free diffusion guidance”, *arXiv preprint arXiv:2207.12598*, 2022. DOI: [10.48550/arXiv.2207.12598](https://doi.org/10.48550/arXiv.2207.12598)
- 115.J. Wildberger, M. Dax, S. Buchholz, S. R. Green, J. H. Macke, and B. Schölkopf, “Flow matching for scalable simulation-based inference”, in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS ’23, New Orleans, LA, USA: Curran Associates Inc., 2023, ISBN: 9781713899112. DOI: [10.52202/075280-0737](https://doi.org/10.52202/075280-0737)
- 116.B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis”, in *European conference on computer vision*, Springer, 2020, pp. 405–421, ISBN: 9783030584511. DOI: [10.1007/978-3-030-58452-8_24](https://doi.org/10.1007/978-3-030-58452-8_24)
- 117.Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, “Test-time training with self-supervision for generalization under distribution shifts”, in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML’20, 2020, pp. 9229–9248. [Online]. Available: <https://proceedings.mlr.press/v119/sun20b/sun20b.pdf>

- 118.L. Theis, A. van den Oord, and M. Bethge, “A note on the evaluation of generative models”, in *International Conference on Learning Representations*, 2016. DOI: [10.48550/arXiv.1511.01844](https://arxiv.org/abs/1511.01844)
- 119.A. Borji, “Pros and cons of gan evaluation measures: New developments”, *Computer Vision and Image Understanding*, vol. 215, p. 103 329, 2022. DOI: [10.1016/j.cviu.2021.103329](https://doi.org/10.1016/j.cviu.2021.103329)
- 120.M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, “Assessing generative models via precision and recall”, *Advances in neural information processing systems*, vol. 31, 2018. [Online]. Available: <https://papers.neurips.cc/paper/7769-assessing-generative-models-via-precision-and-recall.pdf>
- 121.M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium”, *Advances in neural information processing systems*, NIPS’17, vol. 30, pp. 6629–6640, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf
- 122.S. Jayasumana, S. Ramalingam, A. Veit, D. Glasner, A. Chakrabarti, and S. Kumar, “Rethinking fid: Towards a better evaluation metric for image generation”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2024, pp. 9307–9315. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2024/html/Jayasumana_Rethinking_FID_Towards_a_Better_Evaluation_Metric_for_Image_Generation_CVPR_2024_paper.html
- 123.G. Stein et al., “Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models”, *Advances in neural information processing systems*, vol. 36, pp. 3732–3784, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/0bc795afae289ed465a65a3b4b1f4eb7-Paper-Conference.pdf
- 124.M. Otani et al., “Toward verifiable and reproducible human evaluation for text-to-image generation”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2023, pp. 14 277–14 286. [Online].

Available: https://openaccess.thecvf.com/content/CVPR2023/papers/Otani_Toward_Verifiable_and_Reproducible_Human_Evaluation_for_Text-to-Image_Generation_CVPR_2023_paper.pdf

- 125.B. Zanchi et al., “Synthetic ecg signals generation: A scoping review”, *Computers in Biology and Medicine*, vol. 184, p. 109 453, 2025, ISSN: 0010-4825. DOI: [10.1016/j.combiomed.2024.109453](https://doi.org/10.1016/j.combiomed.2024.109453)
- 126.N. Strodthoff, P. Wagner, T. Schaeffter, and W. Samek, “Deep learning for ecg analysis: Benchmarks and insights from ptb-xl”, *IEEE journal of biomedical and health informatics*, vol. 25, no. 5, pp. 1519–1528, 2020. DOI: [10.1109/JBHI.2020.3022989](https://doi.org/10.1109/JBHI.2020.3022989)
- 127.G. Wiher, C. Meister, and R. Cotterell, “On decoding strategies for neural text generators”, *Transactions of the Association for Computational Linguistics*, vol. 10, B. Roark and A. Nenkova, Eds., pp. 997–1012, 2022. DOI: [10.1162/tacl_a_00502](https://doi.org/10.1162/tacl_a_00502)
- 128.H. Lightman et al., “Let’s verify step by step”, in *International Conference on Learning Representations*, vol. 2024, 2024, pp. 39 578–39 601. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf
- 129.L. Breiman, “Bagging predictors”, *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996. DOI: [10.1023/A:1018054314350](https://doi.org/10.1023/A:1018054314350)
- 130.L. Breiman, “Random forests”, *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001, ISSN: 0885-6125. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324)
- 131.A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, in *Advances in neural information processing systems*, vol. 25, 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- 132.Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”, in *International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48,

- PMLR, 2016, pp. 1050–1059. [Online]. Available: <https://proceedings.mlr.press/v48/gal16.html>
- 133.I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks”, in *Advances in neural information processing systems*, vol. 27, 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5a18e133cbf9f257297f410bb7eca942-Paper.pdf
 - 134.D. Silver et al., “Mastering the game of go with deep neural networks and tree search”, *Nature*, vol. 529, no. 7587, pp. 484–489, 2016, ISSN: 00280836. DOI: [10.1038/nature16961](https://doi.org/10.1038/nature16961)
 - 135.D. Silver et al., “A general reinforcement learning algorithm that masters chess, shogi, and go through self-play”, *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018. DOI: [10.1126/science.aar6404](https://doi.org/10.1126/science.aar6404)
 - 136.J. Schrittwieser et al., “Mastering atari, go, chess and shogi by planning with a learned model”, *Nature*, vol. 588, no. 7839, pp. 604–609, 2020, ISSN: 00280836. DOI: [10.1038/s41586-020-03051-4](https://doi.org/10.1038/s41586-020-03051-4)
 - 137.S. Hao et al., “Reasoning with language model is planning with world model”, in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 8154–8173. DOI: [10.18653/v1/2023.emnlp-main.507](https://doi.org/10.18653/v1/2023.emnlp-main.507)
 - 138.A. Zhou, K. Yan, M. Shlapentokh-Rothman, H. Wang, and Y.-X. Wang, “Language agent tree search unifies reasoning acting and planning in language models”, *arXiv preprint arXiv:2310.04406*, 2023. DOI: [10.48550/arXiv.2310.04406](https://doi.org/10.48550/arXiv.2310.04406)
 - 139.M. Besta et al., “Graph of thoughts: Solving elaborate problems with large language models”, in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, AAAI Press, 2024, pp. 17 682–17 690. DOI: [10.1609/aaai.v38i16.29720](https://doi.org/10.1609/aaai.v38i16.29720)
 - 140.P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths”, *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968, ISSN: 05361567. DOI: [10.1109/TSSC.1968.300136](https://doi.org/10.1109/TSSC.1968.300136)

- 141.D. Wang, E. Shelhamer, S. Liu, B. A. Olshausen, and T. Darrell, “Tent: Fully test-time adaptation by entropy minimization”, in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/pdf?id=uXl3bZLkr3c>
- 142.C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks”, in *International conference on machine learning*, PMLR, 2017, pp. 1126–1135. [Online]. Available: <https://proceedings.mlr.press/v70/finn17a/finn17a.pdf>
- 143.P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive nlp tasks”, *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf>
- 144.A. Graves, “Adaptive computation time for recurrent neural networks”, *arXiv preprint arXiv:1603.08983*, 2016. DOI: [10.48550/arXiv.1603.08983](https://doi.org/10.48550/arXiv.1603.08983)
- 145.N. Shazeer et al., “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer”, *arXiv preprint arXiv:1701.06538*, 2017. DOI: [10.48550/arXiv.1701.06538](https://doi.org/10.48550/arXiv.1701.06538)
- 146.Y. Leviathan, M. Kalman, and Y. Matias, “Fast inference from transformers via speculative decoding”, *arXiv preprint arXiv:2211.17192*, 2022. DOI: [10.48550/arXiv.2211.17192](https://doi.org/10.48550/arXiv.2211.17192)
- 147.N. Muennighoff et al., “S1: Simple test-time scaling”, in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2025, pp. 20 275–20 321, ISBN: 979-8-89176-332-6. DOI: [10.18653/v1/2025.emnlp-main.1025](https://doi.org/10.18653/v1/2025.emnlp-main.1025)
- 148.J. Jumper et al., “Highly accurate protein structure prediction with alphafold”, *Nature*, vol. 596, no. 7873, pp. 583–589, 2021, ISSN: 00280836. DOI: [10.1038/s41586-021-03819-2](https://doi.org/10.1038/s41586-021-03819-2)

- 149.Y. Li et al., “Competition-level code generation with alphacode”, *Science*, vol. 378, no. 6624, pp. 1092–1097, 2022, ISSN: 00368075. DOI: [10.1126/science.abq1158](https://doi.org/10.1126/science.abq1158)
- 150.N. Shaul, R. T. Chen, M. Nickel, M. Le, and Y. Lipman, “On kinetic optimal probability paths for generative models”, in *International Conference on Machine Learning*, PMLR, 2023, pp. 30 883–30 907. [Online]. Available: <https://proceedings.mlr.press/v202/shaul23a/shaul23a.pdf>
- 151.A. Q. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models”, in *International conference on machine learning*, PMLR, 2021, pp. 8162–8171. [Online]. Available: <https://proceedings.mlr.press/v139/nichol21a/nichol21a.pdf>
- 152.T. Flash and N. Hogan, “The coordination of arm movements: An experimentally confirmed mathematical model”, *Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985, ISSN: 0270-6474. DOI: [10.1523/JNEUROSCI.05-07-01688.1985](https://doi.org/10.1523/JNEUROSCI.05-07-01688.1985)
- 153.J. J. Craig, *Introduction to robotics: mechanics and control*, 3rd ed. Pearson Education India, 2009, p. 408, ISBN: 8131718360.
- 154.E. Hairer, S. Norsett ta G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*. сiч. 1993, т. 8, ISBN: 978-3-540-56670-0. DOI: [10.1007/978-3-540-78862-1](https://doi.org/10.1007/978-3-540-78862-1)
- 155.A. Krizhevsky, G. Hinton, et al., “Learning multiple layers of features from tiny images”, 2009. [Online]. Available: <https://cave.cs.toronto.edu/kriz/learning-features-2009-TR.pdf>
- 156.C. Saharia et al., “Photorealistic text-to-image diffusion models with deep language understanding”, *Advances in neural information processing systems*, vol. 35, pp. 36 479–36 494, 2022. DOI: [10.52202/068431-2643](https://doi.org/10.52202/068431-2643)
- 157.J. Yu et al., “Scaling autoregressive models for content-rich text-to-image generation”, *Transactions on Machine Learning Research*, 2022, Featured Certification, ISSN: 2835-8856. DOI: [10.48550/arXiv.2206.10789](https://doi.org/10.48550/arXiv.2206.10789)

- 158.P. Wagner et al., “Ptb-xl, a large publicly available electrocardiography dataset”, *Scientific data*, vol. 7, no. 1, pp. 1–15, 2020, ISSN: 2052-4463. DOI: [10.1038/s41597-020-0495-6](https://doi.org/10.1038/s41597-020-0495-6)
- 159.C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826, ISBN: 9781467388511. DOI: [10.1109/CVPR.2016.308](https://doi.org/10.1109/CVPR.2016.308)
- 160.J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database”, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. DOI: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848)
- 161.J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, “Clipscore: A reference-free evaluation metric for image captioning”, in *Proceedings of the 2021 conference on empirical methods in natural language processing*, 2021, pp. 7514–7528. DOI: [10.18653/v1/2021.emnlp-main.595](https://doi.org/10.18653/v1/2021.emnlp-main.595)
- 162.A. Radford et al., “Learning transferable visual models from natural language supervision”, in *International conference on machine learning*, PmLR, 2021, pp. 8748–8763. [Online]. Available: <https://proceedings.mlr.press/v139/radford21a/radford21a.pdf>
- 163.A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test”, *Journal of Machine Learning Research*, vol. 13, pp. 723–773, 2012, ISSN: 1532-4435. [Online]. Available: <https://www.jmlr.org/papers/volume13/gretton12a/gretton12a.pdf>
- 164.K. Wang, B. Yong, X. Gu, P. Xiao, and X. Zhang, “Spectral similarity measure using frequency spectrum for hyperspectral image classification”, *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 1, pp. 130–134, 2015. DOI: [10.1109/LGRS.2014.2329183](https://doi.org/10.1109/LGRS.2014.2329183)
- 165.Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “Diffwave: A versatile diffusion model for audio synthesis”, *arXiv preprint arXiv:2009.09761*, 2020. DOI: [10.48550/arXiv.2009.09761](https://doi.org/10.48550/arXiv.2009.09761)

- 166.J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance”, *Journal of Big Data*, vol. 6, no. 1, p. 27, Mar. 19, 2019, issn: 2196-1115. DOI: [10.1186/s40537-019-0192-5](https://doi.org/10.1186/s40537-019-0192-5)

Додаток А

Візуальне порівняння згенерованих зображень за різних профілів швидкості

У цьому додатку наведено візуальні порівняння зображень, згенерованих переднавченими моделями за однакових текстових запитів і початкового шуму, але за різних профілів швидкості на етапі інференсу. Кожна ілюстрація містить сітку: стовпці – профілі швидкості (лінійний базовий, плавний вхід-вихід, мінімальний ривок), рядки – бюджети кроків інтегрування (16 і 24 NFE). Порівняння доповнює кількісні результати підрозділу 4.2.4: за насиченого та практично однакового CLIPScore симетричні профілі дають візуально чіткіші й деталізованіші зображення, тоді як лінійний і гальмівні профілі видимого покращення не показують.



Рис. А.1. Текстовий запит: “A detailed studio photograph of a ceramic teapot shaped like a lighthouse, soft natural light, neutral background”

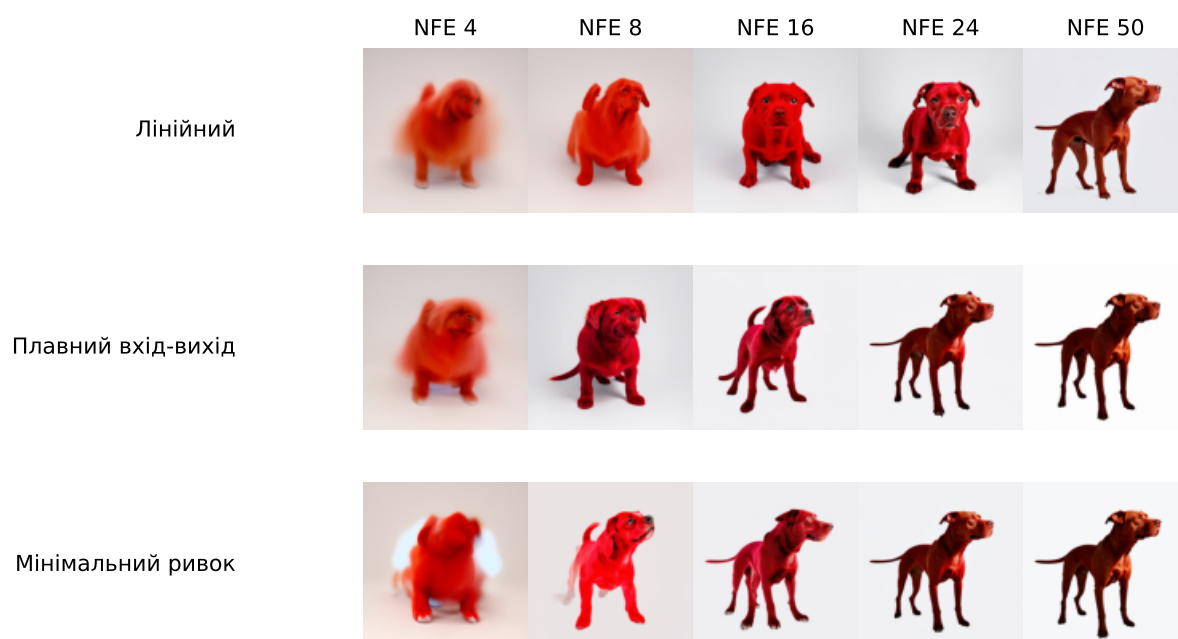


Рис. А.2. Текстовий запит: “A red colored dog”

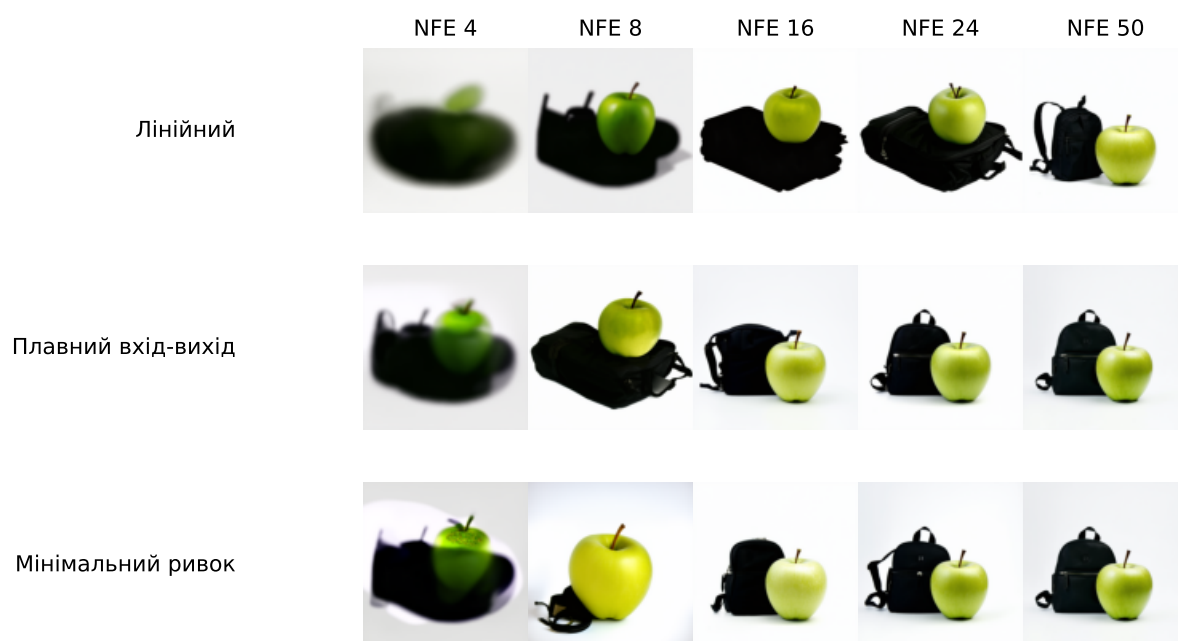


Рис. А.3. Текстовий запит: “A green apple and a black backpack”



Рис. А.4. Текстовий запит: “A panda making latte art”

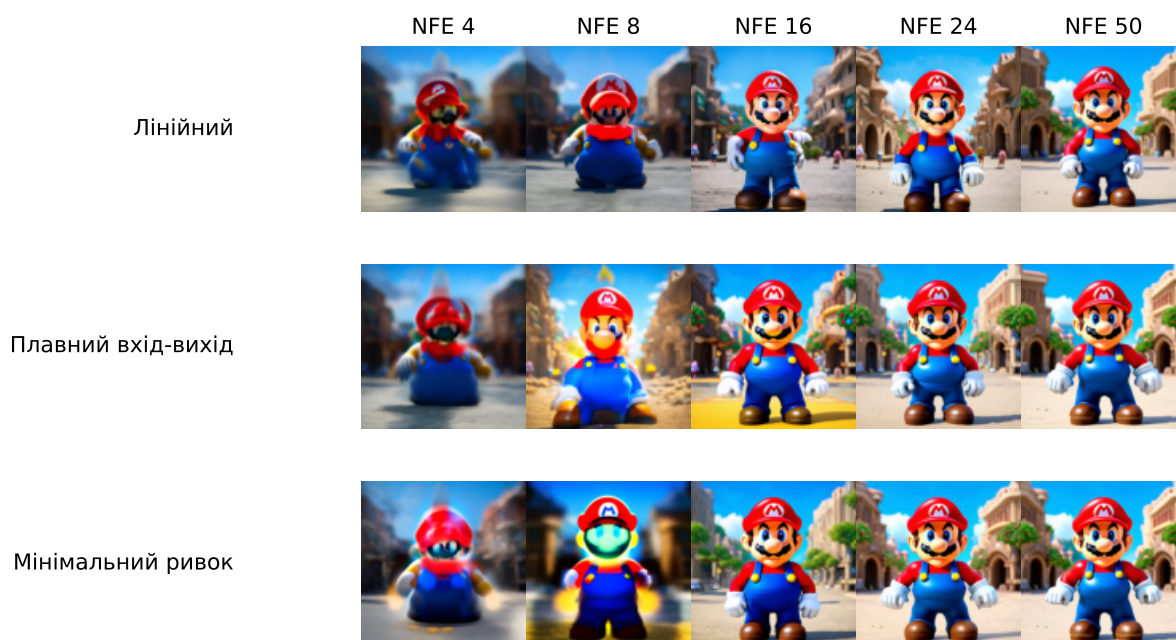


Рис. А.5. Текстовий запит: “A real life photography of super mario, 8k Ultra HD”



Рис. А.6. Текстовий запит: “A photo of an astronaut riding horse”

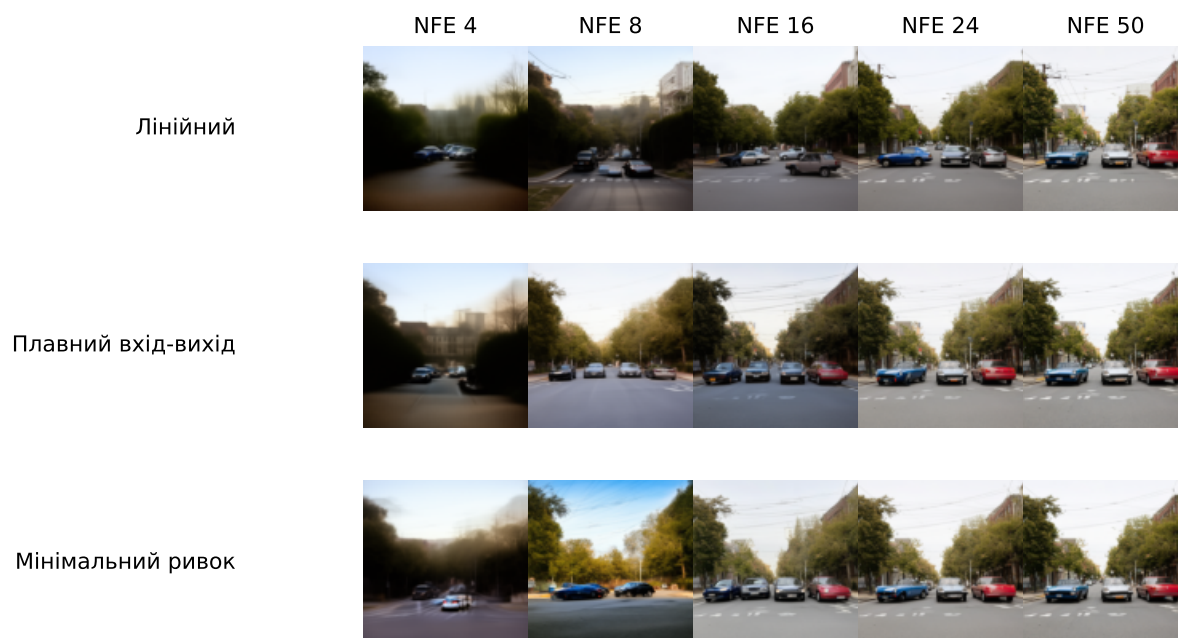


Рис. А.7. Текстовий запит: “Three cars on the street”

«ЗАТВЕРДЖУЮ»Перший проректор Черкаського
державного технологічного
університету

Артем ГОНЧАРОВ

_____ 2026 р.

АКТ**впровадження результатів дисертаційної роботи
Бондаря Віталія Володимировича в навчальний процес кафедри
статистики та прикладної математики
Черкаського державного технологічного університету**

Комісія у складі: кандидата технічних наук, завідувача кафедри статистики та прикладної математики Карапетян Анаїт Радіківни, доктора технічних наук, професора кафедри статистики та прикладної математики Федорова Євгена Євгеновича, кандидата технічних наук, доцента кафедри статистики та прикладної математики Мірошкіної Ірини Володимирівни, розглянувши матеріали дисертаційного дослідження Бондаря Віталія Володимировича, встановила наступне:

1. На лекційних і лабораторних заняттях освітніх компонентів «Нейронні мережі і комп'ютерний зір», «Обробка та аналіз великих даних» для здобувачів освіти першого (бакалаврського) рівня вищої освіти спеціальності 112 «Статистика», освітньої програми «Аналіз даних (DATA SCIENCE) та комп'ютерна статистика» здійснювалась апробація результатів дослідження шляхом доповнення лекційного матеріалу та завдань до лабораторних робіт. У межах практичної підготовки на основі матеріалів дисертаційного дослідження В.В.Бондаря розроблено лабораторні роботи, спрямовані на аналіз великих наборів даних, побудову генеративних моделей та оцінювання якості даних. Це забезпечило формування у здобувачів практичних навичок створення й оцінювання моделей генерації зображень, компетентності з обробки, аналізу та моделювання даних
2. При підготовці магістрів за спеціальністю Е8 «Статистика», освітньої програми «Аналіз даних (DATA SCIENCE) та комп'ютерна статистика» в курсі лекцій з дисципліни «Огляд глибоких генеративних

моделей» використовуються результати дисертаційного дослідження, а саме:

- теоретичний фреймворк, що об'єднує основні класи глибоких генеративних моделей як функції трансформації розподілів імовірності в єдину концептуальну систему, незалежну від архітектурних особливостей конкретних моделей;
- концепція масштабування під час інференсу як універсального механізму підвищення якості генерації, що включає опис структурних компонентів цього механізму та систематизацію шести основних типів, які утворюють єдину основу для порівняльного аналізу підходів у різних галузях машинного навчання.

Наведені наукові результати можуть бути використані при виконанні кваліфікаційних робіт магістрів за спеціальністю Е8 «Статистика».

Результати апробації та впровадження дослідження В.В.Бондаря обговорено та схвалено на засіданні кафедри статистики та прикладної математики (протокол № 10 від 02 березня 2026 року).

Завідувач кафедри СПМ, к.т.н., доц.



Анаїт КАРАПЕТЯН

Професор кафедри СПМ, д.т.н., проф.



Євген ФЕДОРОВ

Доцент кафедри СПМ, к.т.н., доц.



Ірина МІРОШКІНА

«ЗАТВЕРДЖУЮ»

Перший проректор Черкаського
державного технологічного
університету



Артем ГОНЧАРОВ
_____ 2026 р.

АКТ

впровадження результатів дисертаційної роботи Бондаря Віталія Володимировича в навчальний процес кафедри інформаційної безпеки та комп'ютерної інженерії Черкаського державного технологічного університету

Комісія у складі: кандидата технічних наук, доцента, завідувача кафедри інформаційної безпеки та комп'ютерної інженерії Лавданського Артема Олександровича, доктора технічних наук, професора кафедри інформаційної безпеки та комп'ютерної інженерії Бабенко Віри Григорівни, кандидата технічних наук, доцента кафедри інформаційної безпеки та комп'ютерної інженерії Миронюк Тетяни Василівни, розглянувши матеріали дисертаційного дослідження Бондаря Віталія Володимировича, встановила наступне:

При підготовці здобувачів освіти першого (бакалаврського) рівня вищої освіти галузі знань 12 «Інформаційні технології» на лекційних і лабораторних заняттях навчальної дисципліни вільного вибору циклу професійної підготовки «Нейромережі на основі фреймворку PyTorch» здійснювалась апробація та впровадження результатів дослідження шляхом доповнення лекційного матеріалу та розробки лабораторних робіт із практичної реалізації нейромережевих генеративних моделей засобами фреймворку PyTorch. При розробці лабораторних робіт вказаної навчальної дисципліни використані результати дисертаційного дослідження, що спрямовані на побудову й навчання моделей узгодження потоків, чисельне інтегрування траєкторій генерації та оцінювання якості згенерованих об'єктів за метриками FID і CLIPScore, а саме:

- удосконалений метод узгодження потоків із нелінійними профілями швидкості вздовж траєкторій перетворення розподілів, зокрема спосіб переналаштування розкладу чисельного інтегрування вже навченої моделі на етапі інференсу без повторного навчання, що дає змогу

скоротити кількість звернень до нейронної мережі та знизити обчислювальну вартість генерації без зміни архітектури моделі;

- застосування генеративних моделей для розширення навчальних наборів синтетичними даними в задачах, де збір реальних даних ускладнений.

Доповнення навчальної дисципліни вільного вибору циклу професійної підготовки «Нейромережі на основі фреймворку PyTorch» матеріалами дисертаційного дослідження Бондаря Віталія Володимировича та використання отриманих ним наукових результатів забезпечило формування у здобувачів вищої освіти першого (бакалаврського) рівня вищої освіти галузі знань 12 «Інформаційні технології» практичних навичок реалізації, навчання та оптимізації нейромережевих генеративних моделей засобами фреймворку PyTorch, а також компетентностей з ефективного використання обчислювальних ресурсів під час генерації даних.

Результати апробації та впровадження дослідження Бондаря В.В. обговорено та схвалено на засіданні кафедри інформаційної безпеки та комп'ютерної інженерії (протокол № 2 від 29.06.2026 року).

Завідувач кафедри ІБ та КІ, к.т.н., доц.  Артем ЛАВДАНСЬКИЙ

Професор кафедри ІБ та КІ, д.т.н., проф.  Віра БАБЕНКО

Доцент кафедри ІБ та КІ, к.т.н., доц.  Тетяна МИРОНЮК



УКРАЇНА
МІНІСТЕРСТВО ОСВІТИ І НАУКИ
УКРАЇНИ

**ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ імені БОГДАНА
ХМЕЛЬНИЦЬКОГО**

бульвар Шевченка, 81, м. Черкаси, 18031
тел./факс: (0472) 354463, 372142
e-mail: cic@cdu.edu.ua

UKRAINE
MINISTRY OF EDUCATION AND
SCIENCE OF UKRAINE

**BONDAN KHMELNYTSKY NATIONAL
UNIVERSITY of CHERKASY**

81, Shevchenko blvd., Cherkasy, Ukraine, 18031
Phone/Fax: +380(472)354463, 372142
e-mail: cic@cdu.edu.ua

17.09.2026 № 400/04

на № _____

ДОВІДКА

про впровадження результатів дисертаційної праці

Бондаря Віталія Володимировича

**на тему «Моделі та метод нейромережевої генерації об'єктів на основі профілів швидкості
узгодження потоків»,**

поданої на здобуття ступеня доктора філософії зі спеціальності 123 Комп'ютерна інженерія

В освітню практику Черкаського національного університету імені Богдана Хмельницького на базі кафедри прикладної математики та інформатики протягом 2025-2026 навчального року впроваджено результати дисертаційної праці В. В. Бондаря на тему «Моделі та метод нейромережевої генерації об'єктів на основі профілів швидкості узгодження потоків». Роботу виконано на базі кафедри інформаційної безпеки та комп'ютерної інженерії Черкаського державного технологічного університету.

Окремі аспекти напрацювань пошукача були використані в освітньому процесі, зокрема, в лекційних курсах та на практичних заняттях при викладанні дисциплін «Теорія ймовірностей та математична статистика», «Алгебра та геометрія» (для підготовки фахівців освітнього ступеня «бакалавр» спеціальностей F1 – Прикладна математика (ОП «Математичні методи штучного інтелекту») та F6 – Інформаційні системи і технології (ОП «Інтелектуальний аналіз даних»)).

Актуальність обраної дисертантом теми, теоретичне та практичне значення проведеного дослідження мають важливе значення для вдосконалення математичної підготовки здобувачів вищої освіти та розширення практичного застосування математичних методів у сучасних інформаційних технологіях, штучному інтелекті та машинному навчанні. Основні науково-методичні засади та практико-орієнтовані аспекти використання результатів дослідження обговорювалися на засіданні кафедри прикладної математики та інформатики (протокол №01 від 27.08.2026 року). Ухвалено, що узагальнення та висновки наукового дослідження, представлені В. В. Бондарем в оприлюднених наукових працях, дають підстави рекомендувати їх до застосування в освітньому процесі закладів вищої освіти для поглиблення підготовки студентів спеціальностей прикладної математики та інформаційних систем і технологій, зокрема під час опанування методів ймовірнісного моделювання, статистичного аналізу, лінійної алгебри та математичного моделювання задач машинного навчання.

Проректор з наукової, інноваційної
та міжнародної діяльності



Олександр СПРЯГАЙЛО

Додаток В

Декларація використання систем штучного інтелекту

Автор: Бондар Віталій Володимирович.

Назва роботи: дисертація „Моделі та метод нейромережевої генерації об’єктів на основі профілів швидкості узгодження потоків“ на здобуття ступеня доктора філософії за спеціальністю 123 – Комп’ютерна інженерія.

Системи штучного інтелекту, постачальник, модель/версія: Anthropic Claude Sonnet (версії 3.5–5), Anthropic Claude Opus (версії 3–5).

Період використання: вересень 2024 р. – вересень 2026 р.

Делеговані завдання: 1) мовне та стилістичне редагування тексту дисертації (перевірка граматики, орфографії та пунктуації, виявлення стилістичних огріхів, повторів і неузгодженого вживання термінів); 2) технічна підтримка верстки в системі $\text{\LaTeX}$ та оформлення за чинними вимогами.

Спосіб перевірки та доопрацювання виходів: усі зміни вносилися під наглядом автора: кожен запропонований правку тексту й верстки автор переглядав, перевіряв і приймав або відхиляв особисто.

Підтверджую, що системи штучного інтелекту використовувалися під моїм наглядом, усі суттєві твердження, дані, посилання, обчислення та результати перевірено, а повну відповідальність за остаточний текст дисертації несу я як автор. Системи штучного інтелекту не є автором чи співавтором дисертації.

16 вересня 2026 р.



Віталій БОНДАР