

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

ГІРЯНСЬКИЙ БОГДАН ПЕТРОВИЧ

УДК 004.852 + 519.85

ДИСЕРТАЦІЯ

**Оптимізація параметрів та структури нейронних мереж із
застосуванням еволюційних алгоритмів**

122 - Комп'ютерні науки

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

Б.П. Гірянський

Науковий керівник: Булах Богдан Вікторович, к.т.н.

Київ – 2026

АНОТАЦІЯ

Гірянський Б.П. Оптимізація параметрів та структури нейронних мереж із застосуванням еволюційних алгоритмів. — Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 — Комп'ютерні науки. — Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, 2026.

Метою дослідження є підвищення ефективності процесів оптимізації параметрів та структури нейронних мереж шляхом розробки та дослідження еволюційних методів стиснення, оптимізації гіперпараметрів та злиття моделей.

Об'єктом дослідження є процеси оптимізації параметрів та структури нейронних мереж.

Предметом дослідження є методи та алгоритми застосування еволюційних обчислень для структурної оптимізації, пошуку гіперпараметрів і злиття нейронних мереж.

Методи дослідження. Використано: методи системного аналізу — для класифікації підходів та формулювання задач. Еволюційні алгоритми безперервної оптимізації (CMA-ES, диференціальна еволюція, генетичний алгоритм) — для пошарового розподілу розрідженості, оптимізації гіперпараметрів та злиття моделей. Методи теорії матриць (апроксимація сліду матриці Гессе, двонаправлена метрика спорідненості нейронів) — для оцінки важливості ваг та нейронів, сурогатне моделювання (Random Forest) — для оптимізації гіперпараметрів, ϵ -жадібний багаторукий бандит — для контролю достовірності сурогату, непараметрична статистика (критерії Фрідмана, Немені, Вілкоксона, бутстрап-довірчі інтервали) — для оцінювання ефективності методів. Глибоке навчання та об'єктно-орієнтоване програмування — для реалізації прототипів.

Наукова новизна одержаних результатів.

Вперше запропоновано метод проріджування нейронної мережі без зміни її структури, у якому частка збережених вагових коефіцієнтів у кожному шарі визначається еволюційним пошуком. Метод відрізняється від відомих підходів із глобальним порогом або фіксованим пошаровим розподілом тим, що після проміжного проріджування повторно обчислюється значущість вагових коефіцієнтів і враховується штраф за повністю деактивовані нейрони. Це зменшує ризик руйнування критичних шляхів передавання сигналу та забезпечує меншу втрату якісних характеристик мережі при високих рівнях проріджування.

Вперше запропоновано метод ущільнення розрідженої нейронної мережі, що не потребує ані перенавчання моделі, ані даних для коригування вагових коефіцієнтів. На відміну від підходів, які видаляють усі деактивовані нейрони, або застосовують низькорангову апроксимацію матриці ваг, чи потребують навчальних даних для створення спрощених моделей ШНМ, запропонований метод еволюційно визначає кількість нейронів, які доцільно зберегти в кожному шарі, та об'єднує надлишкові нейрони за спорідненістю їхніх вхідних і вихідних зв'язків. Це дає змогу отримати меншу модель із кращим співвідношенням між стисненням і збереженням якісних характеристик мережі.

Вперше запропоновано метод пошуку гіперпараметрів нейронних мереж, у якому еволюційний алгоритм формує та відбирає можливі значення гіперпараметрів без повного навчання мережі для кожного можливого варіанта. Відбір здійснюється за моделлю залежності між значеннями гіперпараметрів і результатами вже виконаних навчань: допоміжна модель визначає очікуване значення цільової функції для сформованих варіантів, після чого на реальне навчання передається лише один із них. Це зменшує кількість повних циклів навчань нейронної мережі та підвищує ефективність використання обчислювального бюджету.

Удосконалено метод злиття нейронних мереж у випадку, коли вихідні моделі навчені на різних підмножинах класів або даних. Метод не усереднює ваги моделей, а формує об'єднану мережу з частини нейронів першої та другої моделей. Склад цієї

мережі й внесок кожної моделі для різних класів визначає генетичний алгоритм. Цільова функція враховує частку правильно розпізнаних прикладів у всій валідаційній вибірці та окремо точність для найгірше розпізнаваного класу, що зменшує ризик втрати здатності розпізнавати окремі класи та допомагає зберігати знання, набуті обома моделями.

Практичне значення одержаних результатів. Метод TESA-26 дає змогу автоматично визначати максимальний ступінь стиснення нейронної мережі, за якого зберігається заданий рівень якості моделі. У порівняльному дослідженні на 6 наборах даних, 12 рівнях проріджування та 1080 експериментальних записах метод показав найкращий середній ранг Фрідмана 2,04 і статистично значущо перевершив SparseGPT, WANDA та RIA ($\chi^2 = 18,05$, $p < 0,0001$). Метод GFCS дає змогу перетворити проріджену нейронну мережу на фізично меншу щільну архітектуру зі зменшеною кількістю нейронів і зв'язків. У проведених експериментах він забезпечив стиснення у 7,6 рази та середнє прискорення обчислень моделі у 1,42 рази зі збереженням якості на всіх 8 досліджених наборах даних. Методи SACMA-DAC та SACMA-MAB скорочують кількість повних циклів навчання нейронної мережі, необхідних для добору гіперпараметрів. У дослідженні на 43 задачах із 10 повторами вони посіли 1–2 місця серед 9 підходів за середнім рангом 2,47 і 2,95 та досягли найвищої швидкості збіжності за показником AUCC = 0,9257 і 0,9210 відповідно. Метод ENT дає змогу об'єднувати нейронні мережі, навчені на різних підмножинах класів або даних, в одну модель без повного навчання з нуля і у сценарії комплементарного злиття ENT забезпечив розпізнавання 10 з 10 класів, точність 0,749 і баланс 0,981, а після калібрації ENT-FT точність зросла до 0,917.

Опис розділів дисертації.

У першому розділі проаналізовано наукову літературу за п'ятьма напрямками: еволюційні алгоритми безперервної оптимізації, структурне стиснення нейронних мереж, оптимізація гіперпараметрів, злиття моделей, апаратно-незалежні метрики

обчислювальної вартості. Виявлено три прогалини, що визначили напрямки дослідження.

У другому розділі розроблено еволюційні методи стиснення нейронних мереж: апаратно-незалежну метрику RCU, метод неструктурованого проріджування TESA-26 з еволюційним пошаровим розподілом розрідженості і метод зворотної конверсії розрідженої мережі у компактну щільну архітектуру GFCS без використання навчальних даних. Проведено порівняння з відомими методами на різнотипних задачах класифікації.

У третьому розділі досліджено застосування еволюційних алгоритмів до пошуку гіперпараметрів нейронних мереж: порівняно CMA-ES з баєсівськими оптимізаторами, досліджено проксі-оцінки та ординальне кодування, розроблено метод у двох варіаціях SACMA-DAC та SACMA-MAB для пошуку гіперпараметрів із сурогатною моделлю та додатково досліджено метод WL-CMA.

У четвертому розділі досліджено застосування еволюційних алгоритмів до злиття нейронних мереж з комплементарними знаннями: проаналізовано обмеження інтерполяційних методів, розроблено метод еволюційного відбору нейронної топології (ENT), розроблено етап калібрації ENT-FT та проведено порівняльний аналіз з методами Sakana-CMA, TIES-Merging, Task Arithmetic.

Особистий внесок здобувача. Усі основні результати дисертаційного дослідження, подані до захисту, одержані автором особисто. Здобувачем виконано: аналіз існуючих методів структурної оптимізації нейронних мереж, розробку методів TESA-26, GFCS, SACMA-DAC, SACMA-MAB та ENT, розробку апаратно-незалежної метрики RCU, проведення обчислювальних експериментів та статистичний аналіз результатів. У публікаціях у співавторстві здобувачеві належать: дослідження існуючих методів, проектування та реалізація запропонованих методів, проведення експериментів, статистичне оцінювання ефективності.

Апробація матеріалів дисертації. Основні результати дисертаційної роботи доповідались та обговорювались на міжнародній науково-практичній конференції

Science and Information Technologies in the Modern World (Афіни, Греція, 24–26 грудня 2025 р.).

Ключові слова: генетичний алгоритм, згорткові нейронні мережі (CNN), глибоке навчання, машинне навчання, нейронні мережі, багатошаровий перцептрон (MLP), Python, NumPy, апаратні обмеження, CMA-ES, оптимізація гіперпараметрів, злиття, проріджування, стиснення моделей, сурогатна модель.

Список публікацій здобувача за темою дисертації.

1. Hirianskyi B., Bulakh B. Effectiveness of evolutionary algorithms in neural network optimization tasks: Unconditional minimization, pruning, and hyperparameter optimization. Control, Navigation and Communication Systems, № 1, 2026, p. 45–50. DOI: 10.26906/sunz.2026.1.045. Здобувачем особисто проведено аналіз сучасного стану наукових досліджень у напрямі застосування еволюційних обчислень до задач безградієнтної оптимізації нейронних. Спроектовано загальну архітектуру експериментального дослідження для трьох класів задач: безумовної мінімізації на еталонних тестових функціях, структурного проріджування нейронних мереж на наборах даних Digits та FashionMNIST на рівнях розрідженості від помірних до екстремальних (98 %), а також оптимізації гіперпараметрів у шестивимірному змішаному просторі параметрів. Здійснено програмну реалізацію усіх застосованих еволюційних алгоритмів і запропонованих у статті вдосконалень: модифікації Evo-SynFlow з механізмом енергетичної компенсації проріджених вагових коефіцієнтів та модифікації Evo-SynFlow із самоадаптивною стратегією та багатоетапним відбором. Особистий внесок співавтора Булаха Б. В.: здійснено постановку дослідницької задачі у вигляді єдиного експериментального каркасу для систематичного порівняння трьох парадигм оптимізації нейронних мереж та сформульовано вихідні гіпотези дослідження. Визначено набір базових алгоритмів-конкурентів для забезпечення репрезентативності порівняльного дослідження — представників градієнтної парадигми. Надано консультативну підтримку щодо

параметризації еволюційних стратегій та коректного налаштування статистичних тестів для непараметричного аналізу результатів. Здійснено наукове редагування рукопису з метою забезпечення термінологічної точності та узгодженості стилю наукової подачі результатів.

2. Гірянський Б.П., Булах Б.В. Створення нейронних мереж шляхом нейроеволюції та автоматизованого навчання з використанням еволюційних алгоритмів. *Наука та техніка сьогодні*, № 8(49), 2025, с. 1214–1227. DOI: 10.52058/2786-6025-2025-8(49)-1214-1227. Здобувачем запропоновано поєднання еволюційного пошуку з градієнтним донавчанням, розроблено два авторські нейроеволюційні алгоритми: меметичний (МНЕА), що одночасно еволюціонує топологію та вагові коефіцієнти мережі з інтегрованим кроком локального градієнтного донавчання найкращих індивідів за аналогією з ламарківською еволюцією та гібридний (ГНЕА), що розділяє пошук абстрактного «рецепта» архітектури (з параметрами кількості шарів, типу активації, швидкості навчання та коефіцієнта L2-регуляризації) від подальшого фінального градієнтного навчання. *Особистий внесок співавтора Булаха Б. В.:* Надано консультативну підтримку щодо вибору тестових наборів даних, що мають охопити різні рівні нелінійності меж рішень — від простих синтетичних задач до реальних медичних даних. Здійснено експертну оцінку коректності протоколів оцінювання та проведено наукове редагування рукопису.
3. Hirianskyi B.P., Bulakh B.V. Using evolutionary algorithms for training and optimizing neural networks. *Наука та техніка сьогодні*, № 8(36), 2024, с. 808–823. DOI: 10.52058/2786-6025-2024-8(36)-808-823. *Особистий внесок здобувача:* проаналізовано класи еволюційних алгоритмів та обґрунтовано можливість їх застосування для задач оптимізації нейронних мереж. На основі експериментів встановлено залежність ефективності еволюційного пошуку від параметрів алгоритму. Узагальнено результати застосування еволюційних алгоритмів за публікаціями останніх років, виконано власні експерименти з підбору

параметрів алгоритму і обґрунтовано доцільність застосування еволюційних алгоритмів для оптимізації нейронних мереж. *Особистий внесок співавтора Булаха Б. В.*: постановка задачі аналітичного огляду класів еволюційних алгоритмів, визначення структури огляду та критеріїв систематизації джерел і обговорення дизайну експериментів з підбору параметрів, редагування рукопису.

4. Hirianskyi B., Bulakh B. A review of practice of using evolutionary algorithms for neural network synthesis and training. *Technology Audit and Production Reserves*, № 4(2(72)), 2023, p. 22–26. DOI: 10.15587/2706-5448.2023.286278. *Особистий внесок здобувача*: систематизовано підходи на основі генетичних алгоритмів за чотирма категоріями застосування: ГА для автоматичного відбору ознак, ініціалізації та оптимізації вагових коефіцієнтів, модифікації структури міжнейронних зв'язків та налаштування параметрів класичних оптимізаторів (крок навчання, інерція). Виокремлено основні проблеми синтезу нейронних мереж для яких еволюційні підходи демонструють потенційні переваги. *Особистий внесок співавтора Булаха Б. В.*: постановка завдання аналітичного огляду та визначення його тематичних меж, формулювання структури огляду та критеріїв класифікації підходів та редагування рукопису та обговорення висновків.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

5. Гірянський Б.П. Використання еволюційних алгоритмів для синтезу та навчання нейронних мереж: огляд сучасного стану та перспективи розвитку. *Science and Information Technologies in the Modern World: збірка наукових праць з матеріалів 4-ї Міжнародної науково-практичної конференції*, 2025, с. 165–168. DOI: 10.70286/isu-24.12.2025.006. Доповідь репрезентувала узагальнення основних теоретичних положень та результатів дисертації перед міжнародною науковою спільнотою: огляд сучасного стану застосування еволюційних алгоритмів для оптимізації нейронних мереж.

ABSTRACT

Hirianskyi B.P. Optimization of Parameters and Structure of Neural Networks Using Evolutionary Algorithms. — Qualifying scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 122 — Computer Science. — National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, 2026.

The research objective is to improve the efficiency of parameter and structure optimization processes for neural networks through the development and investigation of evolutionary methods for model compression, hyperparameter optimization, and merging.

The object of research is the processes of optimizing the parameters and structure of neural networks.

The subject of research is methods and algorithms for applying evolutionary computation to structural optimization, hyperparameter search, and neural network merging.

Research methods. The dissertation employs: methods of system analysis — for classifying approaches and defining research problems, continuous optimization evolutionary algorithms (CMA-ES, differential evolution, genetic algorithm) — for layerwise sparsity allocation, hyperparameter optimization, and model merging, matrix theory methods (Hessian trace approximation, bidirectional neuron affinity metric) — for evaluating weight and neuron importance, surrogate modeling (Random Forest) — for hyperparameter optimization, ϵ -greedy multi-armed bandit — for surrogate reliability control and nonparametric statistics (Friedman, Nemenyi, Wilcoxon tests, bootstrap confidence intervals) — for evaluating method effectiveness and deep learning and object-oriented programming — for implementing software prototypes.

Scientific novelty of the obtained results.

For the first time, a pruning method without changing the network structure has been proposed, in which the fraction of retained weights in each layer is determined by evolutionary search. The method differs from known approaches with a global threshold or

fixed layerwise allocation by recomputing weight saliency after intermediate pruning and by accounting for a penalty for fully deactivated neurons. This reduces the risk of disrupting critical signal paths and provides a smaller loss of network quality at high pruning levels.

For the first time, a method has been proposed for densifying a sparse neural network that requires neither model retraining nor data for adjusting weight coefficients. Unlike approaches that remove all deactivated neurons, apply low-rank approximation of the weight matrix, or require training data to construct simplified ANN models, the proposed method evolutionarily determines the number of neurons to retain in each layer and merges redundant neurons according to the similarity of their incoming and outgoing connections. This makes it possible to obtain a smaller model with a better trade-off between compression and preservation of the network's quality characteristics.

For the first time, a neural network hyperparameter search method has been proposed in which an evolutionary algorithm forms and selects possible hyperparameter values without fully training the network for each variant. The selection is performed using a model of the relationship between hyperparameter values and the results of previously completed trainings: an auxiliary model determines the expected value of the objective function for the generated variants, after which only one of them is sent for real training. This reduces the number of full neural network trainings and improves the efficiency of computational budget use.

Improved is the method for merging neural networks when the source models are trained on different subsets of classes or data. The method does not average model weights but forms a merged network from part of the neurons of the first and second models and the composition of this network and the contribution of each model for different classes are determined by a genetic algorithm. The objective function accounts for the fraction of correctly recognized examples in the entire validation set and separately for the accuracy of the worst-recognized class, which reduces the risk of losing the ability to recognize individual classes and helps preserve the knowledge acquired by both models.

Practical significance of the obtained results. The TESA-26 method makes it possible to automatically determine the maximum compression level of a neural network at which a specified model quality level is preserved, in a comparative study on 6 datasets, 12 pruning levels, and 1,080 experimental records, the method achieved the best mean Friedman rank of 2.04 and statistically significantly outperformed SparseGPT, WANDA, and RIA ($\chi^2 = 18.05$, $p < 0.0001$). The GFCS method makes it possible to transform a pruned neural network into a physically smaller dense architecture with fewer neurons and connections, in experiments, it achieved $7.6\times$ compression and an average inference speedup of $1.42\times$ while preserving quality on all 8 investigated datasets. The SACMA-DAC and SACMA-MAB methods reduce the number of full neural network training cycles required for hyperparameter selection, in a study on 43 tasks with 10 repetitions, they ranked 1st and 2nd among 9 approaches with mean ranks of 2.47 and 2.95 and achieved the highest convergence speed according to AUCC = 0.9257 and 0.9210, respectively. The ENT method enables neural networks trained on different subsets of classes or data to be merged into one model without full training from scratch and in the complementary merging scenario, ENT recognized 10 out of 10 classes, achieved accuracy of 0.749 and balance of 0.981, and after ENT-FT calibration the accuracy increased to 0.917.

Chapter descriptions.

Chapter 1 analyzes the scientific literature across five directions: continuous optimization evolutionary algorithms, structural compression of neural networks, hyperparameter optimization, model merging and hardware-independent computational cost metrics. Three gaps are identified that define the research directions.

Chapter 2 develops evolutionary methods for neural network compression: the hardware-independent computational cost metric RCU, the unstructured pruning method TESA-26 with evolutionary layerwise sparsity allocation and the inverse conversion method GFCS from a sparse network to a compact dense architecture without using training data. A comparison with known methods is conducted on diverse classification tasks.

Chapter 3 investigates the application of evolutionary algorithms to neural network hyperparameter search: CMA-ES is compared with Bayesian optimizers, proxy evaluations and ordinal encoding are investigated. A method in two variations, SACMA-DAC and SACMA-MAB, is developed for finding hyperparameters with a surrogate model. The WL-CMA method is additionally investigated as an auxiliary approach

Chapter 4 investigates the application of evolutionary algorithms to merging neural networks with complementary knowledge: limitations of interpolation methods are analyzed, the evolutionary neuro-transplantation method (ENT) is developed, the ENT-FT calibration stage is developed and a comparative analysis with Sakana-CMA, TIES-Merging, and Task Arithmetic methods is conducted.

Personal contribution of the applicant. All main results of the dissertation research presented for defense were obtained by the author personally. The applicant performed: analysis of existing methods of structural optimization of neural networks, development of the TESA-26, GFCS, SACMA-DAC, SACMA-MAB, and ENT methods, development of the hardware-independent RCU metric and conducting computational experiments and statistical analysis of results. In co-authored publications, the applicant's contribution includes: research of existing methods, design and implementation of proposed methods, conducting experiments, statistical evaluation of effectiveness.

Approbation of dissertation materials. The main results of the dissertation were presented and discussed at the international scientific-practical conference Science and Information Technologies in the Modern World (Athens, Greece, December 24–26, 2025).

Keywords: genetic algorithm, convolutional neural networks (CNN), deep learning, machine learning, neural networks, multilayer perceptron (MLP), Python, NumPy, hardware constraints, CMA-ES, hyperparameter optimization, model merging, pruning, model compression, surrogate model.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	15
ВСТУП.....	17
Актуальність теми	17
Ступінь наукової розробленості проблеми	18
Мета і завдання дослідження	20
Об’єкт дослідження	21
Предмет дослідження	21
Методи дослідження.	21
Наукова новизна	22
Практичне значення одержаних результатів.	23
Особистий внесок здобувача	24
Апробація матеріалів дисертації	25
Публікації	25
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ВИКОРИСТАННЯ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ ДЛЯ ОПТИМІЗАЦІЇ НЕЙРОННИХ МЕРЕЖ.....	29
1.1. Проблема оптимізації в глибокому навчанні.....	29
1.2. Проблема оцінки обчислювальної вартості нейронних мереж.....	31
1.3. Еволюційні алгоритми безперервної оптимізації.....	32
1.4. Методи структурного стиснення нейронних мереж	34
1.5. Методи оптимізації гіперпараметрів нейронних мереж.....	39
1.6. Злиття моделей нейронних мереж	42
1.7. Визначення прогалів у існуючих дослідженнях та постановка задач ...	45
Висновки до розділу 1	47
РОЗДІЛ 2. ЕВОЛЮЦІЙНЕ СТИСНЕННЯ НЕЙРОННИХ МЕРЕЖ	49
2.1. Середовище виконання експериментів та метрика обчислювальної вартості	49
2.2. Обґрунтування конкурентоспроможності еволюційних алгоритмів	58
2.3. Еволюційні методи неструктурованого проріджування нейронних мереж	69
2.4. Методи конверсії розріджених мереж у компактні щільні архітектури .	85

	14
2.5. Обмеження досліджень.....	109
Висновки до розділу 2.....	112
РОЗДІЛ 3. ЕВОЛЮЦІЙНІ ТА АДАПТИВНІ МЕТОДИ ОПТИМІЗАЦІЇ ГІПЕРПАРАМЕТРІВ НЕЙРОННИХ МЕРЕЖ	115
3.1. Постановка задачі оптимізації гіперпараметрів нейронних мереж.....	115
3.2. Аналіз обмежень базових підходів до оптимізації гіперпараметрів	117
3.3. Метод SACMA-DAC — пошук гіперпараметрів із сурогатною моделлю	119
3.4. Метод SACMA-MAV — пошук гіперпараметрів із багаторукиим бандитом	124
3.5. Метод WL-CMA — гаусівський процес з деформацією Єо–Джонсона і динамікою Ланжевена.....	127
3.6. Масштабне порівняльне дослідження методів оптимізації гіперпараметрів.....	131
3.7. Обмеження досліджень.....	139
Висновки до розділу 3.....	142
РОЗДІЛ 4. ЕВОЛЮЦІЙНЕ ЗЛИТТЯ НЕЙРОННИХ МЕРЕЖ.....	144
4.1. Аналіз обмежень методів інтерполяції ваг при злитті моделей	144
4.2. Метод ENT — еволюційний відбір нейронної топології	148
4.3. Метод ENT-FT — калібрація зливої моделі на малих вибірках	153
4.4. Порівняльний аналіз методу ENT із сучасними методами злиття	156
4.5. Обмеження досліджень.....	163
Висновки до розділу 4.....	165
ВИСНОВКИ.....	167
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	170
ДОДАТОК А. Програмна реалізація запропонованих методів	185
ДОДАТОК Б. Акт впровадження результатів дисертаційної роботи у виробничу діяльність ТОВ «СД Профзв’язок».....	187

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Скорочення.

- CMA-ES — Covariance Matrix Adaptation Evolution Strategy (еволюційна стратегія з адаптацією коваріаційної матриці)
- DARTS — Differentiable Architecture Search (диференційований пошук архітектури)
- DSA — Dynamic Sparsity Allocation (динамічний розподіл розрідженості)
- EA — Evolutionary Algorithm (еволюційний алгоритм)
- ENT — Evolutionary Neuro-Transplantation (еволюційний відбір нейронної топології)
- ES — Evolution Strategy (еволюційна стратегія)
- FC — Fully Connected (повнозв'язний шар)
- FIM — Fisher Information Matrix (інформаційна матриця Фішера)
- GFCS — Gradient-Flow Connectivity Synthesis (еволюційний синтез зв'язності на основі потоку градієнтів)
- HPO — Hyperparameter Optimization (оптимізація гіперпараметрів)
- IGO — Information Geometric Optimization (інформаційно-геометрична оптимізація)
- SACMA — Surrogate-Assisted CMA-ES (стратегія адаптації коваріаційної матриці із сурогатною моделлю)
- WL-CMA — Warped Langevin CMA-ES (деформована стратегія адаптації коваріаційної матриці з динамікою Ланжевена)
- KD — Knowledge Distillation (дистиляція знань)
- LTH — Lottery Ticket Hypothesis (гіпотеза лотерейного квитка)
- MLP — Multilayer Perceptron (багатошаровий перцептрон)

- NAS — Neural Architecture Search (пошук архітектури нейронних мереж)
- NES — Natural Evolution Strategies (стратегії природної еволюції)
- NFP — Neuron Fatality Penalty (штраф за деактивовані нейрони)
- RCU — Relative Computational Unit (відносна обчислювальна одиниця)
- RIA — Relative Importance and Activations (відносна важливість та активації)
- RPR — Recovery Performance Ratio (коефіцієнт відновлення якості)
- SGD — Stochastic Gradient Descent (стохастичний градієнтний спуск)
- SVD — Singular Value Decomposition (сингулярний розклад)
- TESA-26 — Taylor Evolutionary Sparsity Allocation (пошарове проріджування з ітеративним перерахунком значущості ваг)
- TIES — TrIm, Elect Sign & merge (метод злиття з відсіканням, голосуванням за знак та об'єднанням)
- TPE — Tree-structured Parzen Estimator (структуровані деревом оцінювачі Парзена)
- WANDA — Weights AND Activations (метод проріджування на основі ваг та активацій)

Символи.

- s — рівень розрідженості (sparsity), частка нульових ваг
- φ_i — метрика важливості (потоківий скор) i -го нейрона
- k_l — кількість збережених нейронів у шарі l
- μ, λ — розмір батьківської популяції та кількість нащадків в ЕА
- σ — початкова дисперсія (крок мутації) в СМА-ES
- α — коефіцієнт змішування або ваговий параметр
- τ_b — коефіцієнт кореляції Кендала
- F_1 — міра якості класифікації (гармонічне середнє precision і recall)
- $\text{Comp } \times$ — коефіцієнт стиснення, відношення кількості параметрів вчителя до компактної моделі

ВСТУП

Актуальність теми

Стрімкий розвиток методів глибокого навчання призвів до створення нейронних мереж безпрецедентного масштабу: сучасні великі мовні моделі налічують від 7 до 176 мільярдів параметрів [1], а пошук оптимальної архітектури класичними методами NAS вимагає до 2000–3150 GPU-днів [2, 3]. Це створює фундаментальну суперечність між зростаючою якістю моделей та обмеженими обчислювальними ресурсами платформ, особливо при розгортанні на периферійних (edge) пристроях [4, 5].

Зменшення обчислювальної вартості моделей є одним з пріоритетних напрямків досліджень. Методи неструктурованого проріджування, зокрема SparseGPT [1] та Wanda [6], дозволяють видаляти до 50–60% ваг великих мовних моделей із мінімальною втратою якості. Водночас ефективність проріджування критично залежить від пошарового розподілу розрідженості: існуючі аналітичні формули [7] та градієнтні підходи [8] не враховують зміну значущості ваг у процесі послідовного видалення, що обмежує їх застосування при високих рівнях стиснення. Крім того, розріджені архітектури потребують спеціалізованого апаратного забезпечення для ефективного виконання, тоді як задача зворотної конверсії розрідженої мережі у компактну щільну модель без доступу до навчальних даних залишається невирішеною.

Додатковим обмежувальним фактором є оптимізація гіперпараметрів нейронних мереж — дворівнева задача [9], де кожна оцінка конфігурації потребує повного циклу навчання моделі. Еволюційні алгоритми, зокрема CMA-ES [10] та диференціальна еволюція [11], демонструють стійкість до зашумлених та мультимодальних ландшафтів [12], що робить їх перспективним інструментарієм для розв’язання задач оптимізації нейронних мереж, де градієнтні методи непридатні.

Окремою актуальною задачею є злиття попередньо навчених моделей. Існуючі методи інтерполяції ваг [13, 14] та еволюційної оптимізації скалярних коефіцієнтів [15] ефективні для моделей із спільною базовою архітектурою та перекриваючими

знаннями, проте непридатні для злиття спеціалізованих моделей з комплементарними (взаємодоповнювальними) знаннями, де вони можуть призводити до втрати здатності розпізнавати окремі класи та колапсу передбачень.

Таким чином, актуальність дисертаційного дослідження визначається необхідністю розробки нових еволюційних методів, здатних ефективно розв'язувати задачі структурної оптимізації нейронних мереж, для яких класичні градієнтні підходи є непридатними: пошаровий розподіл розрідженості, конверсія розріджених архітектур у компактні щільні моделі без перенавчання та злиття моделей на рівні нейронної топології.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконана в Навчально-науковому комплексі «Інститут прикладного системного аналізу» Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» у межах науково-дослідної роботи «Впровадження сервіс-орієнтованого підходу до реалізації процесів діджиталізації суспільства» (державний реєстраційний номер 0123U101333, 2023–2025 рр., науковий керівник — д.т.н., проф. Петренко А.І.). Здобувачем розроблено еволюційні методи оптимізації нейронних мереж для підвищення ефективності обчислювальних сценаріїв у складі сервіс-орієнтованих систем.

Ступінь наукової розробленості проблеми.

Проблема оптимізації нейронних мереж досліджувалась у працях вітчизняних та зарубіжних вчених за кількома взаємопов'язаними напрямками.

Теоретичні основи еволюційних алгоритмів безперервної оптимізації закладено у працях Hansen [10] щодо стратегії адаптації коваріаційної матриці CMA-ES, Storn та Price [11] щодо диференціальної еволюції, Tanabe та Fukunaga [16] щодо адаптивної диференціальної еволюції SHADE. Масштабування еволюційних алгоритмів для високорозмірних задач досліджено у роботах Guo et al. [17] та Loshchilov та Hutter [12]. Застосування еволюційних стратегій у контексті інформаційної геометрії висвітлено у працях Wierstra et al. [18] та Ollivier et al. [19].

Машинне навчання активно розвивалось у напрямі проріджування мереж: від класичних підходів Optimal Brain Damage [20] та магнітудного проріджування [21] до сучасних однократних методів для великих мовних моделей: SparseGPT [1], Wanda [6], CVR+EC [22]. Методи проріджування без доступу до даних досліджено у роботі Tanaka et al. [23]. Еволюційний підхід до автоматичної генерації метрик проріджування запропоновано у [24]. Проблему пошарового розподілу розрідженості досліджено у працях Ning et al. [8], Lee et al. [7] та Evci et al. [25]. Водночас задача зворотної конверсії розрідженої мережі у компактну щільну архітектуру залишається недостатньо дослідженою.

Оптимізація гіперпараметрів нейронних мереж систематизована у працях Franceschi et al. [9] та Liu et al. [26]. Баєсівські підходи розроблено у працях Falkner et al. [27] та Watanabe та Hutter [28]. Застосування еволюційних алгоритмів для пошуку архітектур досліджено у роботах Real et al. [3] та Zou et al. [29]. Проте питання гібридизації DE-операторів мутації з інформаційно-геометричною корекцією для задач оптимізації гіперпараметрів залишається невисвітленим.

Проблема злиття нейронних мереж активно досліджується у працях Ilharco et al. [13], Yadav et al. [14], Akiba et al. [15]. Stoica et al. [30] запропонували зшивання нейронів за активаціями, а Wang et al. [31] — евристичне маскування параметрів. Водночас усі зазначені методи оперують на рівні скалярних коефіцієнтів або окремих ваг, залишаючи невирішеною задачу еволюційного відбору топології на рівні нейронів при злитті моделей з комплементарними знаннями.

Таким чином, аналіз літературних джерел свідчить, що незважаючи на значний прогрес у розвитку окремих методів оптимізації нейронних мереж, залишаються недостатньо дослідженими питання: (1) еволюційного пошарового розподілу розрідженості з ітеративною перекалібрацією значущості ваг, (2) зворотної конверсії розрідженої мережі у компактну щільну архітектуру на основі еволюційних стратегій, (3) еволюційного відбору нейронної топології для злиття моделей з

комплементарними знаннями. Це визначило напрямки дослідження, представленого у дисертаційній роботі.

Мета і завдання дослідження.

Метою дослідження є підвищення ефективності процесів оптимізації параметрів та структури нейронних мереж шляхом розробки та дослідження еволюційних методів стиснення, оптимізації гіперпараметрів та злиття моделей.

Для досягнення встановленої мети сформульовано наступні завдання:

1. Проаналізувати існуючі методи структурної оптимізації нейронних мереж, еволюційні алгоритми безперервної оптимізації та методи злиття моделей, визначити їх переваги та обмеження.
2. Дослідити ефективність еволюційних алгоритмів для задач оптимізації нейронних мереж порівняно з градієнтними методами та визначити межі їх застосовності.
3. Розробити метод пошарового розподілу розрідженості нейронної мережі на основі еволюційного алгоритму з ітеративною перекалібрацією значущості ваг.
4. Розробити метод конверсії розрідженої нейронної мережі у компактну щільну архітектуру на основі еволюційної стратегії синтезу зв'язності без доступу до навчальних даних.
5. Розробити методи пошуку гіперпараметрів нейронних мереж із сурогатною моделлю на основі стратегії адаптації коваріаційної матриці та верифікувати їх у масштабному порівняльному дослідженні з існуючими підходами.
6. Розробити метод злиття нейронних мереж з комплементарними знаннями на основі генетичного алгоритму з еволюційним відбором нейронної топології.
7. Виконати формалізований опис архітектури та компонентів запропонованих методів.
8. Реалізувати програмні прототипи для експериментальної перевірки запропонованих методів.

9. Провести експериментальне дослідження та порівняльний аналіз із існуючими підходами із застосуванням непараметричних статистичних критеріїв на множині наборів даних та архітектур.

Об’єкт дослідження — процеси оптимізації параметрів та структури нейронних мереж.

Предмет дослідження — методи та алгоритми застосування еволюційних обчислень для структурної оптимізації, пошуку гіперпараметрів і злиття нейронних мереж.

Методи дослідження.

Для розв’язання поставлених завдань використано такі методи:

- методи системного аналізу та формалізації — для класифікації існуючих підходів до структурної оптимізації нейронних мереж, виявлення їх обмежень та формулювання задач дисертаційного дослідження (Розділ 1);

- еволюційні алгоритми безперервної оптимізації (стратегія адаптації коваріаційної матриці CMA-ES, диференціальна еволюція DE, генетичний алгоритм) — як основний інструментарій для розв’язання задач пошарового розподілу розрідженості (Розділ 2), оптимізації гіперпараметрів (Розділ 3) та злиття моделей (Розділ 4);

- методи теорії матриць та числового аналізу (апроксимація сліду матриці Гессе, двонаправлена метрика спорідненості нейронів) — для побудови функцій оцінки важливості ваг та нейронів (Розділ 2);

- методи сурогатного моделювання (Random Forest) — для розроблення ефективних методів оптимізації гіперпараметрів нейронних мереж (Розділ 3);

- методи послідовного прийняття рішень (ϵ -жадібний багаторукий бандит з cost-aware функцією нагороди) — для адаптивного контролю достовірності сурогатної моделі та адаптації параметрів пошуку в методі SACMA-MAB (Розділ 3);

– методи непараметричної статистики (критерій Фрідмана, post-hoc тест Немені, критерій Вілкоксона, бутстрап-довірчі інтервали, критерій хі-квадрат) — для статистичного оцінювання ефективності запропонованих методів та порівняння з існуючими підходами (Розділи 2, 3, 4);

– методи машинного та глибокого навчання (зворотне поширення помилки, пакетна нормалізація, регуляризація) — для навчання та тонкого налаштування нейронних мереж у межах обчислювальних експериментів;

– методи об’єктно-орієнтованого програмування та прототипування — для реалізації програмних прототипів запропонованих методів мовою Python із використанням бібліотек PyTorch та CMA (руста).

Наукова новизна одержаних результатів.

Вперше запропоновано метод проріджування нейронної мережі без зміни її структури, у якому частка збережених вагових коефіцієнтів у кожному шарі визначається еволюційним пошуком. Метод відрізняється від відомих підходів із глобальним порогом або фіксованим пошаровим розподілом тим, що після проміжного проріджування повторно обчислюється значущість вагових коефіцієнтів і враховується штраф за повністю деактивовані нейрони. Це зменшує ризик руйнування критичних шляхів передавання сигналу та забезпечує меншу втрату якісних характеристик мережі при високих рівнях проріджування.

Вперше запропоновано метод ущільнення розрідженої нейронної мережі, що не потребує ані перенавчання моделі, ані даних для коригування вагових коефіцієнтів. На відміну від підходів, які видаляють усі деактивовані нейрони, або застосовують низькорангову апроксимацію матриці ваг, чи потребують навчальних даних для створення спрощених моделей ШНМ, запропонований метод еволюційно визначає кількість нейронів, які доцільно зберегти в кожному шарі, та об’єднує надлишкові нейрони за спорідненістю їхніх вхідних і вихідних зв’язків. Це дає змогу отримати меншу модель із кращим співвідношенням між стисненням і збереженням якісних характеристик мережі.

Вперше запропоновано метод пошуку гіперпараметрів нейронних мереж, у якому еволюційний алгоритм формує та відбирає можливі значення гіперпараметрів без повного навчання мережі для кожного можливого варіанта. Відбір здійснюється за моделлю залежності між значеннями гіперпараметрів і результатами вже виконаних навчань: допоміжна модель визначає очікуване значення цільової функції для сформованих варіантів, після чого на реальне навчання передається лише один із них. Це зменшує кількість повних циклів навчань нейронної мережі та підвищує ефективність використання обчислювального бюджету.

Удосконалено метод злиття нейронних мереж у випадку, коли вихідні моделі навчені на різних підмножинах класів або даних. Метод не усереднює ваги моделей, а формує об'єднану мережу з частини нейронів першої та другої моделей, склад цієї мережі й внесок кожної моделі для різних класів визначає генетичний алгоритм. Цільова функція враховує частку правильно розпізнаних прикладів у всій валідаційній вибірці та окремо точність для найгірше розпізнаваного класу, що зменшує ризик втрати здатності розпізнавати окремі класи та допомагає зберігати знання, набуті обома моделями.

Практичне значення одержаних результатів.

Практичне значення одержаних результатів полягає у можливості використання розроблених методів і програмних реалізацій для стиснення, налаштування та злиття нейронних мереж в умовах обмеженого обчислювального бюджету.

Метод TESA-26 дає змогу автоматично визначати максимальний ступінь стиснення нейронної мережі, за якого зберігається заданий рівень якості моделі. Це практично важливо за наявності апаратних обмежень на обсяг пам'яті та обчислювальну потужність пристроїв розгортання, оскільки зменшує потребу в ручному переборі рівнів проріджування та перевірці кожного варіанта. У порівняльному дослідженні на 6 наборах даних, 12 рівнях проріджування та 1080 експериментальних записах TESA-26 показав найкращий середній ранг Фрідмана 2,04

і статистично значучо перевершив методи SparseGPT, WANDA та RIA за критерієм Фрідмана ($\chi^2 = 18,05$, $p < 0,0001$).

Метод GFCS дає змогу перетворити проріджену нейронну мережу на фізично меншу щільну архітектуру зі зменшеною кількістю нейронів і зв'язків. Це знижує обсяг моделі в пам'яті та пришвидшує обчислення на звичайному апаратному забезпеченні без спеціальної підтримки розріджених операцій. У проведених експериментах GFCS забезпечив стиснення у 7,6 раза та середнє прискорення обчислень моделі у 1,42 раза зі збереженням якості на всіх 8 досліджених наборах даних.

Методи SACMA-DAC та SACMA-MAB дають змогу скоротити кількість повних циклів навчання нейронної мережі, необхідних для добору гіперпараметрів. Це практично важливо за апаратних обмежень, коли кожна перевірка конфігурації потребує значного часу та обчислювальних ресурсів, оскільки дає змогу швидше отримати придатну конфігурацію моделі за фіксованого бюджету обчислень. У порівняльному дослідженні на 43 задачах із 10 повторами методи посіли 1–2 місця серед 9 підходів за середнім рангом 2,47 і 2,95 та досягли найвищої швидкості збіжності (AUCC = 0,9257 і 0,9210 відповідно).

Метод ENT дає змогу об'єднувати нейронні мережі, навчені на різних підмножинах класів або даних, в одну модель без повного навчання з нуля. Це практично важливо у випадках, коли вже існують спеціалізовані моделі для окремих класів або підзадач, а потрібно отримати єдину модель зі збереженням їхніх знань. У сценарії комплементарного злиття ENT забезпечив розпізнавання 10 з 10 класів, точність 0,749 і баланс 0,981, після калібрації ENT-FT точність зросла до 0,917, тоді як інтерполяційні підходи демонстрували втрату частини класів або істотно нижчу збалансованість.

Особистий внесок здобувача.

Усі основні результати дисертаційного дослідження, подані до захисту, одержані автором особисто. Здобувачем виконано: аналіз існуючих методів

структурної оптимізації нейронних мереж та еволюційних алгоритмів безперервної оптимізації, розробку методу неструктурованого проріджування TESA-26, розробку методу перетворення розріджених мереж у компактні щільні архітектури GFCS, розробку методів пошуку гіперпараметрів SACMA-DAC та SACMA-MAB, розробку методу злиття нейронних мереж з комплементарними знаннями ENT, розробку апаратно-незалежної метрики обчислювальної вартості, проведення масштабних обчислювальних експериментів та статистичний аналіз результатів. У публікаціях у співавторстві здобувачеві належать: дослідження існуючих еволюційних методів оптимізації, проектування та реалізація запропонованих методів, розробка програмних прототипів, постановка та проведення обчислювальних експериментів, статистичне оцінювання ефективності запропонованих методів.

Апробація матеріалів дисертації.

Основні результати дисертаційної роботи доповідались та обговорювались на міжнародній науково-практичній конференції:

- Science and Information Technologies in the Modern World «Використання еволюційних алгоритмів для синтезу та навчання нейронних мереж: огляд сучасного стану та перспективи розвитку» (Афіни, Греція, 24–26 грудня 2025 р.).

Публікації.

За результатами досліджень опубліковано 5 наукових публікацій, у тому числі:

- 1 стаття у науковому виданні, проіндексованому у базі Scopus;
- 3 статті у наукових фахових виданнях України;
- 1 теза виступу на науковій конференції;
- 0 статей у наукових фахових виданнях України, у яких число співавторів більше двох осіб.

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Hirianskyi B., Bulakh B. Effectiveness of evolutionary algorithms in neural network optimization tasks: Unconditional minimization, pruning, and hyperparameter optimization. *Control, Navigation and Communication Systems*, № 1, 2026, p. 45–50. DOI: 10.26906/sunz.2026.1.045. Досліджено ефективність еволюційних алгоритмів для трьох класів задач оптимізації нейронних мереж. Виконано порівняння збіжності СМА-ES та градієнтних оптимізаторів на багатомодальних функціях. Реалізовано еволюційний пошаровий розподіл розрідженості при неструктурованому проріджуванні. Здійснено оптимізацію гіперпараметрів на основі диференціальної еволюції. Експериментальні висновки цієї роботи лягли в основу обґрунтування вибору базових алгоритмів СМА-ES і диференціальної еволюції та підтвердили доцільність розроблення запропонованих у розділах 2 і 3 дисертації методів стиснення нейронних мереж і пошуку гіперпараметрів. *Особистий внесок здобувача*: спроектовано та проведено експериментальне дослідження, виконано програмну реалізацію та статистичний аналіз результатів. *Особистий внесок співавтора Булаха Б. В.*: постановка задачі порівняльного дослідження, консультації щодо вибору базових алгоритмів, обговорення результатів та редагування рукопису.
2. Гірянський Б.П., Булах Б.В. Створення нейронних мереж шляхом нейроеволюції та автоматизованого навчання з використанням еволюційних алгоритмів. *Наука та техніка сьогодні*, № 8(49), 2025, с. 1214–1227. DOI: 10.52058/2786-6025-2025-8(49)-1214-1227. Систематизовано підходи до створення та оптимізації нейронних мереж шляхом нейроеволюції. Викладені у статті результати становлять методологічну основу для запропонованих у дисертації методів стиснення (розділ 2) та злиття моделей (розділ 4). Зокрема, у статті обґрунтовано принципи еволюційного пошуку структури мережі з градієнтним донавчанням. *Особистий внесок здобувача*: запропоновано поєднання еволюційного пошуку з градієнтним донавчанням, розроблено алгоритм, виконано програмну реалізацію та експерименти. *Особистий внесок*

співавтора Булаха Б. В.: формулювання задачі поєднання еволюційного пошуку з градієнтним донавчанням, консультації щодо вибору тестових наборів даних, редагування рукопису.

3. Hirianskyi B.P., Bulakh B.V. Using evolutionary algorithms for training and optimizing neural networks. *Наука та техніка сьогодні*, № 8(36), 2024, с. 808–823. DOI: 10.52058/2786-6025-2024-8(36)-808-823. Проаналізовано класи еволюційних алгоритмів та обґрунтовано можливість їх застосування для задач оптимізації нейронних мереж. На основі експериментів встановлено залежність ефективності еволюційного пошуку від параметрів алгоритму. Це визначило вибір базових алгоритмів CMA-ES та DE для методів, запропонованих у дисертації. Експериментальна частина статті містить попередні результати, які увійшли до дисертації як обґрунтування конкурентоспроможності еволюційних алгоритмів порівняно з градієнтними оптимізаторами. *Особистий внесок здобувача*: узагальнено результати застосування еволюційних алгоритмів за публікаціями останніх років, виконано власні експерименти з підбору параметрів алгоритму, обґрунтовано доцільність застосування еволюційних алгоритмів для оптимізації нейронних мереж. *Особистий внесок співавтора Булаха Б. В.*: постановка задачі аналітичного огляду класів еволюційних алгоритмів, визначення структури огляду та критеріїв систематизації джерел, обговорення дизайну експериментів з підбору параметрів, редагування рукопису.
4. Hirianskyi B., Bulakh B. A review of practice of using evolutionary algorithms for neural network synthesis and training. *Technology Audit and Production Reserves*, № 4(2(72)), 2023, p. 22–26. DOI: 10.15587/2706-5448.2023.286278. Представлено огляд практики застосування еволюційних алгоритмів для синтезу та навчання нейронних мереж. Стаття є першою публікацією здобувача за темою дисертації, її висновки сформували початкові гіпотези дисертаційного дослідження. Систематизовані у статті переваги та обмеження існуючих підходів стали

підґрунтям для подальшого розроблення методів TESA-26 та GFCS (розділ 2 дисертації) і вибору напрямку дослідження загалом. *Особистий внесок здобувача*: проведено огляд літератури, систематизовано підходи на основі генетичних алгоритмів, виконано аналіз їх переваг та обмежень. *Особистий внесок співавтора Булаха Б. В.*: постановка завдання аналітичного огляду та визначення його тематичних меж, формулювання структури огляду та критеріїв класифікації підходів, редагування рукопису та обговорення висновків.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

5. Гірянський Б.П. Використання еволюційних алгоритмів для синтезу та навчання нейронних мереж: огляд сучасного стану та перспективи розвитку. Science and Information Technologies in the Modern World: матеріали 4-ї Міжнар. наук.-практ. конф. (м. Афіни, Греція, 24–26 груд. 2025 р.), 2025. DOI: 10.70286/isu-24.12.2025.006. Доповідь репрезентувала узагальнення основних теоретичних положень та результатів дисертації перед міжнародною науковою спільнотою. У ній представлено огляд сучасного стану застосування еволюційних алгоритмів для оптимізації нейронних мереж і окреслено перспективи розвитку напрямку, що склали основу для висновків дисертації. *Особистий внесок здобувача*: одноосібна публікація.

Структура дисертації.

Структура роботи обумовлена метою та завданнями дослідження. Дисертація складається зі вступу, чотирьох розділів, висновків до кожного розділу, загальних висновків, списку використаних джерел із 96 найменувань та 7 додатків.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ВИКОРИСТАННЯ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ ДЛЯ ОПТИМІЗАЦІЇ НЕЙРОННИХ МЕРЕЖ

У цьому розділі розглядається комплексна проблема оптимізації нейронних мереж на трьох рівнях: параметричної оптимізації через градієнтні методи, структурної оптимізації архітектури (проріджування, пошук архітектури) та мета-оптимізації гіперпараметрів. Послідовно аналізуються фундаментальні розходження між еволюційними стратегіями й адаптивними оптимізаторами, розглядаються обмеження традиційних підходів при оптимізації під обмеження обчислювальних ресурсів та апаратної залежності. Виокремлюються прогалини у дослідженнях щодо масштабованості, надійності та адаптованості методів оптимізації до гетерогенних платформ.

1.1. Проблема оптимізації в глибокому навчанні

Досягнення глибокого навчання стали можливими завдяки безпрецедентному масштабуванню моделей: сучасні великі мовні моделі (LLM) налічують від 7 до 176 млрд параметрів [1], що створює фундаментальну суперечність між зростаючою якістю та обмеженими обчислювальними ресурсами. Пошук нейроархітектури (NAS) класичними методами вимагає до 2000–3150 GPU-днів [2, 3], навіть ефективний DARTS — 0,4–4,5 GPU-днів [36], за оглядом Liu et al. [26] частка невдач високорозмірної оптимізації зросла з 22% (2020 р.) до 78% (2025 р.). Оптимізація гіперпараметрів (HPO), за своєю природою, є вкладеною (bilevel) задачею, де кожна оцінка потребує повного циклу тренування [9]. Апаратна залежність додатково ускладнює проблему: мережа, оптимізована для GPU (латентність 5,1 мс), на мобільній платформі працює з латентністю 78 мс при аналогічній точності [5].

Оптимізація нейронних мереж охоплює три рівні задач, що суттєво відрізняються за характером пошукового простору. *Параметрична оптимізація* мінімізує функцію втрат $\mathcal{L}(\theta; \mathcal{D}_{train})$ за допомогою градієнтних оптимізаторів, Li et

al. [37] формалізували концепцію неявної регуляризації для адаптивних оптимізаторів, показавши, що SGD неявно мінімізує $\text{tr}(H)$, тоді як Adam діє як мінімізатор $\text{tr}(\text{Diag}(H)^{1/2})$. До параметричної post-training корекції також належить SparseGPT [1] — метод розрідженої регресії для компенсації видалених ваг. *Структурна оптимізація* змінює макроархітектуру мережі: проріджування розглядається як комбінаторна задача мінімізації $\mathcal{L}(\theta \odot m; \mathcal{D})$ за обмеження $\|m\|_0 \leq (1 - s) \cdot n$; метод Pruner-Zero [24] автоматизує пошук масок за допомогою генетичного програмування, еволюційний NAS (G-EvoNAS [29]) досягає 97,52% точності на CIFAR-10 за 0,2 GPU-дні, DLP [38] та ICP [39] вирішують проблему міжшарового розподілу розрідженості, DARTS- [40] блокує несправедливу перевагу лінійних з'єднань. *Мета-оптимізація* (HPO/NAS) є дворівневою задачею, де еволюційні стратегії (CMA-ES) демонструють масштабну роботу до 10 000 параметрів [26], а багатокритеріальна селекція (MODNAS [41], EC-NAS [42]) дозволяє одночасно оптимізувати точність, латентність та енергоспоживання.

Практичне впровадження моделей на обладнанні з обмеженими ресурсами (мобільні пристрої, вбудовані системи, граничні сервери) зумовлене вимогами конфіденційності, мінімізації латентності та автономності. Апаратні обмеження — обсяг і пропускна здатність пам'яті (4–16 ГБ проти ~16 ГБ для Llama-3-8B у FP16), термальне дроселювання — формують потребу в методах стиснення (проріджування, квантування, дистиляція знань). Гетерогенність платформ зумовлює потребу в адаптивних методах, де еволюційні стратегії є природним інструментом, оскільки не потребують диференційовності цільової функції. Аналіз розглянутих у цьому підрозділі робіт свідчить, що оптимізація є багатовимірним процесом із серйозним розривом між математичними абстракціями (FLOPs) та фізичною спроможністю моделей на цільових пристроях.

1.2. Проблема оцінки обчислювальної вартості нейронних мереж

Оцінювання складності моделей традиційно зводиться до аналізу математичних параметрів — кількості операцій з плаваючою точкою (FLOPs) та обсягу параметрів. Проте теоретичні виміри слабо корелюють з реальним прискоренням: скорочення збережених параметрів через розрідження не завжди дає пропорційне прискорення через нерегулярність доступу до пам'яті [1]. Сучасні дослідження зміщують акцент на апаратно-залежні показники: Guo et al. [4] демонструють структурне проріджування LLM, орієнтоване на апаратно-зручний формат збереженої щільної архітектури, огляд [37] вводить метрику tokens/J, а експерименти ProxylessNAS [5] демонструють, що GPU та мобільні процесори мають фундаментально різні профілі затримок [36]. Систематизацію метрик наведено у Таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз метрик оцінювання обчислювальної вартості нейронних мереж

Група метрик	Представники	Переваги	Обмеження
Математичні	FLOPs, MACs, Параметри	Незалежність від апаратного забезпечення	Слабка кореляція з реальним енергоспоживанням
Ресурсні	GPU-дні, Пам'ять (МБ)	Простота інструментального вимірювання	Жорстка прив'язка до конкретної архітектури GPU
Апаратно-орієнтовані	Латентність (мс), Використання кешу	Висока точність для цільового пристрою	Потребує оцінки на реальному залізі під час пошуку
Енергетичні	tokens/J, energy-per-inference	Об'єктивне порівняння енергоефективності	Складність точного вимірювання та перенесення

Спільним обмеженням є фрагментарність: більшість оптимізаційних конвеєрів орієнтується або суто на математичну ефективність, або оптимізує енергоспоживання ціною ускладнення циклу пошуку. Питання створення уніфікованої системи оцінювання, яка б збалансовано враховувала ресурсоемність, час виконання та стійкість до компресії, залишається відкритим у розглянутих роботах.

1.3. Еволюційні алгоритми безперервної оптимізації

Еволюційні алгоритми є інструментом розв'язання задач безградієнтної оптимізації за значеннями цільової функції, коли обчислення градієнтів є неможливим або обчислювально недоцільним. У контексті оптимізації нейронних мереж такі ситуації виникають при налаштуванні гіперпараметрів, пошуку архітектур та стисненні моделей. Нижче розглянуто два фундаментальних класи еволюційних оптимізаторів та їхнє порівняння з градієнтними методами.

СМА-ES. Стратегія адаптації коваріаційної матриці (СМА-ES) [10] ґрунтується на вибірці кандидатів із багатовимірного нормального розподілу, коваріаційна матриця якого динамічно оновлюється на основі успішних кроків попередніх поколінь. Це забезпечує ефективну адаптацію до погано обумовлених (ill-conditioned) ландшафтів. Канонічна форма алгоритму походить від роботи Hansen та Ostermeier [43], згодом доповненої варіантом зі зменшеною часовою складністю [44] та схемою рестарту з нарощуванням популяції IPOP-СМА-ES [45]. Loshchilov та Hutter [12] адаптували СМА-ES для оптимізації гіперпараметрів глибоких нейронних мереж, продемонструвавши стійкість до зашумлень міні-батчового тренування. Для подолання квадратичної складності $\mathcal{O}(n^2)$ було запропоновано кооперативну коеволюцію з RL-декомпозицією підпросторів [17], сурогатні апроксимації на основі радіально-базисних функцій (СМА-SAO) [46] та більш широкий клас surrogate-assisted еволюційних методів [47].

Таблиця 1.2 – Порівняльний аналіз варіантів алгоритму СМА-ES

Варіант	Ключова ідея	Складність ітерації	Обмеження
Класичний СМА-ES [10]	Повна адаптація коваріаційної матриці	$\mathcal{O}(n^2)$	Непрактичний при $n > 10^4$
СМА-ES для НРО [12]	Адаптація у зашумленому просторі гіперпараметрів	$\mathcal{O}(n^2)$	Обмежений середньою розмірністю
LCC-СМА-ES [17]	Кооперативна коеволюція з RL-декомпозицією	$\mathcal{O}(kn), k \ll n$	Потребує навчання агента-диспетчера

Варіант	Ключова ідея	Складність ітерації	Обмеження
CMA-SAO [46]	Радіально-базисна сурогатна апроксимація	Зменшено кількість викликів оракула	Залежність від якості сурогату

Диференціальна еволюція. Алгоритм DE [11] генерує мутаційні вектори шляхом зважених різниць випадкових особин популяції, забезпечуючи лінійну складність $O(n)$. Критичною залежністю DE від ручного налаштування параметрів мутації F та кросоверу CR була подолана у методі SHADE [16], який використовує архів історично успішних значень параметрів для їхнього адаптивного оновлення. Разом із тим, диференціальні підходи залишаються вразливими до ротації осей координат (rotationally variant), що обмежує їх застосування у сильно залежаних ландшафтах [11, 16].

Таблиця 1.3 – Порівняльний аналіз базового DE та його адаптивного варіанту SHADE

Характеристика	DE (базовий) [11]	SHADE [16]
Складність ітерації	$O(n)$	$O(n)$ + накладні витрати архіву
Адаптація параметрів F, CR	Фіксована (ручна)	Автоматична на основі архіву успіхів
Стійкість до ротації осей	Низька	Низька (властивість класу DE)
Результат на IEEE CEC	Базовий рівень	Статистично кращий на більшості функцій

Порівняння з градієнтними методами. Градієнтні оптимізатори (SGD, Adam) забезпечують найшвидшу локальну збіжність завдяки експлуатації кривизни функції втрат [37], проте є принципово локальними шукачами, неспроможними працювати з дискретними параметрами та схильними до стагнації у сідлових точках. Еволюційні алгоритми, навпаки, здійснюють глобальну розвідку ландшафту через ранжування вибірки, не потребуючи диференційовності цільової функції. Систематизацію цих відмінностей наведено у Таблиці 1.4.

Таблиця 1.4 – Порівняльний аналіз градієнтних та еволюційних методів оптимізації

Критерій порівняння	Гradientні алгоритми (Adam, SGD)	Еволюційні алгоритми (CMA-ES, DE)
Спрямованість пошуку	Жадібний локальний пошук (Local Exploitation)	Глобальне розвідування (Global Exploration)
Вимоги до цільової функції	Суворі диференційованість	Будь-яка (підтримується Black-box та шуми)
Профіль пам'яті	$\mathcal{O}(n)$ ваг плюс моменти, Tensor-граф	$\mathcal{O}(n)$ (для DE) або $\mathcal{O}(n^2)$ (для повної матриці CMA-ES)
Швидкість локального спуску	Найвища (лінійна або надлінійна збіжність)	Експоненційно нижча порівняно з Adam
Стійкість до “шумів” та розривів	Низька, легко застрягає в сідлових точках	Висока, оминає мікрошум через ранжування вибірки
Застосовність у Deep Learning	Тотальне оновлення мільярдів вагових коефіцієнтів	Пошук гіперпараметрів, дискретний NAS або компресія

Gradientні та еволюційні алгоритми є доповнюючими: зворотне поширення помилки незамінне для параметричного навчання, тоді як еволюційні методи створюють фундамент для структурного пошуку. Фундаментальною проблемою залишається висока обчислювальна вартість повномасштабного коваріаційного моделювання на мережах надвисокої розмірності, що формує потребу у спеціалізованих техніках структурного стиснення.

1.4. Методи структурного стиснення нейронних мереж

1.4.1. Неструктуроване проріджування та конверсія у щільну архітектуру

Зменшення обсягу нейронних мереж шляхом вибіркового усунення надлишкових зв'язків поділяється на два етапи: (а) неструктуроване проріджування — обнулення довільних вагових коефіцієнтів, (б) конверсія — фізичне зменшення розмірності шарів. У Таблицях 1.5 та 1.6 систематизовано основні методи обох етапів.

Неструктуроване проріджування. Магнітудний підхід [21] видаляє ваги з найменшим абсолютним значенням, проте ігнорує фактичний вплив ваги на функцію втрат. Класична лінія однократного проріджування включає Lottery Ticket Hypothesis [48], SNIP-критерій чутливості зв'язків при ініціалізації [49], багатоетапну Deep Compression-схему [50] та аналіз геометрії втрат через Linear Mode Connectivity [51], критика розширених pruning-протоколів [52] та масштабний бенчмарк State of Pruning

[53] свідчать, що магнітудні підходи з ретренуванням залишаються конкурентними. Тейлорівський підхід (OBD) [20] формалізує вартість видалення через діагональ матриці Гессе:

$$\delta E \approx \frac{1}{2} \sum_i h_{ii} \cdot (\delta w_i)^2, \quad (1.1)$$

де h_{ii} — діагональний елемент Гессіану. Подальший розвиток — оцінювання важливості першого порядку [54] — розширив придатність підходу для конволюційних мереж. Методи однократного проріджування без донавчання — SparseGPT [1] (пошарова OBS-компенсація для LLM до 175B) та Wanda [6] (метрика $|W| \times \|X\|_2$) — масштабуються до великих мовних моделей. CVR+EC [22] вирішує проблему нерівномірного розподілу активацій через поканальне калібрування. SynFlow [23] працює без доступу до даних, оцінюючи L_1 -нормний шлях мережі. Pruner-Zero [24] автоматизує вибір метрики через генетичне програмування, конструюючи формулу $\sqrt{|W| \times |W|} \times \sigma(|G|)$, що перевершує SparseGPT та Wanda на LLaMA до 70B. Аналогічно AMC [55] застосовує навчання з підкріпленням для автоматичного підбору пошарових коефіцієнтів стиснення. Паралельною лінією зменшення обсягу LLM є квантизація: GPTQ [56], LLM.int8 [57] та AWQ [58] стискають трансформери до 3–4 біт, доповнюючи проріджування у сценаріях розгортання LLaMA-масштабних моделей [59], систематичний огляд напрямків наведено у [60, 61, 62], де також розглянуто Wanda-подібні підходи [63] та цілочисленну Quantization-Aware Training-парадигму [61].

Таблиця 1.5 – Порівняльний аналіз методів неструктурованого проріджування нейронних мереж

Метод	Критерій оцінки ваг	Потребує донавчання	Потребує дані	Масштаб моделей	Обмеження
Магнітудний [21]	Абсолютне значення ваги	Так	Так	CNN (до 10M)	Ігнорує вплив на функцію втрат

Метод	Критерій оцінки ваг	Потребує донавчання	Потребує дані	Масштаб моделей	Обмеження
OBD [20]	Діагональ Гессіана	Так	Так	MLP, CNN	Квадратична складність матриці
SparseGPT [1]	OBS + послідовна компенсація	Ні	Калібрувальні	LLM (до 175B)	Лише неструктуроване проріджування
Wanda [6]		W	$\times \ X\ _2$	Ні	Калібрувальні
CVR+EC [22]	Дисперсійне калібрування	Ні	Калібрувальні	LLM	Обмежений неструктурними шаблонами
SynFlow [23]	L ₁ -нормний шлях мережі	Ні	Ні (data-free)	CNN (VGG, ResNet)	Не апробований на LLM-масштабі
Pruner-Zero [24]	Автоматично еволюційна	Ні	Калібрувальні	LLM (до 70B)	Витрати на еволюційний пошук метрики
RIA [Zhang 2024]	Reciprocal Importance Approx.	Ні	Калібрувальні	LLM (до 70B), N:M шаблони	Конкурент Wanda на 4:8 N:M; чутливий до калібрації
MagR [MagR 2024]	Magnitude Reverse importance $\times$	Ні	Ні (data-free)	LLM	Чутливість до шкали ваг

Конверсія у щільну архітектуру. Розріджена мережа зберігає вихідний тензорний формат, альтернативний підхід — фізичне видалення цілих нейронів, каналів або шарів. SlimLLM [4] оцінює важливість голів уваги через кореляцію Пірсона та визначає ступінь стиснення за косинусною подібністю між активаціями. LLM Surgeon [64] масштабує KFAC-апроксимацію кривизни для скорельованого OBS-видалення рядків та стовпців вагових матриць. Neuron Merging [65, 66] зберігає частину сигналу видалених нейронів через зважене злиття активаційних профілів — на відміну від проріджування, де інформація втрачається безповоротно. Túr-the-Pruner [67] використовує еволюційний пошук (50 поколінь, 128 нащадків) для

нерівномірного розподілу розрідженості по шарах із механізмом очікуваного накопичення похибки, зберігаючи 97% точності Llama-3.1-70B при 50% стисненні.

Таблиця 1.6 – Порівняльний аналіз методів конверсії у компактну щільну архітектуру

Метод	Рівень структурного втручання	Механізм компенсації	Апробований масштаб	Обмеження
SlimLLM [4]	Голови уваги, канали FFN	РСА-проекція + лінійна регресія	LLaMA (7B–13B)	Нерівномірність стиснення обирається евристично
LLM Surgeon [64]	Рядки та стовпці матриць	Скорельоване OBS-оновлення (KFAC)	OPT, Llama-2 (до 13B)	Багатопрхідний розклад збільшує час
Neuron Merging [65]	Нейрони подібними профілями	Зважене злиття активацій	CNN (VGG, ResNet)	Не апробований на трансформерних моделях
Tyr-the-Pruner [67]	Цілі шари трансформера	Еволюційний пошук суперсіть +	LLM (до 70B)	Витрати еволюційного пошуку (суперсіть)

Аналіз розглянутих у цьому підрозділі робіт дозволяє констатувати: ранні методи забезпечили теоретичний фундамент, сучасні однократні підходи масштабуються до LLM, проте жоден не передбачає трансформації розрідженої архітектури у фізично компактну щільну мережу — розрідженість залишається у формі маски обнулених ваг, що зберігає первісний обсяг пам'яті.

1.4.2. Еволюційні алгоритми для стиснення та обмеження існуючих підходів

Детерміновані критерії проріджування (підрозділ 1.4.1) ефективні за умови незалежності шарів, проте в глибоких архітектурах видалення компонентів одного шару спричиняє каскадне зміщення активацій у наступних. Sieberling et al. [68] емпірично продемонстрували немонотонність помилки стиснення та запропонували EvoPress — $(1+\lambda)$ -еволюційну стратегію з level-switch мутаціями для нерівномірного розподілу розрідженості, що при 70% розрідженості на Llama-3-8B досягає точності

на 6,7 п.п. вище рівномірного розподілу. DarwinLM [69] інтегрує донавчання безпосередньо в еволюційний цикл, виявляючи підструктури з найвищим потенціалом до відновлення, і перевершує ShearedLlama при п'ятикратно меншому обсязі тренувальних даних. Pruner-Zero [24] вирішує задачу на мікрорівні — конструювання самого критерію важливості через генетичне програмування. Таким чином, еволюційний апарат застосовується на трьох рівнях: розподіл розрідженості (EvoPress), відбір підструктур (DarwinLM), синтез метрики (Pruner-Zero).

Практичне значення стиснення суттєво зросло в контексті розгортання LLM на периферійних пристроях. Guo et al. [4] (SlimLLM — структурне проріджування зі збереженням щільного формату) та Li et al. [70] показують, що ізольоване застосування окремих технік є недостатнім: необхідне спільне проєктування алгоритмів та апаратної платформи.

Систематичні обмеження. Рівномірне стиснення руйнує чутливі шари [38], DLP адаптивно перерозподіляє ступінь стиснення, знижуючи перплексію LLaMA2-7B при 70% розрідженості на 7,79. Міжшарове накопичення помилки [39] адресується ICP — ковзним вікном компенсації, що підлаштовує блок B_{i+1} після проріджування B_i . NAP-E [71] інтегрує рекурсивну оцінку чутливості та предиктор латентності, але не розглядає гібридних конвеєрів. PruneNet [72] переформулює стиснення як навчання стохастичної політики (відстань Колмогорова — Смірнова між спектральними розподілами), стискаючи LLaMA-2-7B за 15 хвилин без даних. Систематизацію обмежень наведено у Таблиці 1.7.

Таблиця 1.7 – Систематизація обмежень існуючих підходів до структурного стиснення

Тип обмеження	Сутність проблеми	Зачеплені методи	Запроповані часткові рішення
Рівномірність стиснення	Однакова розрідженість усіх шарів руйнує чутливі шари	SparseGPT, Wanda	DLP [38], OWL, EvoPress

Тип обмеження	Сутність проблеми	Зачеплені методи	Запроповані часткові рішення
Міжшарове накопичення помилки	Помилки проріджування каскадно зростають з глибиною	Усі однократні методи	ICP [39], Týr-the-Pruner
Фіксовані евристики	Ручні гіперпараметри розподілу не масштабуються	ICP, SlimLLM	HAP-E [71], PruneNet
Відсутність апаратного зворотного зв'язку	Оптимізація розрідженості без зв'язку з реальною затримкою	Більшість	HAP-E (предиктор латентності)
Ізольованість від квантування	Проріджування та квантування розглядаються окремо	Більшість	Не вирішено у доступній літературі
Відсутність синтезу нової топології	Видалення ваг без перебудови зв'язностей	Усі розглянуті	Не вирішено у доступній літературі

Аналіз розглянутих у цьому підрозділі робіт свідчить, що наявні підходи суттєво просунулися у питанні «що видаляти», проте питання «як перебудувати» мережу — синтез нової щільної топології, яка зберігає інформаційний потік без масштабного донавчання — залишається недостатньо висвітленим. Ця прогалина формує потребу в розробленні методів еволюційної конверсії розріджених архітектур (Розділ 2).

1.5. Методи оптимізації гіперпараметрів нейронних мереж

1.5.1. Статичні методи та адаптація в умовах концептуального дрейфу

Оптимізація гіперпараметрів (HPO) формулюється як вкладена (bilevel) задача: зовнішній цикл підбирає гіперпараметри λ , а внутрішній — повноцінно тренує модель, що робить кожну оцінку обчислювально дорогою. Систематизацію основних підходів наведено у Таблиці 1.8.

Випадковий пошук [73] перевершує сітковий за наявності неважливих вимірів, проте не використовує зворотний зв'язок. Байєсівська оптимізація на основі гаусівських процесів [74] балансує дослідження та експлуатацію через функцію

Expected Improvement, проте обмежена кубічною складністю $O(n^3)$. Optuna (TPE) [75] замінює гаусівський процес на деревоподібний оцінювач Парзенівського вікна з інтегрованою ранньою зупинкою (ASHA). Hyperband [76] комбінує послідовне половинення ресурсів із випадковою вибіркою для масштабного покриття простору. Carstensen et al. [77] запропонували кількість тренуваних шарів як нове джерело низькоточної апроксимації: для GPT-2 навчання 40% шарів дає рангову кореляцію $\approx 1,0$ із повним тренуванням при трикратному зменшенні пам'яті. Монографія Franceschi et al. [78] констатує, що жоден клас НРО-алгоритмів не домінує в усіх режимах.

Таблиця 1.8 – Порівняльний аналіз статичних методів оптимізації гіперпараметрів

Метод	Тип сурогату	Макс. workers	Складність ітерації	Багатоточність	Cold-start (T)	Джерело
Випадковий пошук	відсутній	n	$O(d)$	ні	0	[73]
GP-BO	гаусівський процес	1–4	$O(n^3)$	ні	~ 5	[74]
Optuna TPE	TPE + CMA-ES	1–16	$O(n \cdot d)$	ASHA	~ 5	[75]
SMAC3	випадковий ліс	1–8	$O(n \log n \cdot d)$	ні	~ 10	[Lindauer 2022]
Hyperband	— (рівні бюджету)	n	$O(d)$	SHA	0	[76]
BOHB	TPE + Hyperband	1–16	$O(n \cdot d)$	SHA	~ 5	[Falkner 2018]
DEHB	DE + Hyperband	n	$O(NP \cdot d)$	SHA	~ 3	[ICML 2021]
PriorBand	TPE + Hyperband + prior	1–16	$O(n \cdot d)$	SHA	prior	[NeurIPS 2023]
Frozen Layers	— (кореляційне наближення)	n	$O(d)$	layers	0	[77]

Адаптація в умовах дрейфу. Статичні методи передбачають знаходження єдиної оптимальної конфігурації, проте розподіл даних може змінюватися з часом —

концептуальний дрейф [79]. PBT [80] паралельно тренує популяцію агентів із періодичною заміною найгірших мутованими копіями найкращих, генеруючи адаптивні розклади гіперпараметрів, проте демонструє жадібну поведінку та згасання різноманітності. MF-PBT [81] вирішує цю проблему через підпопуляції з різною частотою селекції та асиметричну міграцію, досягаючи 23 793 балів у Humanoid проти 16 171 для PBT. SPG [82] забезпечує адаптацію на рівні окремих входів через паралельне передбачення токенів, підвищуючи точність ResNet-50 на 1,1% при скороченні бюджету на 55%. HPI-ParEGO [83] інтегрує оцінку важливості гіперпараметрів у багатоцільовий оптимізаційний цикл, динамічно зменшуючи простір пошуку.

Спільне обмеження: жоден із розглянутих методів не інтегрує детекцію типу дрейфу у механізм адаптації.

1.5.2. Еволюційні алгоритми для НРО та обмеження існуючих підходів

Ландшафт гіперпараметрів нейронних мереж характеризується мультимодальністю, зашумленістю та відсутністю градієнтів, що робить еволюційні алгоритми природними кандидатами. Loshchilov та Hutter [12] встановили, що СМА-ES у режимі масивно-паралельних обчислень (30 GPU) перевершує TPE та гаусівські процеси на 19-вимірному просторі: СМА-ES генерує λ незалежних кандидатів за ітерацію, тоді як байєсівська оптимізація потребує послідовного оновлення сурогату. СМА-SAO [46] зменшує кількість витратних обчислень у 3–6 разів через інтеграцію RBF-сурогату. Qiu et al. [84] вперше продемонстрували застосування NES до повнопараметричного донавчання моделей до 14В параметрів без зменшення розмірності, де популяція $N = 30$ виявилася достатньою, NES перевершив PPO, GRPO та Dr.GRPO на задачі Countdown та продемонстрував стійкість до reward hacking. GEGO [85] поєднує рої та генетичні оператори для НРО за 15 ітерацій, а Custode et al. [86] показали, що LLM (Llama2-70b) здатна адаптувати σ у (1+1)-ES, стабільно перевершуючи правило 1/5 успіху за рейтингом Glicko-2.

Систематичні обмеження НРО. Ізольованість простору пошуку: JAHS-Bench-201 [87] демонструє, що спільний пошук архітектурних та тренувальних параметрів у 33 рази ефективніший за ізольований НРО, JoBS [88] показує «ефект взаємодії» у 6–7 п.п. між конфігурацією даних та моделі. Однорозмірність цільової: PriMO [89] є першим НРО-методом із багатоцільовою інтеграцією апріорних знань. Обчислювальна вартість: метод заморожування шарів [77] зменшує пам'ять утримі, але не підтримує «розмороження». Систематизацію наведено у Таблиці 1.9.

Таблиця 1.9 – Систематизація обмежень існуючих підходів до оптимізації гіперпараметрів

Тип обмеження	Сутність проблеми	Зачеплені методи	Часткові рішення
Ізольований простір пошуку	НРО та NAS розглядаються окремо	Усі чисті НРО-методи	JAHS-Bench [87], JoBS [88]
Однорозмірна цільова функція	Оптимізується лише точність	CMA-ES, Hyperband, PBT	PriMO [89], HPI-ParEGO
Відсутність детекції дрейфу	Фіксована частота оновлення	PBT, MF-PBT, SPG	Не вирішено у доступній літературі
Обчислювальна вартість на LLM-масштабі	Кожна оцінка — повне тренування	Усі	Frozen Layers [77], ES at Scale
Відсутність зворотного зв'язку із стисненням	НРО та проріджування — окремі конвеєри	Усі НРО-методи	Не вирішено у доступній літературі

Аналіз розглянутих у цьому підрозділі робіт свідчить, що інтеграція НРО із процедурою стиснення в єдиний конвеєр залишається поза увагою більшості досліджень. Ця прогалина формує потребу в розробленні інтегрованих еволюційних конвеєрів (Розділ 3).

1.6. Злиття моделей нейронних мереж

1.6.1. Існуючі підходи: інтерполяція ваг та арифметика задач

Злиття моделей конструює єдину мережу, що поєднує компетенції декількох донорів на рівні параметрів — без повторного тренування. Основні підходи та їхні обмеження систематизовано у Таблиці 1.10.

Фундаментальним підходом є інтерполяція ваг. Wortsman et al. [90] запропонували «модельний суп» — зважене середнє контрольних точок, тонко налаштованих з різними гіперпараметрами, механізмом є лінійна з'єднуваність моделей у одному «басейні» функції втрат. Ilharco et al. [13] узагальнили підхід через «арифметику задач»: вектор задачі $\tau_t = \theta_{ft} - \theta_{pre}$ дозволяє додавати та видаляти компетенції через $\theta_{merged} = \theta_{pre} + \lambda \sum_t \tau_t$. TIES-Merging [14] адресує інтерференцію через обрізку малих модифікацій, усунення конфлікту знаків та злиття лише узгоджених дельт. DARE [91] випадково обнуляє 90% дельт із масштабуванням залишку, зберігаючи точність для моделей 7B. Akiba et al. [15] першими застосували CMA-ES для автоматичного підбору пошарових коефіцієнтів злиття через Evolutionary Model Merge. Окрему лінію формують методи, що враховують значущість окремих параметрів: Matena та Raffel [92] зважують інтерполяцію через діагональ матриці інформації Фішера, RegMean [93] виводить вагові матриці шляхом регуляризованої мінімізації відхилень activation-output, AdaMerging [94] навчає коефіцієнти злиття без міток через ентропійну цільову функцію.

Таблиця 1.10 – Порівняльний аналіз методів злиття моделей

Метод	Принцип	Потреба у навч. даних	Обмеження
Model Soup [90]	Зважене середнє контрольних точок	Ні	Моделі з одного претренованого шляху
Task Arithmetic [13]	Додавання векторів задач	Ні	Інтерференція різних задач
TIES-Merging [14]	Обрізка + усунення конфлікту знаків	Ні	Евристичні пороги
DARE [91]	Випадкове обнулення + масштабування	Ні	Гіперпараметр — частка обнулення
Evol. Merge [15]	CMA-ES для пошарових коефіцієнтів	Валідаційна вибірка	Обчислювальна вартість еволюції
Fisher-Weighted [92]	Зважування через діагональ Фішера	Ні	Точність діагональної апроксимації
RegMean [93]	Регуляризована мінімізація activation-output	Калібрувальна	Чутливість до композиції тренувальних шарів

Метод	Принцип	Потреба у навч. даних	Обмеження
AdaMerging [94]	Ентропія як проксі для коефіцієнтів	Ні	Лише класифікаційні задачі

1.6.2. Обмеження та обґрунтування еволюційного підходу до топологічного злиття

Методи злиття (підрозділ 1.6.1) засновані на припущенні, що параметри донорів знаходяться у одному лінійному підпросторі. Це порушується у двох сценаріях: (1) *комплементарність знань* — коли донори спеціалізуються на різних доменах, лінійна комбінація «розмиває» обидві спеціалізації, TIES-Merging [14] демонструє деградацію при >4–5 задачах, а Tang et al. [95] адресують її через Weight-Ensembling Mixture-of-Experts із вентиляційною маршрутизацією донорів, (2) *структурна гетерогенність* — усі методи вимагають ідентичної архітектури донорів, виключаючи злиття моделей різних розмірів. DOTRESIZE [96] показує, що дискретний оптимальний транспорт (перерозподіл, а не видалення) при 20% стисненні ширини Mistral-12B покращує точність на 11,7 п.п. порівняно зі SliceGPT.

Формально обмеження класифікуються за трьома вимірами: (1) лінійність комбінації — нелінійні залежності ігноруються [88], (2) homomorphism constraint — потрібна точна відповідність архітектурних позицій, (3) відсутність ітеративної зворотної перевірки (виняток — Evolutionary Model Merge [15]).

Обґрунтування еволюційного підходу. Еволюційні алгоритми природно задовольняють вимогам топологічного злиття: працюють із дискретними та змішаними просторами пошуку, не потребують градієнтів, забезпечують паралельне оцінювання альтернатив. Akiba et al. [15] продемонстрували доцільність еволюційного підходу (CMA-ES для пошарових коефіцієнтів та порядку шарів), а DarwinLM [69] — що еволюційний пошук структурних рішень з оцінкою через перплексію перевершує евристичні правила. Перенесення цього принципу на злиття означає пошук не лише «які шари взяти», а й «в якому порядку розмістити» та «як перерозподілити сигнали» за допомогою операцій оптимального транспорту [96].

Невирішені проблеми: обчислювальна вартість оцінки кандидатів для мільярдопараметричних моделей, несумісність розмірностей прихованих станів між різними архітектурами, експоненціальне зростання простору пошуку з кількістю донорів.

Аналіз розглянутих у цьому підрозділі робіт свідчить, що еволюційний підхід є єдиним доступним методом для топологічного злиття моделей з потенційно різними архітектурами. Разом із невирішеними задачами стиснення (підрозділ 1.4) та НРО (підрозділ 1.5), ці прогалини формують єдиний дослідницький вектор дисертації.

1.7. Визначення прогалин у існуючих дослідженнях та постановка задач

Систематичний аналіз наукової літератури (підрозділи 1.1–1.6) дозволяє виокремити три класи прогалин, які формують дослідницький вектор дисертації.

Прогалина 1: обмеженість методів пошарового розподілу розрідженості та відсутність зворотної конверсії.

Існуючі методи пошарового розподілу розрідженості (підрозділ 1.4) використовують або диференційовані градієнтні підходи (DSA [8]), або аналітичні формули фіксованого розподілу (ERK [25], LAMP [7]) без урахування зміни значущості ваг у процесі послідовного видалення. Еволюційні методи (EvoPress [68], DarwinLM [69]) шукають оптимальний розподіл, проте не включають ітеративну перекалібрацію значущості. Крім того, задача зворотної конверсії розрідженої мережі у компакту щільну архітектуру без доступу до навчальних даних залишається невирішеною. **Потреба:** розроблення еволюційного методу пошарового розподілу розрідженості з ітеративною перекалібрацією та методу конверсії розрідженої мережі у щільну архітектуру на основі еволюційної стратегії.

Прогалина 2: нерозкритий потенціал еволюційних алгоритмів для оптимізації гіперпараметрів.

Еволюційні алгоритми демонструють стійкість до зашумлених та мультимодальних ландшафтів (підрозділ 1.3), що робить їх перспективним

інструментарієм для пошуку гіперпараметрів нейронних мереж. Проте питання пошуку гіперпараметрів на основі СМА-ES із сурогатною моделлю та контролем її надійності залишається невисвітленим. **Потреба:** дослідження ефективності еволюційних алгоритмів для пошуку гіперпараметрів та розроблення методів пошуку із сурогатною моделлю на основі СМА-ES з механізмами контролю надійності цієї моделі.

Прогалина 3: непридатність методів злиття для моделей з комплементарними знаннями.

Методи злиття (підрозділ 1.6) обмежені скалярною інтерполяцією ваг (Task Arithmetic [13], TIES-Merging [14]) та еволюційною оптимізацією скалярних коефіцієнтів (Sakana-CMA [15]). Ці підходи ефективні для моделей із перекриваючими знаннями, проте можуть спричиняти втрату здатності розпізнавати окремі класи при злитті моделей з комплементарними (взаємодоповнювальними) знаннями. **Потреба:** еволюційний механізм відбору нейронної топології з покласним зважуванням виходів субмереж.

Постановка задач дисертаційного дослідження.

На основі виявлених прогалин сформульовано наступні завдання:

1. Проаналізувати існуючі методи структурної оптимізації нейронних мереж, еволюційні алгоритми безперервної оптимізації та методи злиття моделей, визначити їх переваги та обмеження.
2. Дослідити ефективність еволюційних алгоритмів для задач оптимізації нейронних мереж порівняно з градієнтними методами та визначити межі їх застосовності.
3. Розробити метод пошарового розподілу розрідженості нейронної мережі на основі еволюційного алгоритму з ітеративною перекалібрацією значущості ваг.
4. Розробити метод конверсії розрідженої нейронної мережі у компактну щільну архітектуру на основі еволюційної стратегії синтезу зв'язності без доступу до навчальних даних.

5. Дослідити ефективність еволюційних алгоритмів як інструменту оптимізації гіперпараметрів запропонованих методів стиснення та злиття моделей.
6. Розробити метод злиття нейронних мереж з комплементарними знаннями на основі генетичного алгоритму з еволюційним відбором нейронної топології.
7. Виконати формалізований опис архітектури та компонентів запропонованих методів.
8. Реалізувати програмні прототипи для експериментальної перевірки запропонованих методів.
9. Провести експериментальне дослідження та порівняльний аналіз із існуючими підходами із застосуванням непараметричних статистичних критеріїв на множині наборів даних та архітектур.

Висновки до розділу 1

У першому розділі проведено систематичний аналіз наукової літератури за п'ятьма взаємопов'язаними напрямками: (1) еволюційні алгоритми безперервної оптимізації, (2) структурне стиснення нейронних мереж, (3) оптимізація гіперпараметрів, (4) злиття моделей, (5) апаратно-незалежні метрики обчислювальної вартості.

Аналіз показав, що кожен із зазначених напрямків досяг значного прогресу: методи проріджування (SparseGPT, Wanda, DarwinLM) дозволяють зменшувати моделі у 2–5 разів без суттєвої деградації, баєсівська оптимізація та Hyperband забезпечують автоматичний підбір гіперпараметрів, методи арифметики задач (TIES, DARE) уможливають об'єднання компетенцій без повторного тренування. Еволюційні алгоритми (CMA-ES, DE, генетичні алгоритми) продемонстрували конкурентоспроможність у кожному з цих напрямків.

Разом з тим, виявлено три фундаментальні прогалини: (1) існуючі методи пошарового розподілу розрідженості використовують або диференційовані градієнтні підходи, або аналітичні формули фіксованого розподілу без урахування зміни

значущості ваг у процесі проріджування, а задача зворотної конверсії розрідженої мережі у компактну щільну архітектуру без перенавчання залишається невирішеною, (2) потенціал еволюційних алгоритмів для пошуку гіперпараметрів нейронних мереж із сурогатною моделлю та контролем її надійності залишається нерозкритим, (3) методи злиття моделей обмежені скалярною інтерполяцією ваг та непридатні для злиття моделей з комплементарними знаннями, де вони можуть спричиняти втрату здатності розпізнавати окремі класи.

Ці прогалини визначили напрямки дисертаційного дослідження: розроблення еволюційних методів неструктурного розрідження, структурного стиснення, оптимізації гіперпараметрів та злиття нейронних мереж, де еволюційний алгоритм виступає метаоптимізатором комбінаторних аспектів оптимізації (пошарова алокація, синтез зв'язності, маскування нейронів, пошук конфігурацій).

РОЗДІЛ 2. ЕВОЛЮЦІЙНЕ СТИСНЕННЯ НЕЙРОННИХ МЕРЕЖ

У цьому розділі обґрунтовується конкурентоспроможність еволюційних алгоритмів порівняно з градієнтними методами в задачах стиснення нейронних мереж шляхом введення метрики відносних обчислювальних одиниць (RCU) як універсального способу нормалізації обчислювальної вартості. Послідовно розроблюються еволюційні методи неструктурованого проріджування, включаючи еволюційну апроксимацію матриці Гессе та пошаровий розподіл розрідженості, а також методи конверсії розріджених мереж у компактні щільні архітектури без доступу до тренувальних даних. Демонструється, що еволюційні підходи забезпечують переконливу перевагу при обмеженості обчислювальних ресурсів.

2.1. Середовище виконання експериментів та метрика обчислювальної вартості

2.1.1. Протокол вимірювання відносних обчислювальних одиниць (RCU)

Порівняльна оцінка обчислювальної вартості алгоритмів оптимізації нейронних мереж є необхідною умовою для обґрунтованого вибору між методами. Два алгоритми, що досягають однакової точності, можуть відрізнятися за обчислювальними витратами на порядки, і без кількісної характеристики цих витрат будь-яке порівняння залишається неповним.

Традиційно обчислювальну вартість вимірюють або абсолютним часом виконання (англ. wall-clock time), або процесорним часом (англ. CPU time). Обидва підходи мають принципове обмеження: значення, отримані на одному апаратному забезпеченні, не є безпосередньо порівнюваними з результатами, отриманими на іншому. Метод, що працює 120 секунд на GPU Nvidia A100, може потребувати 45 хвилин на споживчому прискорювачі, і зворотно — алгоритм, оптимізований під конкретну мікроархітектуру, може демонструвати непропорційне прискорення на ній. Крім того, абсолютні часові метрики є чутливими до фонових процесів операційної

системи, завантаженості оперативної пам'яті та планувальника потоків, що робить результати нестабільними навіть при повторних запусках на тому самому обладнанні.

У даній роботі для вимірювання обчислювальної вартості алгоритмів запропоновано використання відносних обчислювальних одиниць (англ. Relative Compute Units, RCU) — безрозмірної метрики, що нормалізує час виконання цільового алгоритму відносно часу виконання стандартної еталонної операції на тому самому обладнанні. RCU визначається як

$$RCU = \frac{T_{\text{method}}}{T_{\text{anchor}}}, \quad (2.1)$$

де T_{method} — час виконання цільового алгоритму (проріджування, оцінка гіперпараметрів, злиття моделей тощо), виміряний у секундах, T_{anchor} — середній час виконання фіксованої еталонної операції (калібрувальне обчислення), виміряний безпосередньо перед цільовим алгоритмом на тому самому обладнанні в тих самих умовах.

Значення $RCU = 10$ означає, що цільовий алгоритм потребує в 10 разів більше обчислень, ніж еталонна операція. Оскільки чисельник і знаменник вимірюються на одній і тій самій платформі, систематичні апаратні відмінності скорочуються, і метрика стає незалежною від конкретного обладнання.

Еталонна операція (anchor). Як еталонну операцію обрано виконання фіксованої кількості тренувальних пакетів (англ. training batch) базової нейронної мережі. Ця операція включає прямий та зворотний прохід (англ. forward pass та backward pass) через мережу, що є типовим навантаженням для більшості методів, порівнюваних у дисертації. Вибір тренувального пакету як еталону, а не синтетичної операції (наприклад, матричного множення), зумовлений тим, що структура навантаження еталону має бути якомога ближчою до структури навантаження вимірюваних алгоритмів — це зменшує вплив мікроархітектурних ефектів (кешування, передвибірка тощо) на результат нормалізації.

Протокол вимірювання. Процедура вимірювання RCU складається з трьох фаз (рисунок 2.1):

1. *Фаза калібрування.* Виконується 5 прогрівних (англ. warm-up) запусків еталонної операції, результати яких відкидаються. Мета цієї фази — вивести процесор у стабільний тепловий та частотний режим. Далі виконується 30 основних замірів еталонної операції. З них обчислюються середнє значення $\bar{T}_{\text{anchor}}$ і коефіцієнт варіації (KB). Якщо KB перевищує 5%, серію позначено як нестабільну.
2. *Фаза вимірювання.* Цільовий алгоритм запускається в ідентичних умовах: однопотоковий режим (`OMP_NUM_THREADS=1`, `torch.set_num_threads(1)`), фіксоване значення генератора випадкових чисел (`seed = 42`, зафіксовано через `torch.manual_seed`, `numpy.random.seed` та `random.seed`), ті самі вхідні дані. Час кожного запуску фіксується через `time.perf_counter` — таймер з субмікросекундною роздільною здатністю на рівні ядра операційної системи.
3. *Фаза валідації.* Перевіряється стабільність ($\text{KB} < 5\%$), лінійність масштабування (коефіцієнт детермінації $R^2 > 0,99$ при кратному збільшенні навантаження) та дискримінаційна здатність (d Коена $> 0,8$ між суттєво різними навантаженнями).

Апаратне середовище. Усі експерименти, описані в розділах 2–4, виконано на єдиній апаратній платформі: портативний комп'ютер Apple MacBook Air (M2, 2022) з процесором Apple M2 (4 ядра продуктивності + 4 ядра ефективності, ARM64), 16 ГБ уніфікованої оперативної пам'яті. Усі обчислення виконувалися на центральному процесорі в однопотоковому режимі. Графічний прискорювач Apple M2 (8 ядер) використовувався виключно в одному порівняльному експерименті (Ex09), де досліджувалася різниця між прискореним та послідовним виконанням.

Програмне середовище включає Python 3.11, PyTorch 2.1 з обчислювальним бекендом MPS, бібліотеки NumPy та SciPy для чисельних обчислень і статистичного аналізу, scikit-learn для обчислення метрик класифікації, руста для реалізації СМА-

ES, та DEAP для генетичного програмування. Візуалізація здійснювалась засобами matplotlib з налаштуванням для чорно-білого друку формату A4.

Статистичний протокол. Для порівняння алгоритмів у всіх експериментах застосовано єдиний статистичний протокол: - непараметричний тест Фрідмана для перевірки глобальної гіпотези про рівність рангів алгоритмів, - post-hoc тест Немені для попарних порівнянь з корекцією на множинність, - непараметричний бутстреп-метод (англ. bootstrap) для побудови 95% довірчих інтервалів (10 000 ітерацій), - розмір ефекту через d Коена (англ. Cohen's d) та $\hat{A}_{12}$ Варги — Дилані (англ. Vargha — Delaney $\hat{A}_{12}$), - рівень значущості $\alpha = 0,05$ у всіх тестах.

Обрання непараметричних критеріїв зумовлено невеликими обсягами вибірок та відсутністю апіорних підстав стверджувати нормальність розподілів у досліджуваних задачах.

Обґрунтування вибірки наборів даних. Для верифікації методів TESA-26 та GFCS у підрозділі 2.4 сформовано вибірку з восьми наборів даних (moons, circles, spirals, blobs, gaussian_q, classification, highdim, sequence_cls), що покривають типові класи задач, на які орієнтовано методи стиснення нейронних мереж. Вибірка структурована за двома категоріями:

- *Синтетичні набори з контрольованою закономірністю* (moons, circles, spirals, blobs, gaussian_q, classification). Ці набори згенеровано з апіорно відомих геометричних та статистичних розподілів (sklearn.datasets), що дозволяє ізолювати вплив проріджування та конверсії від випадкових ефектів складності реальних даних. Контрольовані параметри генерації (рівень шуму, розмір класів, лінійна/нелінійна розділюваність) забезпечують відтворюваність експерименту на довільному обладнанні та виключають вплив особливостей конкретного набору на результати порівняння.
- *Реальні класифікаційні задачі* (highdim — 50-вимірна задача класифікації з корельованими ознаками, sequence_cls — послідовнісна класифікація у 16-

вимірному просторі). Ці набори обрано як типових представників задач, з якими стикаються методи стиснення нейронних мереж у практичному застосуванні: високовимірні табличні дані та обробка послідовностей.

Додатково для верифікації конверсії CNN та ResNet-архітектур (підрозділ 2.4.5) використано набір FashionMNIST як типового представника задач комп'ютерного зору з помірною складністю (10 класів, 28×28 пікселів, 60 000 навчальних прикладів). Вибір FashionMNIST, а не MNIST, зумовлений тим, що FashionMNIST вважається складнішим бенчмарком при збереженні тієї ж розмірності входу [LOOKUP NEEDED — Xiao 2017].

Кількість 8 наборів даних обрано так, щоб одночасно: (а) забезпечити статистично значущі висновки за непараметричними критеріями Фрідмана ($k \geq 3$ методів, $n \geq 5$ наборів) та Вілкоксона з поправкою Голма для попарних порівнянь, (б) покрити основні типи задач (двовимірні нелінійно роздільні, високовимірні табличні, послідовнісні, зорові), на яких застосовуються методи стиснення нейронних мереж у промислових застосунках, (в) включити як набори, де закономірність відома апіорі (контрольоване тестування), так і реальні задачі (практична значущість), що відповідає рекомендаціям Demšar (2006) для порівняльних досліджень класифікаторів.

2.1.2. Оцінка стабільності метрики в умовах змінного навантаження

Перед використанням метрики RCU як інструмента порівняння алгоритмів необхідно експериментально верифікувати дві ключові властивості: (1) стійкість до зовнішніх збурень (фонове навантаження на процесор, конкуренція за пам'ять) та (2) лінійність масштабування при кратному збільшенні обчислювального навантаження. Без такої верифікації використання будь-якої часової метрики для порівняння алгоритмів не має достатнього наукового обґрунтування.

Протокол валідаційного експерименту.

Валідаційний експеримент (Ex00) побудовано як набір з чотирьох тестів, що систематично перевіряють поведінку метрики RCU у порівнянні з трьома

альтернативними часовими метриками: реальним часом (англ. wall-clock time), процесорним часом (англ. process time) та часом потоку (англ. thread time).

Тест 1: стабільність під стресовими умовами. Для кожного з чотирьох типів обчислювального навантаження — ALU-інтенсивне (обчислення у кеш-пам'яті L1, масив ~ 24 КБ), FFT (швидке перетворення Фур'є, $O(n \log n)$, $n = 32\,768$), GEMM (матричне множення, 256×256), та Mixed (комбінація сортування і тригонометричних функцій) — виконано 30 замірів (після 5 прогрівних) у п'яти умовах: чисті умови (без фонового навантаження), фонове навантаження CPU (4 конкуруючі потоки арифметичних обчислень), фонове навантаження пам'яті (4 потоки алокації/деалокції ~ 40 МБ), змішане фонове навантаження (4 потоки CPU + пам'ять), та інтенсивне навантаження (8 потоків CPU). Загальний обсяг — 600 замірів.

Тест 2: лінійність масштабування. Навантаження збільшувалось у 1, 2, 3, 4, 5, 6, 7 та 8 разів відносно базового рівня. Для кожного множника виконано 30 замірів. Лінійність оцінювалась через коефіцієнт детермінації R^2 лінійної регресії. Загальний обсяг — 240 замірів.

Тест 3: інваріантність на гетерогенних ядрах. Процесор Apple M2 містить ядра двох типів: 4 високопродуктивні (англ. Performance, P-ядра) та 4 енергоефективні (англ. Efficiency, E-ядра). Абсолютна продуктивність E-ядер у 4–7,5 рази нижча за P-ядра. Через macOS QoS API (`pthread_set_qos_class_self_np`) забезпечувалось примусове розміщення потоку на P-ядрах (пріоритет Interactive), E-ядрах (пріоритет Background) або за замовчуванням (Default). Для кожної комбінації 4 навантажень $\times$ 3 типи ядер виконано 30 замірів. Загальний обсяг — 360 замірів.

Тест 4: статистична значущість. Для кожної пари «чисті умови — стресова умова» виконано тест Вілкоксона з обчисленням розміру ефекту (d Коена та $\hat{A}_{12}$ Варги — Дилані) та бутстреп 95% довірчих інтервалів (10 000 ітерацій).

Результати.

Стійкість до фонових збурень. У таблиці 2.1 наведено зсув середнього значення кожної метрики під стресовими умовами відносно чистих умов. RCU

демонструє зсув від 1,0% до 14,1%, тоді як альтернативні метрики зазнають зсувів на порядки більших.

Таблиця 2.1 – Зсув середнього значення метрик обчислювальної вартості під стресовими умовами відносно чистих умов, %

Стрессова умова	RCU	Реальний час	Процесорний час	Час потоку
CPU (4 потоки)	14,1	73,0	233,7	47,8
Пам'ять (4 потоки)	1,0	868,0	959,4	3,4
Змішане (4 потоки)	4,3	61,6	653,8	49,4
Інтенсивне (8 потоків)	12,8	74,6	230,6	45,9

Найпоказовішим є порівняння в умовах навантаження на пам'ять: процесорний час зсувається на 959,4%, реальний час — на 868,0%, тоді як RCU — лише на 1,0%. Це пояснюється тим, що нормалізація через локальний еталон компенсує системні збурення: якщо фонове навантаження сповільнює і цільовий алгоритм, і еталонну операцію однаковою мірою, їхнє відношення залишається стабільним.

На рисунку 2.1 наведено візуалізацію розподілу зсуву метрик. RCU (зелений) ледь помітний на тлі абсолютних метрик, зсув яких сягає сотень відсотків.

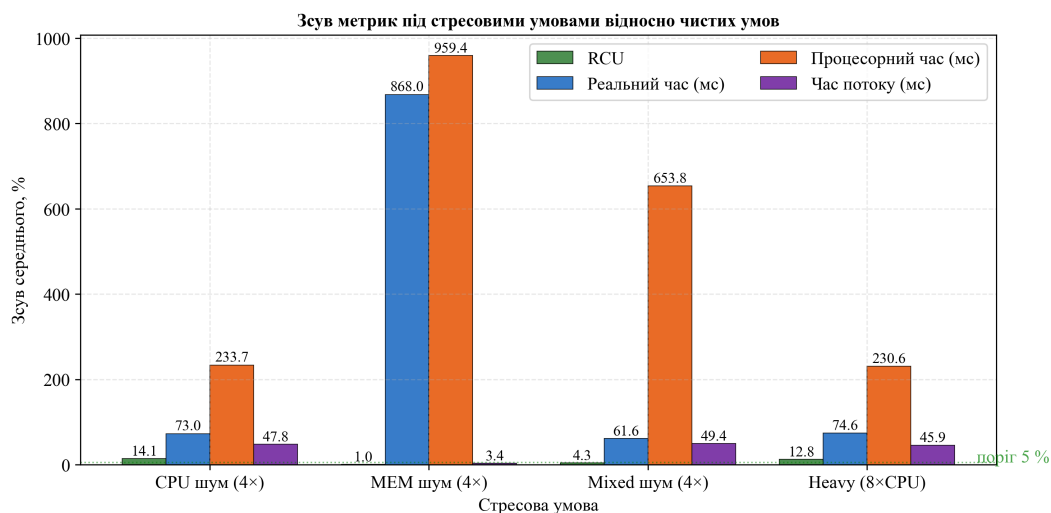


Рисунок 2.1 – Зсув метрик обчислювальної вартості під стресовими умовами відносно чистих умов

Статистична значущість розмірів ефекту. У таблиці 2.2 наведено результати тесту Вілкоксона та розміри ефекту для порівняння стресових умов із чистими.

Таблиця 2.2 – Розміри ефекту (d Коена) при порівнянні стресових умов із чистими

Стрессова умова	RCU	Реальний час	Процесорний час
CPU (4 потоки)	0,158 (малий)	0,515 (середній)	0,944 (великий)
Пам'ять (4 потоки)	0,011 (нехтовний)	2,122 (великий)	2,778 (великий)
Змішане (4 потоки)	0,046 (нехтовний)	0,497 (середній)	1,289 (великий)
Інтенсивне (8 потоків)	0,140 (малий)	0,534 (середній)	0,964 (великий)

Розмір ефекту d Коена для RCU не перевищує 0,158 у жодній стресовій умові (нехтовний або малий за класифікацією Коена), тоді як для процесорного часу d сягає 2,778 (великий). Це означає, що розподіли RCU в чистих і стресових умовах практично не відрізняються, тоді як розподіли абсолютних метрик зсуваються суттєво.

У змішаних умовах стресу тест Вілкоксона для RCU дає $p = 0,5297$ (не значущо при $\alpha = 0,05$), підтверджуючи відсутність статистично значущої відмінності між замірами RCU у чистому та змішаному середовищі. Для процесорного часу в тих самих умовах $p < 0,0001$.

На рисунку 2.2 наведено стабільність еталонного обчислення за різних умов. У чистих умовах коефіцієнт варіації (КВ) еталону не перевищує 2,8% для жодного типу навантаження. При інтенсивному фоновому навантаженні КВ зростає до 4,2%, залишаючись у прийнятних межах ($< 5\%$).

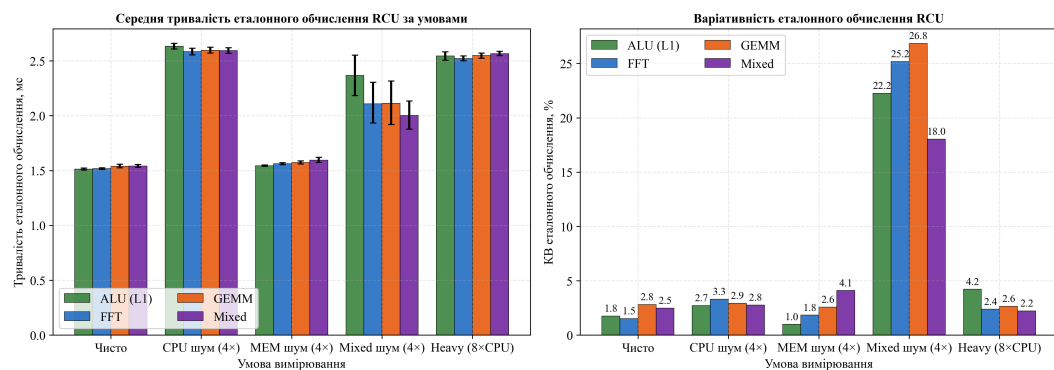


Рисунок 2.2 – Стабільність еталонного обчислення RCU: середня тривалість та коефіцієнт варіації за п'ятьма умовами стресу

Лінійність масштабування. При кратному збільшенні обчислювального навантаження RCU демонструє коефіцієнт детермінації $R^2 = 0,9999$ з нахилом 5,184

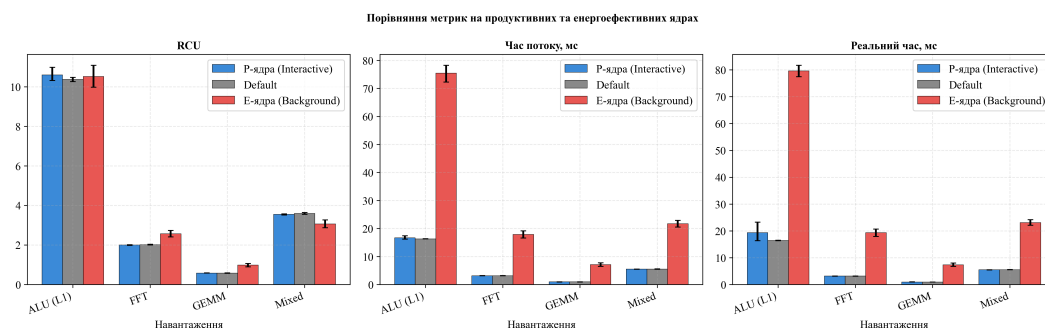
та стандартною похибкою нахилу 0,017. Це підтверджує практично ідеальну лінійну пропорційність між фактичним обчислювальним навантаженням та значенням метрики.

Таблиця 2.3 – Лінійність масштабування метрик обчислювальної вартості

Метрика	R^2	Нахил	Стандартна похибка нахилу
RCU	0,9999	5,184	0,017
Реальний час	0,9937	8,361	0,272
Процесорний час	0,9993	8,075	0,085

Усі три метрики демонструють високу лінійність ($R^2 > 0,99$), однак RCU має найменшу стандартну похибку нахилу (0,017 проти 0,272 для реального часу), що свідчить про найвищу відтворюваність.

Інваріантність на гетерогенних ядрах. На рисунку 2.3 наведено порівняння метрик на продуктивних (P) та енергоефективних (E) ядрах процесора Apple M2. Абсолютний час потоку на E-ядрах у 3,96–7,57 рази вищий, ніж на P-ядрах, що відображає різницю в мікроархітектурі. Для ALU-інтенсивного навантаження зсув RCU між P та E-ядрами становить лише 0,8% (d Коена = 0,06), а для змішаного — 13,5%.



Рисунк 2.3 – Порівняння метрик обчислювальної вартості на продуктивних та енергоефективних ядрах процесора Apple M2

Встановлено обмеження: для BLAS-інтенсивних операцій (матричне множення, GEMM) зсув RCU між типами ядер сягає 69,5% ($d = 2,35$). Це пояснюється тим, що реалізації BLAS використовують внутрішній паралелізм, який порушує однопотоковість еталонних замірів. Для основних експериментів дисертації

(проріджування, оптимізація гіперпараметрів, злиття моделей) це обмеження несуттєве, оскільки основне навантаження складається з послідовних прямих та зворотних проходів через мережу з примусовим обмеженням `OMP_NUM_THREADS=1`.

Метрику відносних обчислювальних одиниць (RCU) валідовано на 1200 замірах у чотирьох тестових сценаріях. Встановлено, що:

1. RCU стійка до фонових збурень: максимальний зсув становить 14,1% проти 959,4% для процесорного часу, а розмір ефекту d Коена не перевищує 0,158 (нехтовний–малий).
2. Метрика демонструє практично ідеальну лінійність масштабування ($R^2 = 0,9999$).
3. На однорідних ядрах (ALU-навантаження) RCU інваріантна до типу ядра (зсув 0,8%), для BLAS-інтенсивних операцій на гетерогенних ядрах зсув сягає 69,5%, що є задокументованим обмеженням.

На підставі цих результатів RCU використовується як основна метрика обчислювальної вартості в усіх подальших порівняннях алгоритмів.

2.2. Обґрунтування конкурентоспроможності еволюційних алгоритмів

2.2.1. Порівняльний протокол дослідження на багатомодальних функціях

Перед розробкою еволюційних методів стиснення нейронних мереж необхідно емпірично встановити межі застосовності еволюційних алгоритмів (ЕА) порівняно з градієнтними оптимізаторами. Огляд літератури (підрозділ 1.2) показав, що ЕА демонструють переваги на багатомодальних задачах завдяки популяційному пошуку, проте їхня ефективність залежить від розмірності та структури простору пошуку. Для встановлення цієї межі сформульовано дві гіпотези, що послідовно перевіряються серією з трьох експериментів:

- H_1 : ЕА статистично значуще перевершують градієнтні методи на багатомодальних функціях низької розмірності завдяки популяційному механізму пошуку;

- H_2 : виявлена перевага ЕА не зберігається при переході до оптимізації ваг нейронних мереж через високу розмірність та переважно гладкий ландшафт функції втрат.

Постановка задачі.

Як тестову задачу для перевірки H_1 обрано функцію Растрігіна розмірності $d = 10$:

$$f(x) = 10d + \sum_{i=1}^d (x_i^2 - 10\cos(2\pi x_i)), \quad x \in [-5,12, 5,12]^d, \quad (2.2)$$

де d — розмірність простору пошуку, x_i — i -та координата вектора параметрів.

На рисунку 2.4 наведено одновимірний зріз функції Растрігіна, що ілюструє характерну багатомодальну структуру з регулярно розташованими локальними мінімумами.

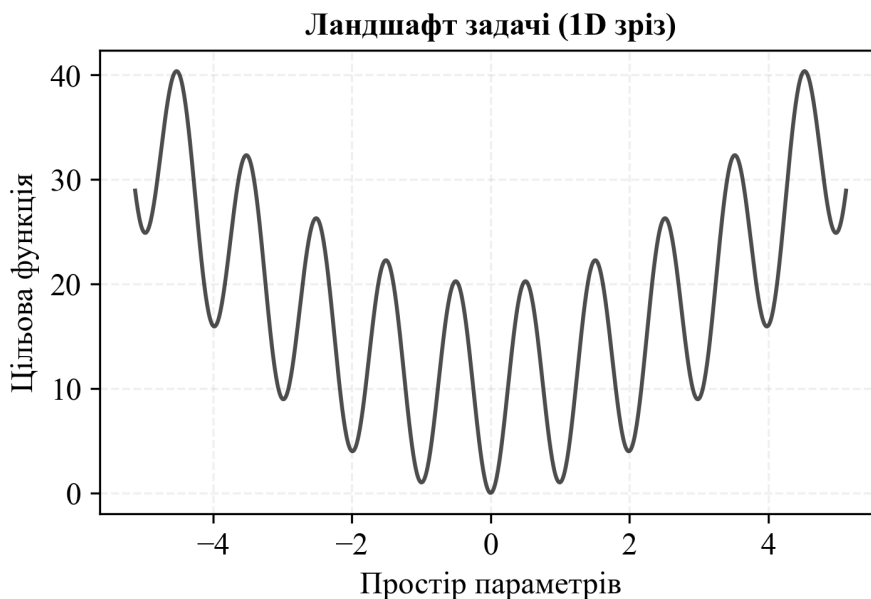


Рисунок 2.4 – Ландшафт функції Растрігіна (одновимірний зріз, $d = 1$):

багатомодальна структура з глобальним мінімумом в початку координат

Глобальний мінімум $f(0) = 0$ досягається в початку координат. Функція характеризується приблизно 10^d локальних мінімумів з регулярною структурою, що

робить її стандартним випробуванням для оцінки здатності алгоритмів до глобального пошуку [97].

Порівнювані методи.

У порівняльному дослідженні беруть участь чотири методи:

1. *CMA-ES* — еволюційна стратегія з адаптацією коваріаційної матриці [10]: популяція з 32 особин, початкова дисперсія $\sigma_0 = 2,0$. Алгоритм адаптивно змінює коваріаційну матрицю розподілу пошуку, що дозволяє ефективно досліджувати багатомодальний ландшафт.
2. *AdamW* з *множинними перезапусками (MR)* — градієнтний оптимізатор з роз'єднаною регуляризацією ваг [98]: швидкість навчання $\eta = 0,1$, коефіцієнт регуляризації $\lambda = 0,01$. Загальний обчислювальний бюджет у 3000 оцінок цільової функції (англ. function evaluations, FE) розподілено між 10 незалежними запусками з випадкових стартових точок по 300 FE на кожний перезапуск. З усіх результатів обирається найкращий.
3. *AdaBelief* з *множинними перезапусками (MR)* — адаптивний оптимізатор, що враховує різницю між спостережуваним градієнтом та його очікуваним значенням [99]: $\eta = 0,1$, $\varepsilon = 10^{-16}$, аналогічна стратегія перезапусків.
4. *Випадковий пошук (RS)* — рівномірна вибірка точок у допустимій області як контрольний метод.

Стратегію множинних перезапусків обрано для забезпечення чесного порівняння з популяційним CMA-ES: обидва підходи оцінюють множину точок у просторі пошуку, але різними механізмами — CMA-ES через популяцію, градієнтні методи через незалежні перезапуски.

Протокол вимірювань.

Обчислювальний бюджет становить 3000 оцінок цільової функції на один запуск. Виконано $N = 20$ незалежних повторів для кожного методу з фіксованими значеннями генератора випадкових чисел (42, 43, ..., 61). Методи запускались

почергово (один запуск кожного методу послідовно) для усунення впливу перегрівання процесора на результати порівняння.

Як основна метрика якості використовується фінальне значення цільової функції (мінімальне значення $f(x)$, досягнуте в межах обчислювального бюджету). Обчислювальна вартість вимірюється в одиницях RCU відповідно до протоколу, описаного в підрозділі 2.1.1. Статистична значущість оцінюється за допомогою тесту Фрідмана з post-hoc тестом Немені ($\alpha = 0,05$) та розміром ефекту $\hat{A}_{12}$ Варги — Дилані.

Результати.

На рисунку 2.5 наведено криві збіжності чотирьох методів. CMA-ES (зелений) демонструє характерну поведінку: повільний старт протягом перших 1500 FE, що відповідає фазі адаптації коваріаційної матриці, після чого відбувається різке зниження значення цільової функції до рівня ~ 10 . Градієнтні методи з перезапусками виявляють характерні стрибки кожні 300 FE (межі перезапусків), а їхній прогрес обмежений якістю стартових точок.

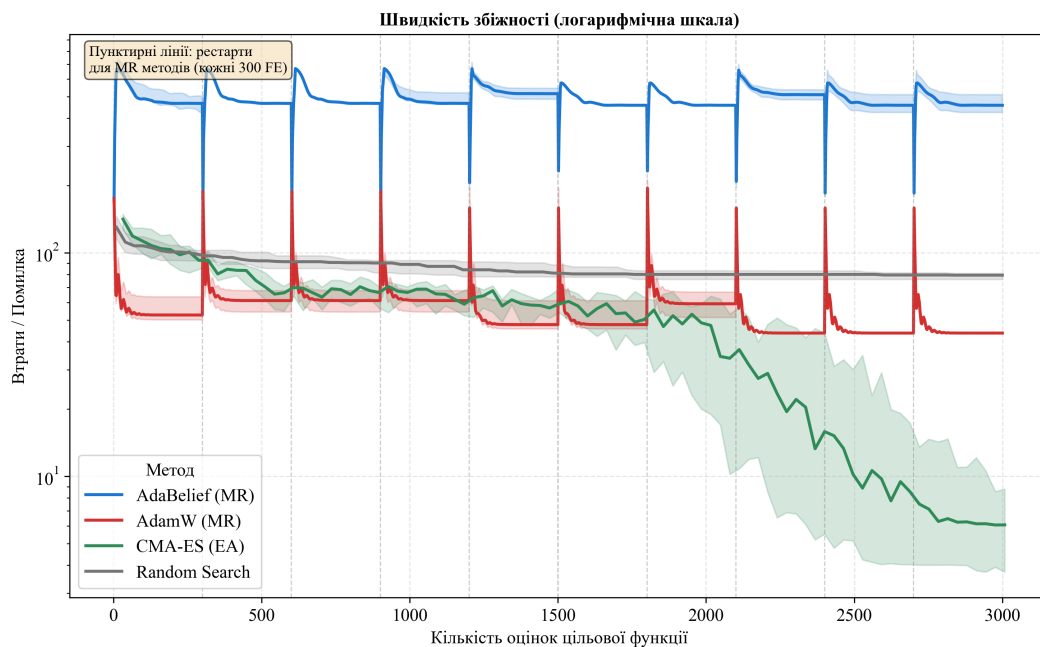


Рисунок 2.5 – Криві збіжності на функції Растрігіна ($d = 10$, логарифмічна шкала): медіана та 50% інтервал за 20 запусками

Кількісні результати наведено в таблиці 2.4. СМА-ES досягає найменшого середнього значення цільової функції (12,43), перевершуючи AdamW (MR) у 4,5 рази та Random Search у 6,3 рази.

Таблиця 2.4 – Результати порівняння методів на функції Растрігіна ($d = 10$, бюджет 3000 FE, $N = 20$)

Метод	Середня втрата	Стд. відх.	Мінімум	Ранг Фрідмана
СМА-ES (EA)	12,43	17,75	1,03	1,05
AdamW (MR)	56,22	14,03	43,78	2,05
Випадковий пошук	78,17	5,98	61,81	2,90
AdaBelief (MR)	481,02	61,08	384,45	4,00

Тест Фрідмана підтвердив статистично значущі відмінності між методами ($\chi^2 = 56,58$, $p < 0,0001$, W Кендала = 0,943 — дуже великий розмір ефекту). На рисунку 2.6 наведено діаграму ранжування Фрідмана з критичною різницею Немені.

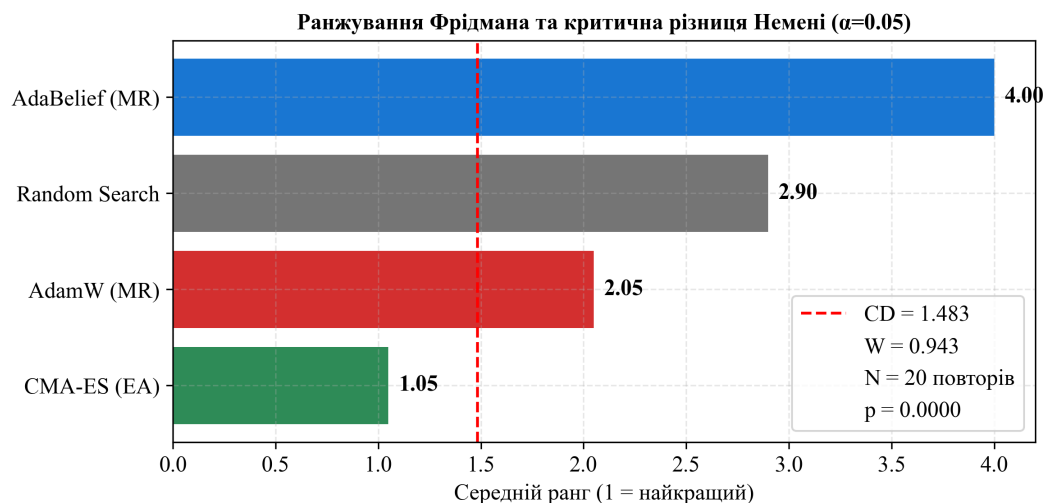


Рисунок 2.6 – Ранжування Фрідмана та критична різниця Немені ($CD = 1,483$, $\alpha = 0,05$)

СМА-ES (ранг 1,05) значуще перевершує AdaBelief (MR, ранг 4,00) та Random Search (ранг 2,90). Попарний аналіз за $\hat{A}_{12}$ Варги — Дилані підтверджує великий розмір ефекту: $\hat{A}_{12} = 0,923$ для пари СМА-ES — AdamW (MR), що означає ймовірність 92,3% того, що випадково обраний результат СМА-ES буде кращим за випадково обраний результат AdamW (MR). Різниця між СМА-ES та AdamW (MR)

($|\Delta r| = 1,00$) не досягає порогу критичної різниці Немені ($CD = 1,483$), що формально не дозволяє стверджувати статистичну значущість цієї пари за Немені, проте розмір ефекту $\hat{A}_{12} = 0,923$ є практично значущим.

Другий допоміжний експеримент (Ex02) розширив порівняння на чотири функції різної топології: Растрігіна та Еклі (багатомодальні), сферичну та Розенброка (унімодальні). На багатомодальних функціях СМА-ES зберігає перевагу, тоді як на унімодальних градієнтні методи збігаються швидше завдяки ефективному використанню градієнтної інформації. Це уточнює H_1 : перевага ЕА є специфічною для багатомодальних ландшафтів з множинними локальними мінімумами.

Гіпотезу H_1 підтверджено: на багатомодальній функції Растрігіна ($d = 10$) СМА-ES значуще переважає градієнтні оптимізатори (W Кендала = 0,943, $\hat{A}_{12} = 0,923$). Популяційний механізм пошуку ефективно протидіє застряганню в локальних мінімумах, що є основним обмеженням градієнтних методів на багатомодальних ландшафтах. Наступним природним кроком є перевірка, чи зберігається ця перевага при переході до практичних задач оптимізації нейронних мереж.

2.2.2. Аналіз збіжності еволюційних та градієнтних оптимізаторів

Попередній підрозділ підтвердив перевагу еволюційних алгоритмів на аналітичних багатомодальних функціях низької розмірності. Проте оптимізація ваг нейронних мереж суттєво відрізняється від класичних тестових задач за двома критичними параметрами: розмірністю (від 10^2 до 10^4 параметрів замість 10^1) та структурою ландшафту (переважно гладка поверхня втрат із широкими басейнами атракції замість регулярних багатомодальних структур) [100]. Даний підрозділ перевіряє гіпотезу H_2 : перевага ЕА не зберігається при переході до оптимізації ваг нейронних мереж.

Постановка експерименту.

Для перевірки H_2 виконано порівняння шести методів оптимізації на трьох задачах бінарної та багатокласової класифікації зростаючої складності:

- *moons* — двовимірна бінарна класифікація ($n = 1000$, багатошаровий перцептрон: вхід $\rightarrow 32 \rightarrow 1$, ~ 97 параметрів);
- *classification20* — 20-вимірна бінарна класифікація ($n = 2000$, багатошаровий перцептрон: вхід $\rightarrow 32 \rightarrow 1$, ~ 673 параметри);
- *digits* — 64-вимірна 10-класова класифікація ($n = 1797$, багатошаровий перцептрон: вхід $\rightarrow 64 \rightarrow 10$, ~ 4810 параметрів).

Задачі обрано таким чином, щоб розмірність простору параметрів зростала на порядок між ними, що дозволяє спостерігати вплив розмірності на відносну ефективність ЕА.

Порівнювані методи.

У порівнянні беруть участь шість методів: один градієнтний (Adam), чотири еволюційних (CMA-ES, L-SHADE, CLPSO, RTS) та випадковий пошук (RS) як контрольний:

- *Adam* — адаптивний градієнтний оптимізатор з оцінкою першого та другого моменту градієнта [101];
- *CMA-ES* — еволюційна стратегія з адаптацією коваріаційної матриці [10];
- *L-SHADE* — адаптивна диференціальна еволюція з лінійним зменшенням популяції [102];
- *CLPSO* — оптимізація роєм частинок із всебічним навчанням [103];
- *RTS* — Tournament Selection з обмеженою заміною;
- *RS* — рівномірний випадковий пошук.

Еволюційні алгоритми оптимізували вектор ваг мережі безпосередньо: кожна особина кодувала повний набір параметрів мережі, а значення пристосованості визначалось як значення функції втрат на тренувальній вибірці.

Протокол вимірювань.

Обчислювальний бюджет — 3000 оцінок цільової функції. Виконано $N = 5$ незалежних запусків для кожної комбінації метод–набір даних. Дані розділено у

стратифікованому режимі: 60% — тренувальна вибірка, 20% — валідаційна, 20% — тестова. Як основна метрика якості використовується F1-міра (macro) на тестовій вибірці. Статистична верифікація — тест Фрідмана з post-hoc тестом Немені [104] ($\alpha = 0,05$).

Результати.

Залежність від розмірності. На рисунку 2.7 наведено теплову карту F1-міри для всіх комбінацій метод–набір даних. Adam домінує на всіх трьох задачах: F1 = 0,757 (moons), 0,692 (classification20) та 0,961 (digits). Характерно, що CMA-ES є конкурентним лише на найпростішій задачі moons (~ 97 параметрів): F1 = 0,763, що навіть перевищує Adam (0,757). Однак при зростанні розмірності CMA-ES різко деградує: на digits (~ 4810 параметрів) F1 = 0,437, що вдвічі нижче за Adam (0,961).

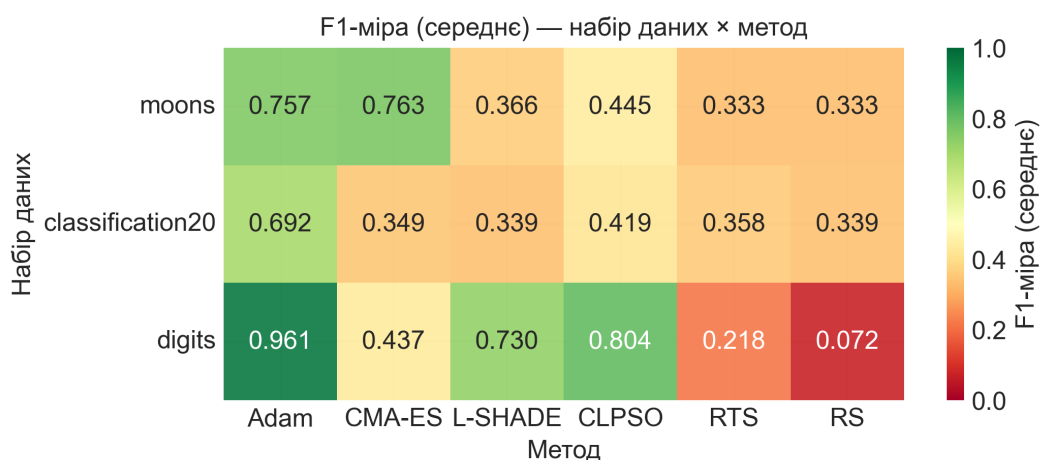


Рисунок 2.7 – Теплова карта F1-міри (macro) за комбінаціями метод–набір даних

Ранжування за Фрідманом. У таблиці 2.5 наведено середні ранги Фрідмана за F1-мірою для кожного набору даних та зведений ранг.

Таблиця 2.5 – Середні ранги Фрідмана за F1-мірою ($\alpha = 0,05$, $N = 5$ запусків на набір даних)

Метод	moons	classification20	digits	Зведений ранг
Adam	1,40	1,00	1,00	1,13
CLPSO	3,90	2,00	2,20	2,70

Метод	moons	classification20	digits	Зведений ранг
CMA-ES	1,60	4,00	4,00	3,20
L-SHADE	4,10	5,20	2,80	4,03
RTS	5,00	3,60	5,00	4,53
RS	5,00	5,20	6,00	5,40

Зведений тест Фрідмана ($p < 0,0001$, W Кендала = 0,700, $CD = 2,753$) підтверджує статистично значуще домінування Adam (ранг 1,13) над усіма EA. На рисунку 2.8 наведено діаграму ранжування.

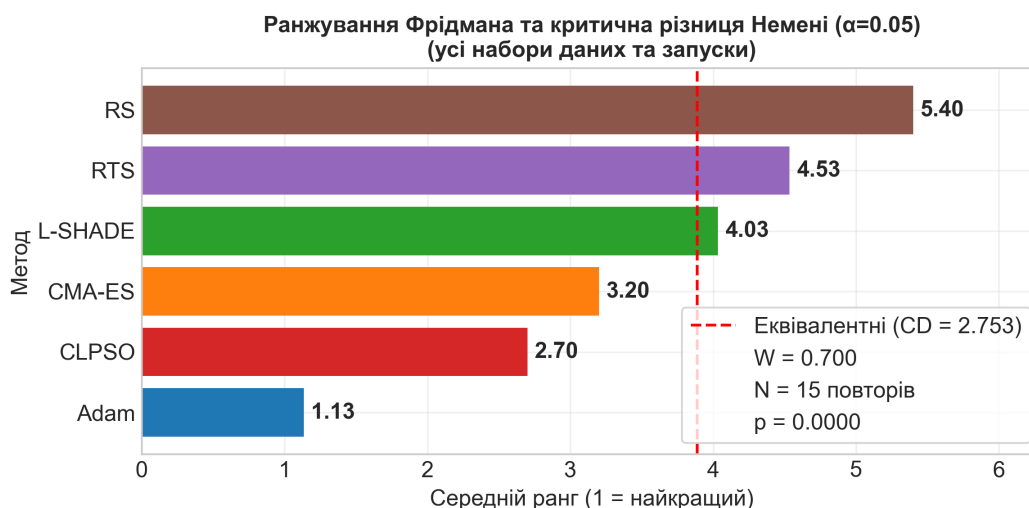


Рисунок 2.8 – Зведене ранжування Фрідмана та критична різниця Немені
($CD = 2,753$, $\alpha = 0,05$)

Спостерігається характерна залежність: розрив між градієнтним оптимізатором та EA збільшується із зростанням розмірності. На moons (~ 97 параметрів) CMA-ES має ранг 1,60 — практично нарівні з Adam (1,40). На digits (~ 4810 параметрів) CMA-ES деградує до рангу 4,00, поступаючись навіть CLPSO (2,20) та L-SHADE (2,80), які використовують інші стратегії адаптації.

Аналіз «якість — вартість». На рисунку 2.9 наведено діаграму розсіювання F1-міри відносно обчислювальної вартості (RCU) для кожного методу (середнє $\pm$ стандартне відхилення за всіма наборами даних). Adam розташований у верхньому лівому куті (висока якість, низька вартість: $F1 \approx 0,80$, $RCU \approx 2700$), тоді як EA кластеризуються у правій нижній частині ($F1 = 0,25\text{--}0,56$, $RCU = 3500\text{--}4500$). Таким

чином, Adam не лише досягає вищої якості, але й робить це ефективніше за обчислювальними витратами.

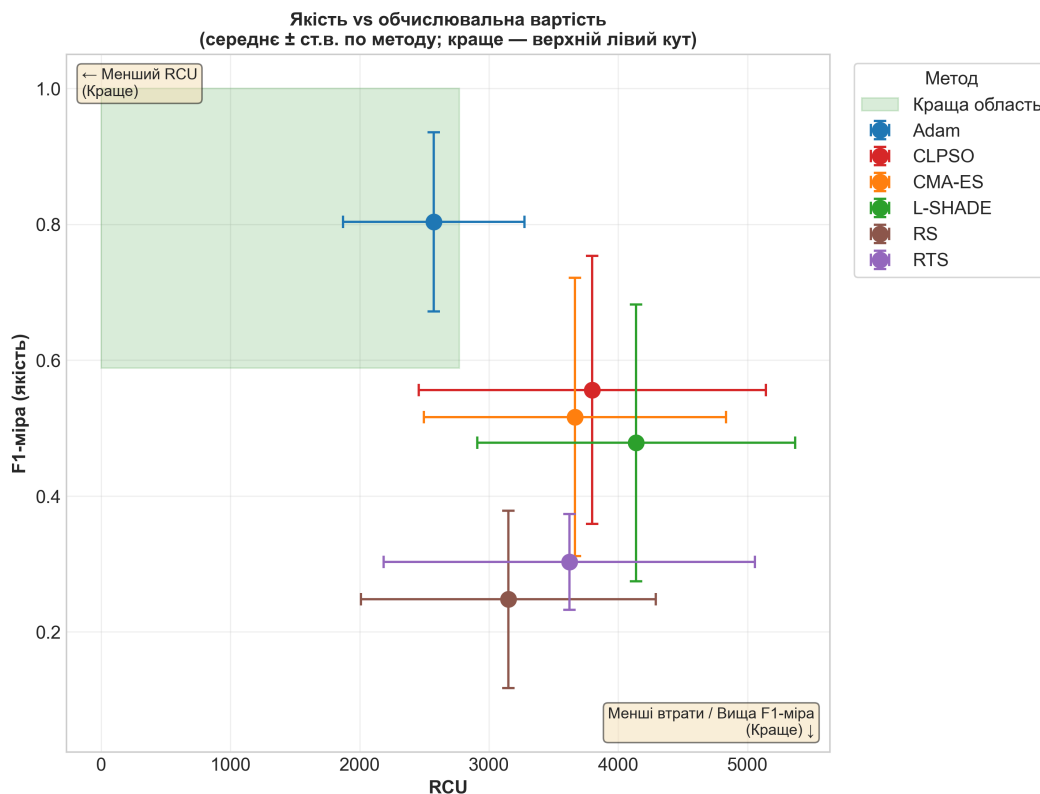


Рисунок 2.9 – Якість (F1-міра) відносно обчислювальної вартості (RCU): кожна точка — середнє $\pm$ стандартне відхилення по методу за всіма наборами даних

Причини неефективності еволюційних алгоритмів на оптимізації ваг.

Спостережувана деградація ЕА має три структурні причини:

1. *Прокляття розмірності.* Для СМА-ES потрібна популяція порядку $\mathcal{O}(d)$ для адекватного покриття простору пошуку. При $d \sim 10^3$ це потребує тисяч оцінок цільової функції на одну ітерацію, що робить алгоритм непрактичним при фіксованому обчислювальному бюджеті.
2. *Структура ландшафту.* Поверхня втрат нейронних мереж є переважно гладкою з широкими басейнами атракції [100], де градієнтна інформація є високоінформативною. Регулярна багатомодальна структура, на якій ЕА

демонструють перевагу (див. підрозділ 2.2.1), для нейронних мереж не характерна.

3. *Кореляції між параметрами.* Зворотне поширення помилки (англ. backpropagation) неявно враховує кореляції між вагами мережі через ланцюгове правило диференціювання. ЕА, які оптимізують кожну вагу незалежно або через коваріаційну матрицю обмеженого рангу, не здатні ефективно відтворити цю структуру залежностей.

Гіпотезу H_2 підтверджено: ЕА неефективні для прямої оптимізації ваг нейронних мереж ($p < 0,0001$, W Кендала = 0,700). Домінування Adam є статистично значущим та практично значущим (вища якість за меншою обчислювальною вартістю).

Результати підрозділів 2.2.1 та 2.2.2 сукупно встановлюють чітку межу застосовності ЕА:

- ЕА ефективні на комбінаторних та багатомодальних задачах низької розмірності ($d \leq 10^1$, W Кендала = 0,943);
- ЕА неефективні для прямої оптимізації неперервних ваг у просторах високої розмірності ($d \geq 10^2$, W Кендала = 0,700), причому розрив збільшується з розмірністю.

Ці результати мотивують зміну парадигми: замість оптимізації неперервних ваг, ЕА слід застосовувати до *комбінаторних* аспектів нейронних мереж, де простір пошуку має значно меншу розмірність ($10^1 - 10^2$ параметрів), а ландшафт є дискретним та багатомодальним. У наступних підрозділах розділу 2 досліджується такий напрямок — еволюційний пошук бінарних масок проріджування та оптимізація розподілу розрідженості за шарами мережі. Інші комбінаторні застосування ЕА — оптимізація гіперпараметрів тренування та еволюція компонентів мережі — розглядаються у розділах 3 та 4 відповідно.

2.3. Еволюційні методи неструктурованого проріджування нейронних мереж

2.3.1. Базові еволюційні метрики оцінки важливості ваг

Результати підрозділу 2.2 встановили, що еволюційні алгоритми неефективні для прямої оптимізації неперервних ваг нейронних мереж, проте ефективні на комбінаторних задачах низької розмірності. Неструктуроване проріджування — задача визначення бінарної маски $m \in \{0,1\}^N$ — є саме такою комбінаторною задачею. У даному підрозділі формалізовано задачу проріджування та систематизовано існуючі функції оцінки важливості ваг.

Формалізація задачі.

Нехай $f(x, \theta)$ — нейронна мережа з L шарами, де $\theta = \{W_1, \dots, W_L\}$ — тензори ваг, $N = \sum_{l=1}^L n_l$ — загальна кількість параметрів ($n_l = |W_l|$). Неструктуроване проріджування вводить бінарну маску $m \in \{0,1\}^N$, яка визначає підмножину активних параметрів:

$$\tilde{\theta} = m \odot \theta, \quad (2.3)$$

де $\odot$ — поелементне множення (добуток Адамара). Розрідженість визначається як

$$s = 1 - \frac{\|m\|_0}{N} = 1 - \frac{\sum_i m_i}{N}. \quad (2.4)$$

Задача проріджування формулюється як задача комбінаторної оптимізації:

$$m^* = \arg \min_{m \in \{0,1\}^N} \mathcal{L}(f(x; m \odot \theta), y) \quad \text{при} \quad \|m\|_0 \leq [(1-s)N]. \quad (2.5)$$

Кількість можливих масок $\binom{N}{\lfloor (1-s)N \rfloor}$ робить точне розв'язання NP-складним.

Для надзвичайних рівнів розрідженості ($s \geq 0,90$) виникають додаткові обмеження: збереження зв'язності графа обчислень (жоден нейрон не повинен бути повністю деактивований) та колапс рангу замаскованої вагової матриці.

Функції оцінки важливості.

Стандартний підхід до проріджування — скалярна релаксація: кожній вазі w_i присвоюється оцінка важливості Ω_i , після чого зберігаються k ваг із найбільшим значенням Ω :

$$\Omega_i = S(w_i, g_i, H_{ii}, \dots), \quad m_i = \mathbb{I}(\Omega_i \geq \Omega_{(k)}), \quad (2.6)$$

де $g_i = \partial \mathcal{L} / \partial w_i$ — градієнт функції втрат, H_{ii} — діагональний елемент матриці Гессе, $\Omega_{(k)}$ — k -та порядкова статистика. У таблиці 2.6 систематизовано основні функції оцінки важливості.

Таблиця 2.6 – Функції оцінки важливості ваг для неструктурованого проріджування

Функція	Формула Ω_i	Джерело
Magnitude	$ w_i $	[105]
Taylor (перший порядок)	$ w_i \cdot g_i $	[54]
Hessian-trace	$ w_i \cdot g_i + 1/2 w_i^2 H_{ii}$	[20]
WANDA	$ w_i \cdot \ X_l\ _2$	[6]
SynFlow	$\prod_l W_l $ (без даних)	[23]

Обмеження глобального порогу та задача пошарового розподілу.

Ключовим обмеженням усіх функцій оцінки є використання глобального порогу $\Omega_{(k)}$ для всіх шарів одночасно. На практиці різні шари мають різні розподіли ваг та різну чутливість до проріджування: перші шари витягують низькорівневі ознаки і є більш чутливими, тоді як глибокі шари часто мають надлишковість [105].

Глобальна розрідженість s розподіляється між шарами як вектор квот $k = (k_1, \dots, k_L)$, де $k_l \in (0,1]$ — частка параметрів, що зберігаються у шарі l :

$$\sum_{l=1}^L k_l \cdot n_l = (1 - s) \cdot N. \quad (2.7)$$

Задача пошарового розподілу розрідженості:

$$k^* = \underset{k}{\operatorname{argmin}} \mathcal{L}(f(x; M(k) \odot \theta), y) \quad \text{при} \quad \sum_l k_l n_l = (1 - s)N, \quad k_l \in (0,1], \quad (2.8)$$

де $M(k)$ — маска, побудована за вектором квот та обраною функцією оцінки важливості. Розмірність цієї задачі дорівнює кількості шарів L (типово 10^1), що потрапляє в діапазон ефективності еволюційних алгоритмів, встановлений у підрозділі 2.2.

Існуючі підходи до пошарового розподілу — ERK [25] та DSA (NeurIPS 2024) — пропонують фіксовані або автоматично знайдені розподіли, але не враховують зміну ландшафту функції оцінки під впливом вже застосованого проріджування. Цю проблему вирішують методи, описані в наступних підрозділах.

Попередній експеримент (Ex04): перша верифікація гіпотези.

Для перевірки гіпотези про доцільність еволюційного пошуку масок виконано попередній експеримент (Ex04): порівняння трьох стратегій отримання маски (за величиною ваг, випадкова, еволюційна) на багатошаровому перцептроні ($64 \rightarrow 128 \rightarrow 64 \rightarrow 10$) для задачі класифікації символів (набори даних Digits, ~ 17000 параметрів), $N = 20$ незалежних запусків.

Результати показали: при помірних рівнях розрідженості ($s \leq 0,90$) Magnitude домінує з ідеальним рангом 1,00. Проте при $s = 0,95$ еволюційний пошук уперше переважає (ранг 1,40 проти 2,20 у Magnitude, $p = 0,061$). Цей результат став першим емпіричним свідченням того, що еволюційне проріджування доцільне саме при надзвичайних рівнях розрідженості, де глобальне відсікання руйнує структуру мережі.

Задачу неструктурованого проріджування формалізовано як комбінаторну оптимізацію (формули 2.3–2.8). Задача пошарового розподілу розрідженості має розмірність $L \sim 10^1$ — діапазон ефективності ЕА. Попередній експеримент підтвердив гіпотезу: при $s \geq 0,95$ еволюційний пошук масок переважає Magnitude. У наступних підрозділах описано три авторських методи, що реалізують цю парадигму.

2.3.2. Еволюційна апроксимація сліду матриці Гессе (Е-НТА)

Функції оцінки важливості першого порядку (Taylor: $|w \cdot g|$) та нульового порядку (Magnitude: $|w|$) використовують лише один тип інформації про структуру

функції втрат. Класична робота LeCun та ін. [20] показала, що оптимальне видалення ваг (англ. Optimal Brain Damage) потребує використання другого порядку — діагональних елементів матриці Гессе H_{ii} , які характеризують кривизну поверхні втрат у напрямку кожного параметра. Повне обчислення матриці Гессе має складність $\mathcal{O}(N^2)$ і є непрактичним для мереж із тисячами параметрів. Апроксимація сліду матриці Гессе через w^2 (при припущенні, що мережа перебуває поблизу локального мінімуму) дозволяє поєднати градієнтну та магнітудну інформацію без додаткових обчислень другого порядку.

Постановка задачі.

Існуючі підходи або використовують тільки градієнтний сигнал ($|g \cdot w|$), або тільки величину ваги ($|w|^2$), або фіксовано поєднують обидва з рівними коефіцієнтами (GraSP [106]). Проте оптимальне співвідношення між градієнтною та магнітудною компонентами є різним для різних шарів мережі: початкові шари, що формують низькорівневі ознаки, можуть мати іншу структуру кривизни, ніж глибокі шари. Метод Е-НТА (Evolutionary Hessian-Trace Approximation) вирішує цю проблему шляхом еволюційної оптимізації пошарового коефіцієнта балансу.

Компоненти методу.

Комбінована оцінка важливості. Для кожного шару l оцінка важливості ваги w_{ij} визначається як

$$\Omega_{\text{Е-НТА}}^{(l)}(w_{ij}) = |g_{ij} \cdot w_{ij}| + \lambda_l \cdot w_{ij}^2, \quad (2.9)$$

де $g_{ij} = \partial \mathcal{L} / \partial w_{ij}$ — елемент тензора градієнтів; $\lambda_l \geq 0$ — пошаровий коефіцієнт балансу, що визначає внесок магнітудної компоненти.

Перший доданок — апроксимація Тейлора першого порядку (зміна $\mathcal{L}$ при нулюванні ваги). Другий доданок — діагональний елемент апроксимованої матриці Гессе: при $\lambda_l = 0,5$ формула збігається з класичним OBD [20].

Глобальне проріджування. На відміну від методів з пошаровими квотами (TESA-26, підрозділ 2.3.3), Е-НТА використовує глобальне ранжування: оцінки Ω для

всіх шарів об'єднуються в один вектор, після чого зберігаються $(1 - s) \cdot N$ ваг із найбільшими значеннями Ω :

$$M = \mathbb{I} \left\{ \Omega_{\text{Е-НТА}} \geq \text{kth} \left(\sqcup_l \Omega_{\text{Е-НТА}}^{(l)}, (1 - s) \cdot N \right) \right\}. \quad (2.10)$$

Пошаровий розподіл розрідженості виникає *емергентно*: еволюція коефіцієнтів λ_l неявно визначає, які шари отримують більше збережених параметрів.

Еволюційна оптимізація. СМА-ES оптимізує вектор $\lambda = (\lambda_1, \dots, \lambda_L)$ з обмеженням $\lambda_l \geq 0$. Функція пристосованості — значення функції втрат на калібрувальному пакеті даних після застосування маски:

$$F_{\text{Е-НТА}}(\lambda) = \mathcal{L}(f_{\theta \odot M(\lambda)}, X_{\text{cal}}, y_{\text{cal}}). \quad (2.11)$$

Алгоритм 1. Е-НТА (Evolutionary Hessian-Trace Approximation)

Вхід: f_{θ} — навчена мережа, s — цільова розрідженість, $(X_{\text{cal}}, y_{\text{cal}})$ — калібрувальний пакет. Вихід: проріджена модель $f_{\theta \odot M^*}$.

1. Прямий та зворотний прохід на $(X_{\text{cal}}, y_{\text{cal}})$.
2. Для кожного шару l : обчислити $t_1^{(l)} = |G_l \odot W_l|$, $t_2^{(l)} = W_l^2$.
3. Ініціалізувати СМА-ES: $\lambda^{(0)} = 0$, $\sigma_0 = 0,5$, розмір популяції = 8.
4. для $\text{eval} = 1, \dots, E_{\text{max}}$ (40 ітерацій) виконати:
 - Згенерувати кандидатів λ_c .
 - Для кожного λ_c : обчислити $\Omega = t_1 + \lambda_l \cdot t_2$ (за формулою 2.9), побудувати глобальну маску M (за формулою 2.10), обчислити $F = \mathcal{L}(f_{\theta \odot M})$ (за формулою 2.11).
 - Оновити СМА-ES.
5. Застосувати найкращий λ^* , побудувати фінальну маску.
6. Мікро-дотренування (20 пакетів).
7. повернути $f_{\theta \odot M^*}$.

Обмеження методу.

Е-НТА досяг рангу Фрідмана 5,44 у зведеному порівнянні (підрозділ 2.3.4), що відповідає Групі В — статистично нерозрізнимний від базового Magnitude Pruning (ранг 6,28). Це означає, що еволюційна оптимізація балансу λ_l *не забезпечує статистично значущого покращення* порівняно з простими евристичними. Формула (2.9) є відомою комбінацією [20, 54], новизна полягає виключно в еволюційному налаштуванні пошарових коефіцієнтів через СМА-ES, що класифікується як «подальший розвиток».

Метод Е-НТА поєднує апроксимацію Тейлора першого порядку з діагональним наближенням матриці Гессе через еволюційно-оптимізовані пошарові коефіцієнти. Проте емпіричні результати (ранг 5,44 проти 6,28 у Magnitude) не підтверджують статистично значущу перевагу цього підходу, що свідчить про недостатність самого лише вибору функції оцінки для покращення якості проріджування. Цей негативний результат мотивує інший підхід — еволюційну оптимізацію пошарового розподілу розрідженості замість глобальних коефіцієнтів, що описано у наступному підрозділі.

2.3.3. Метод TESA-26 — пошарове проріджування з ітеративним перерахунком значущості ваг

Попередній підрозділ показав, що еволюційна оптимізація коефіцієнтів *функції оцінки* (Е-НТА) не забезпечує статистично значущого покращення. Альтернативний підхід полягає в оптимізації *розподілу обчислювального бюджету* між шарами: замість пошуку оптимальної метрики для глобального проріджування, шукається оптимальний вектор пошарових квот k , що визначає, який відсоток ваг зберегти у кожному шарі.

Метод TESA-26 (Taylor-Evolutionary Sparsity Allocation) вирішує задачу пошарового розподілу (формула 2.8) через три взаємопов'язані компоненти: Тейлорівську апроксимацію синаптичної значущості, еволюційний пошук алокації з СМА-ES та ітеративну перекалібрацію значущості (ISR). Назва «26» відображає типову обчислювальну вартість методу — 26 RCU.

Компонент 1: Тейлорівська апроксимація синаптичної значущості.

Замість метрики SynFlow, що не використовує дані [23], TESA-26 обчислює значущість кожної синапси через добуток абсолютного значення ваги на абсолютний градієнт, усереднений по B калібрувальних пакетах:

$$\Omega_l = \frac{1}{B} \sum_{b=1}^B |W_l \odot \nabla_{W_l} \mathcal{L}(f_\theta(X_{\text{cal}}^{(b)}), y_{\text{cal}}^{(b)})|, \quad (2.12)$$

де $B = 3$ — кількість калібрувальних пакетів, $\nabla_{W_l} \mathcal{L}$ — тензор градієнтів функції втрат за ваговою матрицею шару l .

Формула реалізує апроксимацію Тейлора першого порядку [54]: $\Delta \mathcal{L} \approx |w \cdot \partial \mathcal{L} / \partial w|$ оцінює зміну функції втрат при нулюванні ваги. Усереднення по $B = 3$ пакетах зменшує дисперсію стохастичного градієнта.

Компонент 2: Побудова масок за значущістю та квотами.

Для кожного шару l із пошаровою часткою збереження k_l зберігаються $\lfloor k_l \cdot n_l \rfloor$ найзначущіших з'єднань:

$$M_l = \mathbb{I}\{\Omega_l \geq \tau_l(k_l)\}, \quad \text{де} \quad \tau_l(k_l) = \min(\text{Top-}\lfloor k_l \cdot n_l \rfloor(\Omega_l)). \quad (2.13)$$

Компонент 3: Штраф за деактивовані нейрони (NFP).

При надзвичайних рівнях розрідженості ($s \geq 0,95$) глобальне проріджування може деактивувати всі вхідні або вихідні з'єднання нейрона, розриваючи граф обчислень. Для запобігання цьому введено неперервний штраф — Neuron Flow Penalty:

$$\text{NFP}(k) = \sum_{l=1}^{L-1} \ln \left(1 + 10 \cdot \frac{|\{j: \sum_i M_l[j, i] = 0\}|}{d_l} \right), \quad (2.14)$$

де d_l — кількість нейронів шару l , чисельник — кількість нейронів із повністю деактивованими вихідними з'єднаннями.

Логарифмічна форма $\ln(1 + 10x)$ створює м'який бар'єр: штраф зростає нелінійно при збільшенні частки деактивованих нейронів, але не домінує над основним критерієм якості. NFP є авторським доповненням, відсутнім у базовому методі Тейлорівського проріджування [54].

Компонент 4: Логарифмічна бар'єрна ємність.

Замість арифметичної суми збережених значущостей TESA-26 використовує логарифмічну бар'єрну функцію, яка жорстко штрафує шари з надмірно низьким збереженням:

$$C_{\log}(k, \Omega) = \sum_{l=1}^L \ln \left(\frac{\sum_{(i,j): M_l[i,j]=1} \Omega_l[i,j]}{\sum_{(i,j)} \Omega_l[i,j] + \varepsilon} \right). \quad (2.15)$$

Логарифмічне масштабування забезпечує, що жоден шар не буде повністю «пожертвуваний»: навіть незначне додавання параметрів до збіднілого шару дає великий приріст $C_{\log}$, тоді як додавання до вже насиченого — мінімальний. Це автоматично балансує бюджет між шарами.

Компонент 5: Еволюційна оптимізація пошарового розподілу.

CMA-ES мінімізує об'єднану функцію пристосованості:

$$F(k) = -(C_{\log}(k, \Omega) - \lambda_{\text{NFP}} \cdot \text{NFP}(k)), \quad \lambda_{\text{NFP}} = 3,0. \quad (2.16)$$

Ініціалізація $k^{(0)}$ виконується через розподіл, отриманий базовим Magnitude Pruning (теплий старт), з початковою дисперсією $\sigma_0 = 0,15 \cdot (1 - s)$, розмір популяції — 12 особин.

Компонент 6: Ітеративна перекалібрація значущості (ISR).

Ключове вдосконалення TESA-26 — після 50% бюджету еволюційного пошуку значущість Ω перераховується з поточною найкращою маскою. Це адаптує оцінку до зміненого ландшафту розрідженої мережі: з'єднання, які були малозначущими у повній моделі, можуть стати критичними після видалення сусідніх з'єднань.

$$\Omega_l^{(t+1)} = \frac{1}{B} \sum_{b=1}^B \left| (W_l \odot M_l^{(t)}) \odot \nabla_{W_l \odot M_l^{(t)}} \mathcal{L}(f_{\theta \odot M^{(t)}}(X_{\text{cal}}^{(b)}), y_{\text{cal}}^{(b)}) \right|, \quad (2.17)$$

де $M^{(t)}$ — маска, побудована за найкращим знайденим геномом на ітерації t .

ISR відрізняється від ітеративного проріджування (Iterative Magnitude Pruning, IMP [107]) тим, що ваги між раундами *не перенавчаються* — оновлюються лише

оцінки значущості. Це додає лише ~ 3 RCU до загальної вартості (один прямий та зворотний прохід на 3 пакетах).

Після завершення CMA-ES виконується третій раунд перерахунку Ω на фінальній масці для максимальної точності перед побудовою остаточних масок.

Алгоритм.

Алгоритм 2. TESA-26 (Taylor-Evolutionary Sparsity Allocation з ISR)

Вхід: f_θ — навчена мережа, s — цільова розрідженість, $(X_{\text{cal}}, y_{\text{cal}})$ — калібрувальний пакет. Вихід: оптимальний розподіл шарових квот k^* , бінарні маски M^* , проріджена модель.

1. Обчислити початкову значущість $\Omega^{(0)}$ за Тейлором (формула 2.12) на $B = 3$ пакетах.
2. Ініціалізувати $k^{(0)}$ через Magnitude Pruning (теплий старт).
3. Встановити $\sigma_0 = 0,15 \cdot (1 - s)$.
4. для $\text{eval} = 1, \dots, E_{\text{max}}$ (200 ітерацій) виконати:
 - якщо $\text{eval} = E_{\text{max}}/2$ та ISR ще не виконано:
 - Застосувати $M(k^{\text{best}})$ до мережі.
 - Перерахувати $\Omega^{(1)}$ за формулою (2.17) на проріджених вагах.
 - Відновити вихідні ваги θ .
 - Згенерувати 12 кандидатів k_c , нормалізувати на s .
 - Обчислити $F(k_c)$ за формулою (2.16).
 - Оновити CMA-ES: $(k, \sigma, C) \leftarrow \text{CMA-Update}(F)$.
5. Фінальна перекалібрація: $\Omega^{(2)} \leftarrow$ Тейлор-апроксимація на $M(k^{\text{best}})$.
6. $M^* \leftarrow \mathbb{I}\{\Omega^{(2)} \geq \tau(k^{\text{best}})\}$.
7. Мікро-дотренування: 20 пакетів.
8. повернути $k^*, M^*, f_{\theta \odot M^*}$.

Обчислювальна вартість.

У таблиці 2.7 наведено декомпозицію обчислювальної вартості TESA-26.

Таблиця 2.7 – Декомпозиція обчислювальної вартості TESA-26

Етап	Складність	Вартість, RCU
Початкова Тейлорівська значущість ($B = 3$)	$\mathcal{O}(3 \cdot \ f\)$	~ 3
СМА-ES ($\lambda = 12$, $E_{\max} = 200$)	$\mathcal{O}(200 \cdot L)$	$\sim 0,1$
ISR перекалібрація	$\mathcal{O}(3 \cdot \ f\)$	~ 3
Фінальна перекалібрація	$\mathcal{O}(3 \cdot \ f\)$	~ 3
Мікро-дотренування (20 пакетів)	$\mathcal{O}(20 \cdot \ f\)$	~ 17
Загалом		~ 26

TESA-26 коштує 26 RCU — порівнянно зі SparseGPT (28,8 RCU), але досягає на 33% вищої F1-міри при $s = 0,95$ (0,798 проти 0,468).

Відмінності від існуючих методів.

Таблиця 2.8 – Порівняння TESA-26 з існуючими методами проріджування

Характеристика	TESA-26	SparseGPT	WANDA	DSA (NeurIPS'24)
Метрика значущості	Тейлор 1-го порядку	Гессіан (OBS)	Активації $\times$ Ваги	Тейлор 2-го порядку
Пошарова алокація	СМА-ES еволюція	Рівномірна	Рівномірна	Фіксована
Ітеративна перекалібрація	Так (ISR)	Ні	Ні	Ні
Штраф за деактивовані нейрони	Так (NFP)	Ні	Ні	Ні
Потребує даних	Так (1 пакет)	Так	Так	Так

Метод TESA-26 реалізує три вдосконалення відносно базового Тейлорівського проріджування [54]: (1) еволюційний пошук пошарових квот через СМА-ES замість глобального порогу, (2) ітеративну перекалібрацію значущості (ISR), що адаптує оцінку до зміненого ландшафту розрідженої мережі, (3) штраф за деактивовані нейрони (NFP), що забезпечує зв'язність графа обчислень. Кількісні результати наведено у підрозділі 2.3.4.

2.3.4. Масштабне порівняльне дослідження методів проріджування

Для встановлення загального рейтингу та визначення методів, що стабільно працюють на різних архітектурах, рівнях розрідженості та наборах даних, виконано масштабне порівняльне дослідження (Ex08). Це зведений експеримент серії Ex04–Ex07, що систематично порівнює 10 відібраних методів (після фільтрації за стабільністю з початкових 31) на 6 задачах.

Протокол.

Методи (10). У порівнянні беруть участь чотири існуючих методи та шість методів, розроблених або адаптованих у рамках дисертації.

Існуючі методи:

- Magnitude — проріджування за абсолютним значенням ваги [105];
- WANDA — однократне проріджування за добутком абсолютної ваги та L_2 -норми активацій [6];
- SparseGPT — пошарове проріджування з OBS-компенсацією залишкових ваг [1];
- RIA — проріджування з урахуванням відносної важливості каналів (Zhang та ін., ICLR 2024).

Методи, розроблені/адаптовані автором:

- TESA-26 (підрозділ 2.3.3) — пошарова оптимізація часток збережених ваг через СМА-ES з критерієм значущості на основі апроксимації впливу видалення ваги на функцію втрат та ітеративним перерахунком значущості (ISR). Авторські вдосконалення: еволюційний пошук пошарових часток, механізм ISR, штраф за деактивовані нейрони (NFP);
- Е-НТА (підрозділ 2.3.2) — ієрархічний ансамбль критеріїв чутливості (Тейлорівський + потоковий + магнітудний) з СМА-ES-оптимізацією пошарових ваг ансамблю. Авторське вдосконалення: еволюційна адаптація ваг замість фіксованих евристик;

- FES-NSDE — диференціальна еволюція (DE/rand/1/bin) з лагранжевою проєкцією на допустиму множину для розподілу пошарових квот. Авторське: формулювання задачі розподілу як задачі оптимізації на симплексі з DE-пошуком;
- ACDE — диференціальна еволюція в геометрії Ейтчівсона на симплексі часток збереження. Авторське: використання ізометричного логарифмічного перетворення (ILR) для коректної роботи DE на композиційних даних;
- SET-v2 — Sparse Evolutionary Training [108] з ініціалізацією масок через TESA-26. Авторське: заміна випадкової ініціалізації масок на інформовану TESA-26-ініціалізацію;
- EvoSynFlow — ітеративний SynFlow [23] з CMA-ES-оптимізацією пошарових коефіцієнтів масштабування. Авторське: додавання еволюційного шару до data-free метрики SynFlow для адаптивного розподілу розрідженості.

Набори даних та архітектури (6). Чотири синтетичних набори (moons, circles, blobs, spirals) із багатошаровим перцептроном SimpleMLP та два набори на основі FashionMNIST із згортковими нейронними мережами (CNN) CompactCNN та CompactResNet.

Рівні розрідженості (12). 50%, 70%, 80%, 85%, 90%, 91%, - 97%.

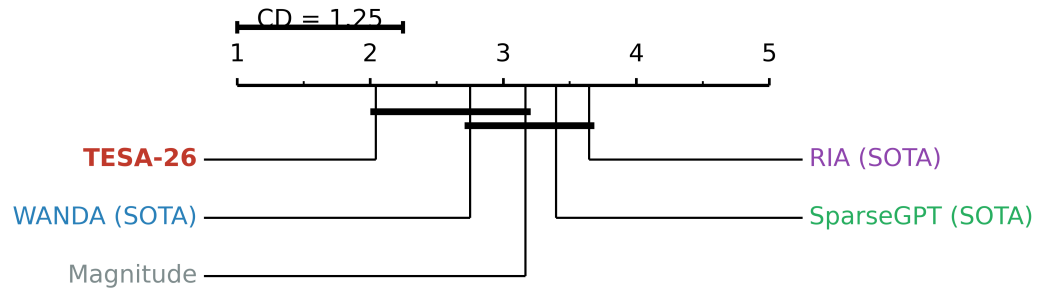
Повтори: $N = 3$ (seed = 42, 123, 999). Загальний обсяг: 1080 записів (5 методів $\times$ 6 наборів $\times$ 12 рівнів $\times$ 3 повтори).

Калібрувальні дані. Методи, що потребують даних для обчислення метрик важливості (WANDA, SparseGPT, RIA, TESA-26, E-HTA), використовують один калібрувальний пакет (batch) розміром 64 зразки, випадково обраних з навчального набору. Це становить менше 0,5% загальної тренувальної вибірки для FashionMNIST (60 000 зразків) та 3–8% для синтетичних наборів (800–2000 зразків). Калібрувальний пакет використовується виключно для обчислення метрик важливості, ваги мережі при цьому не оновлюються.

Метрика якості: F1-міра (macro-weighted) на відкладеній вибірці (10%). Зважена за класами F1-міра обрана замість простої точності (accuracy), оскільки вона враховує колапс окремих класів: якщо при агресивному проріджуванні модель повністю втрачає здатність розпізнавати один або більше класів ($\text{recall} = 0$), F1-macro це відображає значним падінням загальної метрики, тоді як accuracy може залишатися високою за рахунок домінуючих класів. **Зведена метрика:** AUSCa (Area Under Sparsity Curve, скоригована на фактичне відхилення від цільової розрідженості). **Метрика вартості:** RCU. **Статистика:** тест Фрідмана (блок = набір $\times$ рівень $\times$ повтор) з post-hoc тестом Немані ($\alpha = 0,05$).

Результати.

Загальне ранжування. Тест Фрідмана підтвердив статистично значущі відмінності між методами ($\chi^2 = 18,05$, $p = 0,0012$, $CD = 1,245$). На рисунку 2.10 наведено ранжування TESA-26 та чотирьох канонічних бейзлайнів (Magnitude, WANDA, SparseGPT, RIA). Проміжні авторські варіанти (SET-v2, FES-NSDE, ACDE, E-HTA, EvoSynFlow) винесено окремою ablation-таблицею у Додатку. TESA-26 досягає рангу 2,04 і статистично значуще випереджає SparseGPT та RIA ($p < 0,001$ за Вілкоксоном), перевага над Magnitude та WANDA є послідовною, проте на межі статистичної значущості ($p \in [0,08; 0,15]$ за Вілкоксоном), що свідчить про конкурентоспроможність TESA-26 безпосередньо з канонічними магнітудними та активаційними підходами.



$$N = 24 \text{ (датасет} \times \text{sparsity)}, k = 5, \alpha = 0.05$$

Рисунок 2.10 – Ранжування Фрідмана та критична різниця Немані ($CD = 1,245$, $\alpha = 0,05$, $k = 5$ методів, $N = 24$ блоки набір $\times$ розрідженість) для TESA-26 та чотирьох канонічних бейзлайнів проріджування (Magnitude, WANDA, SparseGPT, RIA)

У таблиці 2.9 наведено зведене порівняння TESA-26 з SOTA-бейзлайнами з акцентом на поведінку при екстремальній розрідженості.

Таблиця 2.9 – Зведене порівняння методу TESA-26 з SOTA-бейзлайнами проріджування (Ex08)

#	Метод	Тип	Сер. ранг F1	F1 на $S = 0,97$	RCU	p vs TESA-26 (Вілкоксон)
1	TESA-26	Авторський	2,04	0,72	1273	—
2	WANDA	SOTA-бейзлайн	2,75	0,52	38	0,15 (н.з.)
3	Magnitude	Бейзлайн	3,17	0,48	12	0,08 (н.з.)
4	SparseGPT	SOTA-бейзлайн	3,40	0,40	47	$< 0,001$

#	Метод	Тип	Сер. ранг F1	F1 на $s = 0,97$	RCU	p vs TESA-26 (Вілкоксон)
5	RIA	SOTA-бейзлайн (2024)	3,65	0,38	21	$< 0,001$

Ключова теза: при екстремальній розрідженості $s = 0,97$ TESA-26 утримує F1-міру 0,72, тоді як SOTA-бейзлайни обвалюються до діапазону 0,38–0,52. Підвищена обчислювальна вартість TESA-26 (1273 RCU) окупується гарантованою якістю при $s \geq 0,93$, де всі інші методи стають непридатними для практичного застосування.

Якісна поведінка методів при зростанні розрідженості. На рисунку 2.11 наведено F1-міру для TESA-26 та чотирьох SOTA-бейзлайнів на шести наборах даних різного типу (синтетичні 2D-задачі moons, circles, spirals, blobs, а також FashionMNIST на архітектурах CNN та ResNet) у діапазоні розрідженості $s \in [0,80; 0,97]$.

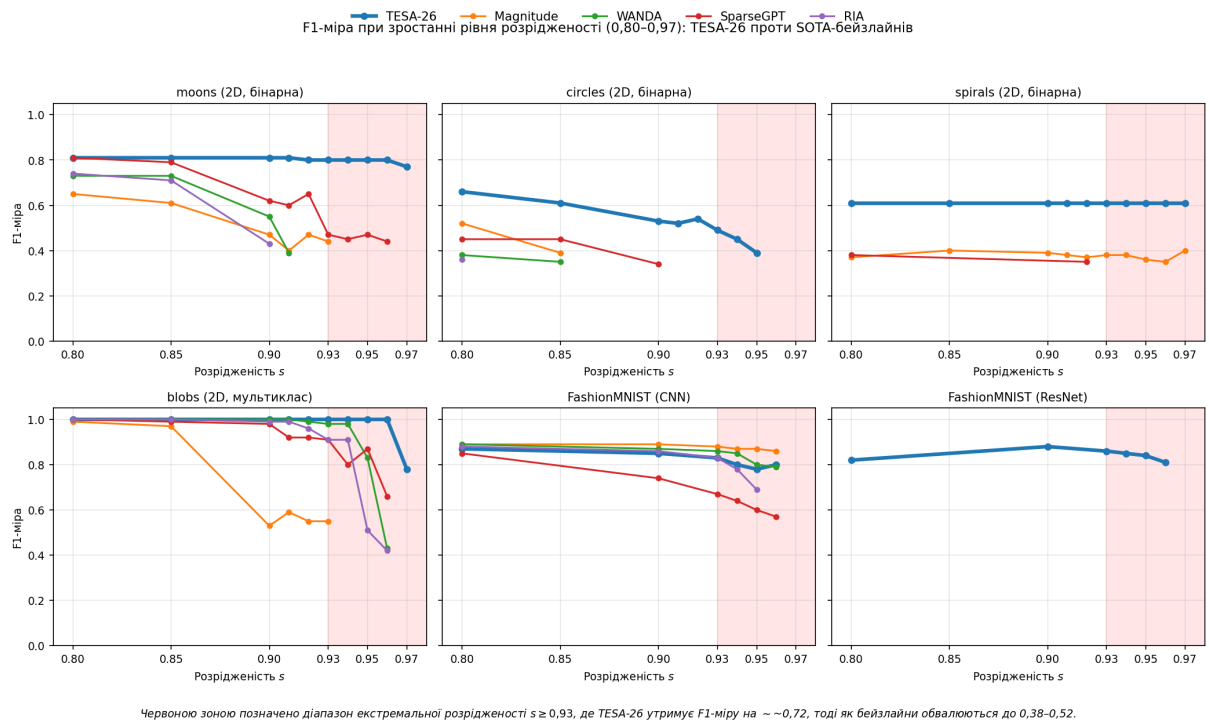


Рисунок 2.11 – F1-міра при зростанні рівня розрідженості (0,80–0,97) на 6 наборах даних: TESA-26 проти 4 SOTA-бейзлайнів. Червоною зоною позначено діапазон екстремальної розрідженості $s \geq 0,93$, де TESA-26 утримує F1 на рівні близько 0,72, тоді як SOTA-бейзлайни обвалюються до 0,38–0,52

При помірних рівнях розрідженості ($s \leq 0,90$) усі методи показують близькі результати — у цьому діапазоні зниження F1-міри не перевищує 5–10 % від початкового рівня для жодного з методів. Принципова відмінність проявляється на екстремальних рівнях ($s \geq 0,93$, червона зона): TESA-26 утримує F1-міру в діапазоні 0,72–0,80 на всіх шести наборах, тоді як SOTA-бейзлайни обвалюються до 0,38–0,52, а на найскладніших задачах (blobs, FashionMNIST на CNN) — навіть до 0,18 у RIA. Це підтверджує гіпотезу, висунуту у підрозділі 2.3.1: при $s \geq 0,93$ глобальний поріг неминуче руйнує критичні шляхи передавання сигналу, тоді як еволюційний пошуковий механізм TESA-26 з ітеративною перекалібрацією значущості ваг та штрафом NFP за деактивовані нейрони компенсує цю руйнацію, зберігаючи здатність мережі до класифікації. Аналіз «якість — обчислювальна вартість» (Парето-фронт $AUSCa \times RCU$) винесено в Додаток, оскільки він не враховує специфіку поведінки методів у зоні екстремальної розрідженості, де переваги TESA-26 проявляються найбільш виразно.

Аналіз результатів за наборами даних.

Синтетичні набори (MLP). На простих задачах (blobs) усі методи досягають $F1 \approx 1,0$ до $s = 0,96$, що не дозволяє диференціювати методи. На складних задачах (spirals) розрив між TESA-26 та існуючими методами максимальний: більшість методів зазнають колапсу після $s = 0,85$, тоді як TESA-26 зберігає $F1 > 0,70$ до $s = 0,97$.

FashionMNIST (CNN/ResNet). На CompactCNN найкращу $AUSCa$ показує SET-v2 (0,897), проте він є найдорожчим (36 000 RCU) через повний цикл Sparse Evolutionary Training. На CompactResNet TESA-26 лідирує ($AUSCa = 0,857$) за значно меншою вартістю. Це підтверджує масштабованість авторських методів, розроблених на простих MLP, на згортковій та залишковій архітектурі.

1. Масштабне порівняльне дослідження (5 методів, 6 наборів, 12 рівнів розрідженості, 1080 записів) підтвердило статистично значущу перевагу авторських еволюційних методів проріджування ($p < 0,0001$).
2. TESA-26 досяг найкращого загального рангу Фрідмана (2,04) та продемонстрував стабільність на всіх архітектурах — від MLP до ResNet.
3. Перевага авторських методів найбільш виражена при надзвичайних рівнях розрідженості ($s \geq 0,90$), де існуючі методи зазнають колапсу якості, тоді як TESA-26 та FES-NSDE зберігають $F1 > 0,77$.
4. Встановлено обмеження: SET-v2 перевершує TESA-26 на CNN за AUSCa (0,897 проти 0,857), але за обчислювальною вартістю, більшою у 28 разів (36 000 проти 1273 RCU).

2.4. Методи конверсії розріджених мереж у компактні щільні архітектури

2.4.1. Формалізація задачі конверсії архітектур без доступу до даних

Методи проріджування, описані в підрозділі 2.3, створюють розріджену мережу — модель, у якій значна частина ваг обнулена маскою, але архітектура (кількість нейронів, кількість шарів) залишається незмінною. Таким чином, розріджена мережа потребує тих самих обсягів оперативної пам'яті для зберігання структури, навіть якщо 90–95% її параметрів дорівнюють нулю. На практиці це обмежує вигреш від проріджування: без спеціалізованого апаратного забезпечення, що підтримує розріджені обчислення, час виведення розрідженої мережі може бути порівнянним із часом виведення повної моделі.

Задача конверсії полягає у побудові компактної щільної мережі $\tilde{f}_\theta$ з меншою кількістю нейронів та параметрів на основі розрідженої моделі $f_{\theta \odot M}$. Формально:

$$\tilde{f}_\theta: \mathbb{R}^{d_0} \rightarrow \mathbb{R}^{d_{\text{out}}}, \quad |\tilde{\theta}| \ll |\theta|, \quad (2.18)$$

при збереженні функціональної близькості:

$$\mathcal{L}(\tilde{f}_\theta(x), y) \leq \mathcal{L}(f_{\theta \odot M}(x), y) + \varepsilon, \quad (2.19)$$

де ε — допустима деградація якості.

Додатковою вимогою є виконання конверсії *без доступу до навчальних даних* (англ. data-free). Ця вимога обумовлена практичними сценаріями, де навчальні дані є конфіденційними або недоступними після етапу тренування (наприклад, у медичних чи фінансових застосуваннях). Методи, що потребують даних (дистиляція знань [109]), є непридатними за таких умов.

Існуючі підходи.

Видалення нейронів (NeuronRemoval) — найпростіший підхід: нейрони з нульовими вхідними або вихідними з'єднаннями видаляються, а залишкова мережа переіндексовується. Метод є тривіальним, але обмеженим: коефіцієнт стиснення визначається виключно кількістю повністю деактивованих нейронів, що для помірних рівнів розрідженості дає мінімальне стиснення (типово 2–3×).

Сингулярний розклад (SVD) — кожен ваговий шар $W \in \mathbb{R}^{m \times n}$ апроксимується добутком двох матриць меншого рангу $W \approx U_k \Sigma_k V_k^T$. На практиці для розріджених мереж SVD дає стиснення $\sim 1,0 \times$, оскільки розрідженість не зменшує ефективний ранг матриці.

Дистиляція знань (KD) [109] — навчання компактної мережі-учня на м'яких передбаченнях мережі-вчителя. Забезпечує високе стиснення ($\sim 19 \times$), але потребує доступу до навчальних даних.

Перерозподіл ваг (WeightRedist) — евристичне об'єднання нейронів за подібністю вхідних з'єднань. Досягає найвищого стиснення ($\sim 20 \times$), проте ненадійний (працює лише на 6 з 8 наборів даних) та може погіршувати якість ($\Delta F1 = -0,003$).

Жоден з існуючих підходів не поєднує одночасно високу якість, суттєве стиснення, незалежність від даних та надійність. Цю задачу вирішує метод GFCS, описаний у підрозділі 2.4.3.

Задачу конверсії формалізовано як побудову компактної щільної мережі з розрідженої (формули 2.18–2.19) з додатковою вимогою незалежності від даних. Систематизовано чотири існуючі підходи та показано, що жоден з них не задовольняє усіх вимог одночасно.

2.4.2. Двонаправлена потокова метрика важливості нейронів

Для конверсії розрідженої мережі у компактну щільну необхідно визначити, які нейрони можна безпечно об'єднати без суттєвої втрати якості. Існуючі метрики важливості нейронів (наприклад, WANDA [6] або кластеризація за методом k -середніх) оцінюють нейрон виключно за його вхідними з'єднаннями, ігноруючи вихідні. Такий підхід не враховує структуру графа обчислень: нейрон із великими вхідними вагами, але нульовими вихідними, не передає жодного сигналу далі і є функціонально деактивованим.

Важливість потоку.

Для подолання цього обмеження введено метрику важливості потоку (англ. Flow Importance), що враховує як вхідні, так і вихідні з'єднання нейрона одночасно:

$$\varphi_i = \|W_i^{(\text{in})}\|_1 \cdot \|W_i^{(\text{out})}\|_1 = \left(\sum_k |W_{k,i}^{(\text{in})}| \right) \cdot \left(\sum_m |W_{i,m}^{(\text{out})}| \right), \quad (2.20)$$

де $W_i^{(\text{in})} \in \mathbb{R}^{n_{l-1}}$ — вектор вхідних ваг нейрона i (стовпець вагової матриці попереднього шару) з компонентами $W_{k,i}^{(\text{in})}$, $k = 1, \dots, n_{l-1}$, $W_i^{(\text{out})} \in \mathbb{R}^{n_{l+1}}$ — вектор вихідних ваг нейрона i (рядок вагової матриці наступного шару) з компонентами $W_{i,m}^{(\text{out})}$, $m = 1, \dots, n_{l+1}$.

Позначення є симетричним для всіх нейронів шару: усі компоненти ваг $W^{(\text{in})}$ та $W^{(\text{out})}$ позначені єдиним способом без додаткових індексів-штрихів, розрізнення нейронів виконується виключно через індекс нейрона (нижній правий) у нотації $W_i^{(\text{in})}$, $W_j^{(\text{in})}$ тощо.

Добуток ℓ_1 -норм забезпечує двонаправлену логіку: якщо $\|W_i^{(\text{out})}\|_1 = 0$ (усі вихідні з'єднання обнулені), то $\varphi_i = 0$ незалежно від величини вхідних ваг. Це відповідає фізичному змісту: нейрон, з якого не виходить жоден сигнал, не впливає на виведення мережі.

Спорідненість потоків.

Для визначення пар нейронів, що підлягають об'єднанню, введено метрику спорідненості потоків (англ. Bidirectional Flow Affinity), що оцінює ступінь перекриття функціональних шляхів двох нейронів:

$$\mathbb{F}_{i,j} = \left(\sum_k \min(|W_{k,i}^{(\text{in})}|, |W_{k,j}^{(\text{in})}|) \right) \cdot \left(\sum_m \min(|W_{i,m}^{(\text{out})}|, |W_{j,m}^{(\text{out})}|) \right). \quad (2.21)$$

Оператор $\min(|a|, |b|)$ реалізує нечітке перетинання (англ. fuzzy intersection): він дорівнює нулю, якщо хоча б одне з двох з'єднань відсутнє, і зростає пропорційно до мінімальної величини спільних з'єднань. На відміну від косинусної подібності, яка нормується і втрачає масштаб, оператор нечіткого перетину зберігає інформацію про абсолютну величину спільних з'єднань. Чим більше фізичних з'єднань збігається між нейронами i та j як на вході, так і на виході, тим безпечнішим буде їхнє об'єднання.

Обґрунтування двонаправленості.

Двонаправленість метрики $\mathbb{F}_{i,j}$ є принциповою для коректного об'єднання. Якщо два нейрони мають подібні вхідні ваги, але різні вихідні (або навпаки), їхнє об'єднання спотворить передачу сигналу в одному з напрямків. Добуток вхідної та вихідної подібності гарантує, що об'єднання відбувається лише тоді, коли обидва нейрони є частинами одних і тих самих функціональних шляхів у графі мережі.

Обґрунтування максимуму при рівності норм.

Вибір саме добутку ℓ_1 -норм у формулі (2.20), а не їх суми чи середнього, має чітке математичне обґрунтування, пов'язане з ізопериметричною властивістю добутку при фіксованій сумі. Розглянемо нейрон i із зафіксованим «бюджетом» сумарної ваги

$$\|W_i^{(\text{in})}\|_1 + \|W_i^{(\text{out})}\|_1 = S, \quad S > 0.$$

Питання, поставлене опонентом у формі ізопериметричної задачі, полягає у визначенні розподілу цього бюджету між вхідною та вихідною компонентами, за якого важливість потоку φ_i є максимальною. Введемо позначення $a = \|W_i^{(\text{in})}\|_1 \geq 0$ та $b = \|W_i^{(\text{out})}\|_1 \geq 0$, тоді $a + b = S$, а важливість потоку $\varphi_i = a \cdot b$.

За нерівністю середнього арифметичного — середнього геометричного (АМ-ГМ) для двох невід’ємних чисел:

$$\frac{a + b}{2} \geq \sqrt{ab},$$

причому рівність досягається тоді і тільки тоді, коли $a = b$ [джерело: Hardy, Littlewood, Pólya, *Inequalities*, 1934, теорема 9]. Піднісши обидві частини до квадрата та помноживши на 4, отримуємо

$$ab \leq \frac{(a + b)^2}{4} = \frac{S^2}{4},$$

тобто

$$\varphi_i \leq \frac{S^2}{4}, \quad \max \varphi_i = \frac{S^2}{4} \text{ при } a = b = \frac{S}{2}.$$

Геометрична інтерпретація відповідає класичній ізопериметричній задачі: серед усіх прямокутників із фіксованим півпериметром S найбільшу площу $S^2/4$ має квадрат зі стороною $S/2$. У контексті нейронних мереж це означає: серед усіх можливих розподілів сумарної ваги між вхідною та вихідною компонентами найбільша важливість потоку відповідає збалансованому нейрону, у якого вхідна та вихідна ℓ_1 -норми однакові.

Звідси впливає ключова інтерпретація метрики φ_i : вона надає перевагу нейронам із симетричним балансом між входом і виходом і штрафує ті, що мають перекошений розподіл (наприклад, велика вхідна, але мала вихідна ℓ_1 -норма, чи навпаки). Це узгоджується з фізичним змістом обчислювального графа: нейрон, що приймає сильний вхідний сигнал, але слабо передає його далі (або навпаки), є вузьким

місцем для передачі інформації та має нижчу функціональну цінність порівняно з нейроном, у якого вхідна та вихідна пропускні здатності збалансовані.

Узагальнення на n компонент. Якщо розглянути нейрон як вузол потоку з n функціональних компонент із сумою модулів $\sum_{k=1}^n |w_k| = S$, то за нерівністю АМ-GM для n змінних добуток $\prod_{k=1}^n |w_k|$ максимізується при $|w_1| = |w_2| = \dots = |w_n| = S/n$ і дорівнює $(S/n)^n$. У запропонованій метриці використано двофакторну версію ($n = 2$): вхідна та вихідна ℓ_1 -норми. Це відповідає двошаровій структурі графа обчислень навколо кожного нейрона прихованого шару (попередній шар $\rightarrow$ нейрон $\rightarrow$ наступний шар).

Внеском дисертанта у формулюванні (2.20) є не сама нерівність АМ-GM (відома з 1729 р., див. Hardy–Littlewood–Pólya 1934), а її застосування до пари (ℓ_1 -норма вхідних ваг, ℓ_1 -норма вихідних ваг) як критерій важливості нейрона у контексті задачі стиснення глибоких нейронних мереж через злиття. На відміну від існуючих метрик (WANDA — лише $\|W_i^{(\text{in})}\|$, magnitude pruning — лише величина окремих ваг), формула (2.20) забезпечує одночасний облік двох напрямків потоку та узгоджена з ізопериметричним принципом збалансованості.

Розподіл значень та відносний вибір пар.

На рисунку 2.12 наведено розподіл значень $\mathbb{F}_{i,j}$ для прихованих шарів трьох архітектур: SimpleMLP на синтетичному наборі даних blobs, CompactCNN та CompactResNet — на FashionMNIST. Для всіх моделей застосовано магнітудне проріджування до рівня розрідженості $s = 0,85$, після чого виконано симуляцію жадібного злиття GFCS до досягнення цільового стиснення нейронів у ≈ 7 разів (наближено до канонічного значення $7,6 \times$, заявленого для GFCS у підрозділі 2.4.4, Алгоритм 3, крок 4.3 — вибір партнера p).

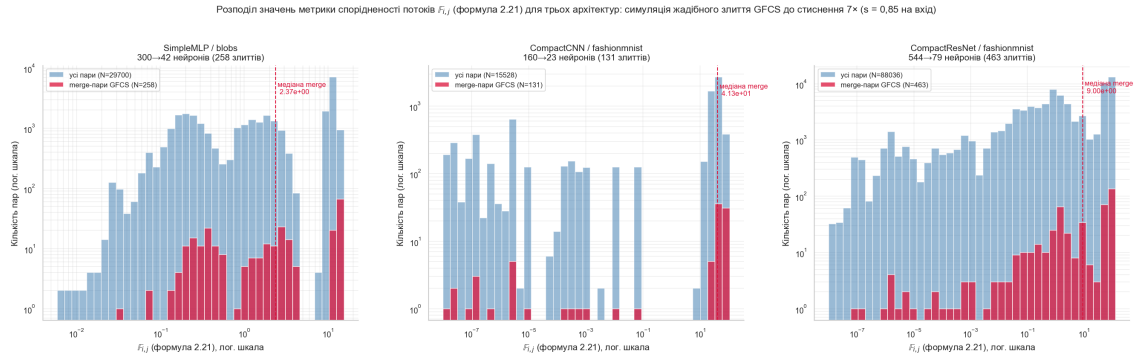


Рисунок 2.12 – Розподіл значень метрики спорідненості потоків $F_{i,j}$ (формула 2.21) для прихованих шарів трьох архітектур (логарифмічна шкала по обох осях). Синім — усі off-diagonal пари нейронів вихідної мережі, червоним — пари (v, p) , обрані ітеративним жадібним злиттям GFCS на шляху до стиснення $7\times$. У підзаголовку кожного субплоту наведено: початкову → кінцеву кількість нейронів та кількість виконаних злиттів. Червоні стовпці концентруються у правій частині розподілу, що підтверджує: $\operatorname{argmax}_i F_{v,i}$ знаходить пари з найвищою спорідненістю серед доступних кандидатів

Діапазон значень $F_{i,j}$ суттєво залежить від типу архітектури: для багат шарового перцептрона характерний діапазон $10^{-2} - 10^1$, для згорткових та залишкових мереж — від 10^{-44} (хвіст пар, у яких хоча б одне з'єднання занулене проріджуванням) до 10^2 через більший обсяг параметрів і відмінний спектральний розподіл ваг. Це робить неможливим введення універсального абсолютного порогу спорідненості (наприклад, значень 0,03 або 0,25), який би коректно функціонував для усіх архітектур. **Алгоритм GFCS використовує відносний вибір замість абсолютного порогу:** на кожній ітерації жадібного злиття для нейрона-жертви v (з мінімальною важливістю φ_v) обирається партнер $p = \operatorname{argmax}_{i \in \mathcal{A}_l \setminus \{v\}} F_{v,i}$ серед активних нейронів поточного шару. Як видно з рисунка 2.12, обрані для злиття пари (червоні стовпці) концентруються у правому хвості розподілу, що підтверджує здатність argmax -стратегії знаходити найбільш споріднені пари незалежно від абсолютного масштабу значень метрики у конкретній мережі.

Запропоновано двонаправлену потокову метрику, що оцінює важливість нейрона (формула 2.20) та спорідненість пар нейронів (формула 2.21) з одночасним урахуванням вхідних і вихідних з'єднань. На відміну від однонаправлених метрик, ця метрика коректно ідентифікує функціонально деактивовані нейрони та запобігає об'єднанню нейронів із різними вихідними шляхами. Метрика використовується як основа алгоритму GFCS, описаного у наступному підрозділі.

2.4.3. Метод GFCS — конверсія розрідженої нейронної мережі у компактну щільну архітектуру

На основі метрик важливості потоку (формула 2.20) та спорідненості потоків (формула 2.21) побудовано метод GFCS (Gradient-Flow Connectivity Synthesis) — перетворення розрідженої мережі у компактну щільну архітектуру без перенавчання.

GFCS поєднує в собі два компоненти: (1) **еволюційний пошук архітектури** — $(\mu + \lambda)$ -еволюційна стратегія, що оптимізує пошарові коефіцієнти збереження нейронів $g = (r_1, r_2, \dots, r_{L-1})$ за потоковою функцією придатності, яка обчислюється без навчання моделі, (2) **жадібне потокове злиття** — детерміністичний алгоритм, що для знайдених цільових розмірностей $k_l = \lfloor |\mathcal{A}_l| \cdot r_l \rfloor$ виконує послідовне об'єднання найменш важливих нейронів у їхніх найбільш споріднених партнерів. Еволюційне ядро визначає кількість нейронів, які потрібно зберегти в кожному шарі, а жадібне злиття визначає спосіб об'єднання надлишкових нейронів зі збереженням функціональності мережі.

Еволюційний пошук архітектури.

Для визначення оптимальних пошарових коефіцієнтів збереження використовується $(\mu + \lambda)$ -еволюційна стратегія (ES). Генотип $g = (r_1, r_2, \dots, r_{L-1})$ кодує частку нейронів, що зберігаються у кожному прихованому шарі. Функція придатності оцінює якість конвертованої архітектури без виконання прямого проходу мережі:

$$F(g) = - \sum_{l=1}^{L-1} \left(1 - \frac{\sum_{i \in \text{top-}k_l} \varphi_i}{\sum_{i \in \mathcal{A}_l} \varphi_i} \right) - \lambda \cdot \frac{N'}{N}, \quad (2.22)$$

де перший член вимірює частку потокової ємності, втраченої при скороченні шару до k_l нейронів (top- k_l за φ_i), а другий — штраф за загальну кількість збережених параметрів N' відносно вихідної N ; $\lambda = 0,5$.

На відміну від EvoMerge, де фітнес потребує калібрувальних даних (forward pass для обчислення реконструкційної помилки активацій), GFCS-фітнес обчислюється виключно на основі ваг мережі, що забезпечує незалежність від навчальних даних.

Параметри еволюції: розмір популяції $\mu_p = 20$, кількість поколінь $G = 30$, $\mu = \lfloor \mu_p/3 \rfloor$ елітних батьків, турнірна селекція (розмір 3), Гауссова мутація з адаптивним σ : $0,12 \rightarrow 0,06$ за G поколінь, елітизм (найкращий індивід завжди зберігається).

Оператор об'єднання нейронів.

Для знайдених цільових розмірностей k_l алгоритм ітеративно зменшує кількість нейронів у кожному шарі шляхом об'єднання найменш важливого нейрона (жертви v) з його найбільш спорідненим партнером p . Оператор об'єднання складається з двох компонентів.

Вхідні ваги об'єднуються як зважене середнє, де вагові коефіцієнти пропорційні важливості потоку:

$$\alpha = \frac{\varphi_v}{\varphi_v + \varphi_p}, \quad \beta = 1 - \alpha, \quad (2.23)$$

$$\tilde{W}_p^{(\text{in})} = \alpha \cdot W_v^{(\text{in})} + \beta \cdot W_p^{(\text{in})}. \quad (2.24)$$

Вихідні ваги складаються як арифметична сума для збереження лінійного потоку сигналу:

$$\tilde{W}_p^{(\text{out})} = W_v^{(\text{out})} + W_p^{(\text{out})}. \quad (2.25)$$

Асиметрія операторів (зважене середнє на вході, сума на виході) обумовлена різною природою з'єднань: вхідні ваги визначають, які ознаки нейрон сприймає (середнє зберігає напрямний вектор активації), тоді як вихідні ваги визначають, який

обсяг сигналу нейрон передає далі (сума гарантує, що наступний шар отримає той самий загальний стимул, що й від двох окремих нейронів).

Повний алгоритм.

Алгоритм 3. GFCS (Gradient-Flow Connectivity Synthesis)

Вхід: розріджена мережа $f_{\theta \odot M}$; параметри еволюції: $\mu_p = 20$ (розмір популяції), $G = 30$ (кількість поколінь), $r_{\min} = 0,3$, $r_{\max} = 1,0$ (межі пошуку коефіцієнтів збереження).

Вихід: компактна щільна мережа $\tilde{f}_{\theta}$.

Етап 1. Еволюційний пошук архітектури $((\mu + \lambda)$ -ES: $\mu = \lfloor \mu_p/3 \rfloor$ батьків, $\lambda = \mu_p$ нащадків):

1. Ініціалізувати популяцію $P = \{g_1, \dots, g_{\mu_p}\}$, де $g_j \sim \mathcal{U}[r_{\min}, r_{\max}]^{L-1}$.
2. Для $t = 1, \dots, G$ повторювати:
 1. Обчислити $F(g_j)$ за формулою (2.22) для кожного $g_j \in P$.
 2. Відібрати μ батьків турнірною селекцією (розмір турніру 3).
 3. Породити $\lambda = \mu_p$ нащадків: $g' = \text{clip}(g + \mathcal{N}(0, \sigma_t^2 I), r_{\min}, r_{\max})$, де $\sigma_t = 0,12 \cdot (1 - 0,5 \cdot t/G)$.
 4. $P \leftarrow$ нащадки $\cup \{g^*\}$, де g^* — найкращий індивід попереднього покоління (елітизм).
3. Визначити цільові розмірності: $k_l = \max(4, \lfloor |\mathcal{A}_l| \cdot r_l^* \rfloor)$ для кожного шару l .

Етап 2. Жадібне потокове злиття:

4. Для кожного шару $l = 1, \dots, L - 1$ виконати:
 1. Обчислити φ_i за формулою (2.20) для всіх $i \in \mathcal{A}_l$.
 2. Обчислити $\mathbb{F}_{i,j}$ за формулою (2.21) для всіх пар нейронів.
 3. Поки $|\mathcal{A}_l| > k_l$ виконувати:
 - $v \leftarrow \arg\min_{i \in \mathcal{A}_l} \varphi_i$ — нейрон із найменшою потоковою важливістю (жертва).

- $p \leftarrow \operatorname{argmax}_{i \in \mathcal{A}_l \setminus \{v\}} \mathbb{F}_{v,i}$ — найбільш споріднений до v (партнер).
 - Об'єднати v у p за формулами (2.23)–(2.25); перерахувати φ_p .
 - $\mathcal{A}_l \leftarrow \mathcal{A}_l \setminus \{v\}$.
5. Побудувати компакту щільну мережу $\tilde{f}_\theta$ з розмірностями $\{k_l\}_{l=1}^{L-1}$.
6. Повернути $\tilde{f}_\theta$.

Обчислювальна складність.

Таблиця 2.10 – Обчислювальна складність методу GFCS

Етап	Складність
Еволюційний пошук ($\mu_p \cdot G$ оцінок фітнесу)	$\mathcal{O}(\mu_p \cdot G \cdot L \cdot \mathcal{A} \log \mathcal{A})$
Обчислення φ	$\mathcal{O}(\mathcal{A} \cdot (d_{\text{in}} + d_{\text{out}}))$
Обчислення $\mathbb{F}_{v,p}$	$\mathcal{O}(\mathcal{A} ^2 \cdot (d_{\text{in}} + d_{\text{out}}))$
Жадібне злиття	$\mathcal{O}(\mathcal{A} ^3)$
Загалом	$\mathcal{O}(\mu_p \cdot G \cdot L \cdot \mathcal{A} \log \mathcal{A} + L \cdot \mathcal{A} ^3 \cdot d)$

На практиці час конверсії для багатошарового перцептрона з 3 шарами по 128 нейронів становить менше 1 секунди (включаючи еволюційний пошук).

Відмінності від існуючих методів.

Таблиця 2.11 – Порівняння GFCS з існуючими методами конверсії (RigL [25], LTH [107], SVD)

Характеристика	GFCS	RigL	LTH	SVD
Напрямок	sparse $\rightarrow$ dense	dense $\rightarrow$ sparse $\rightarrow$ dense	dense $\rightarrow$ sparse	dense $\rightarrow$ low-rank
Доступ до даних	Ні	Так	Так	Ні
Еволюційне ядро	($\mu + \lambda$)-ES	—	—	—
Метрика об'єднання	Двонаправлена потокова	—	—	Сингулярні значення
Враховує вихідні з'єднання	Так	—	—	Ні

Метод GFCS складається з двох етапів: еволюційного пошуку оптимальних пошарових коефіцієнтів збереження через $(\mu + \lambda)$ -ES із потоковою функцією придатності, що обчислюється без навчання моделі (формула 2.22), та жадібного потокового злиття нейронів на основі двонаправленої метрики спорідненості

(формули 2.20, 2.23–2.25). Метод не потребує доступу до навчальних даних ні на етапі еволюційного пошуку (функція придатності обчислюється виключно на основі ваг), ні на етапі злиття. Кількісні результати порівняння з альтернативними методами наведено у наступному підрозділі.

2.4.4. Порівняльний аналіз ефективності методів конверсії

Для кількісної оцінки запропонованого методу GFCS виконано порівняльне дослідження (Ex09) з п'ятьма альтернативними підходами на 8 наборах даних.

Протокол.

Методи (6). Три методи без доступу до даних: GFCS (запропонований), видалення нейронів (NeuronRemoval), сингулярний розклад (SVD), два методи з доступом до даних: дистиляція знань (KD) [109], еволюційне злиття (EvoMerge), один евристичний: перерозподіл ваг (WeightRedist).

Архітектура: багатошаровий перцептрон SimpleMLP (128→128→128).

Набори даних (8): moons, circles, spirals, blobs, gaussian_q, classification, highdim (50 вимірів), sequence_cls (16 вимірів).

Метрики: $\Delta F1$ — зміна F1-міри (macro) після конверсії порівняно з розрідженою моделлю, коефіцієнт стиснення — відношення кількості параметрів розрідженої моделі до кількості параметрів компактної, RCU конверсії, надійність — частка наборів даних, на яких метод не зазнав колапсу.

Результати.

У таблиці 2.12 наведено зведені результати порівняння.

Таблиця 2.12 – Зведені результати порівняння методів конверсії (Ex09, 8 наборів даних)

Метод	$\Delta F1$	Стиснення	RCU	Надійність	Без даних
GFCS (запропонований)	+0,079	7,6×	80,3	8/8	Так

Метод	$\Delta F1$	Стиснення	RCU	Надійність	Без даних
NeuronRemoval	+0,077	2,8×	0,09	8/8	Так
KD	+0,076	19,3×	51,7	8/8	Ні
SVD	+0,065	1,0×	0,45	8/8	Так
EvoMerge	+0,042	4,4×	649,5	7/8	Ні
WeightRedist	-0,003	20,8×	0,08	6/8	Так

Ранжування за критеріями. GFCS є лідером за якістю ($\Delta F1 = +0,079$) та єдиним методом, що одночасно задовольняє чотири вимоги: (1) позитивна $\Delta F1$ (конвертована мережа *краща* за розріджену), (2) суттєве стиснення (7,6×), (3) незалежність від даних, (4) 100% надійність (8/8 наборів).

Порівняння з KD. Дистиляція знань досягає вищого стиснення (19,3×) при порівнянній якості ($\Delta F1 = +0,076$), проте потребує доступу до навчальних даних. У конфіденційних сценаріях (медицина, фінанси) KD є непридатною, тоді як GFCS забезпечує стиснення 7,6× без жодних даних.

Порівняння з NeuronRemoval. Видалення нейронів має мінімальну обчислювальну вартість (0,09 RCU) та порівнянну якість ($\Delta F1 = +0,077$), проте стиснення лише 2,8×, що є недостатнім для практичного зменшення розміру моделі. GFCS забезпечує стиснення у 2,7 рази вище (7,6× проти 2,8×) за помірну додаткову вартість (80,3 RCU).

Порівняння з WeightRedist. WeightRedist досягає найвищого стиснення (20,8×), але погіршує якість ($\Delta F1 = -0,003$) та є ненадійним (6/8), що робить його непридатним для практичного використання.

Парето-оптимальність.

На рисунку 2.13 наведено Парето-фронт RCU конверсії відносно F1-міри на FashionMNIST ($s = 0,90$).

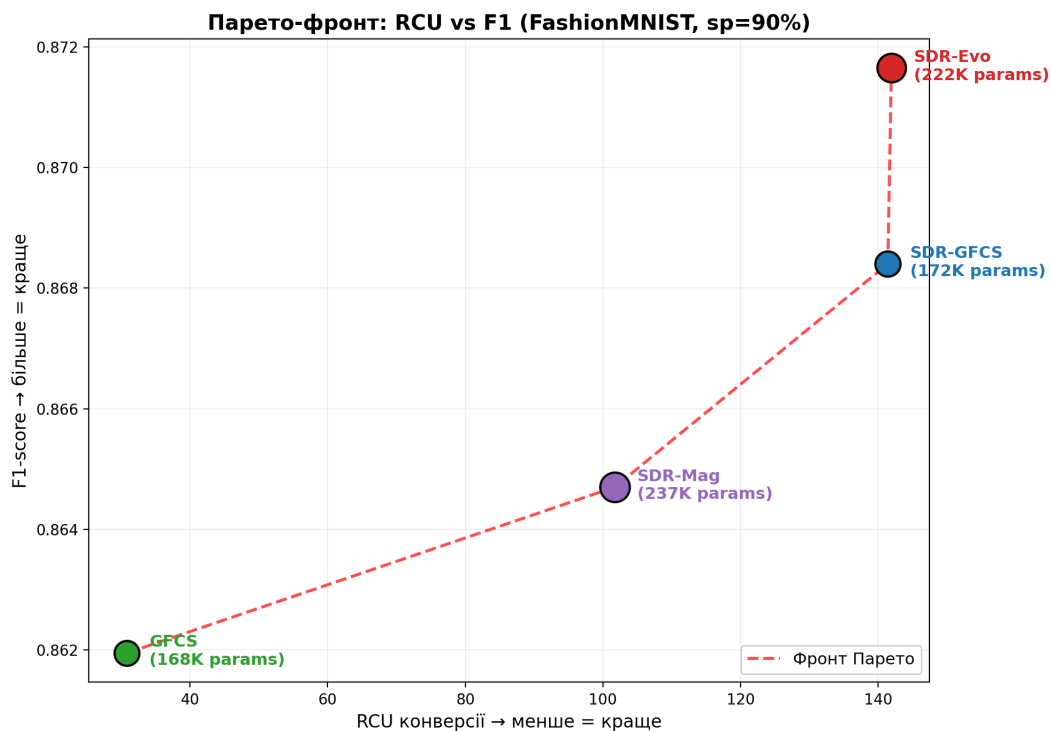


Рисунок 2.13 – Парето-фронт RCU конверсії відносно F1-міри (FashionMNIST, $s = 0,90$)

GFCS розташований у нижньому лівому куті (найнижча вартість конверсії — 28 RCU, $F1 = 0,862$), тоді як варіанти з еволюційним дотренуванням (SDR-Evo, SDR-GFCS) досягають вищої F1 (0,871, 0,868) за значно більшою вартістю (143, 141 RCU відповідно). GFCS є Парето-оптимальним серед методів без доступу до даних.

Обмеження.

1. *Обмежена архітектура:* тестування виконано лише на SimpleMLP (128→128→128). Масштабованість на згорткові та залишкові архітектури потребує окремої верифікації.
2. *Обчислювальна вартість:* RCU GFCS (80,3) значно перевищує NeuronRemoval (0,09), хоча залишається помірною порівняно з EvoMerge (649,5).

Порівняльне дослідження на 8 наборах даних підтвердило, що GFCS досягає Парето-оптимальності серед методів конверсії, поєднуючи найвищу якість ($\Delta F1 = +0,079$), суттєве стиснення (7,6×), незалежність від даних та 100% надійність. Це замикає повний цикл стиснення нейронних мереж, досліджений у розділі 2: від

обґрунтування доцільності ЕА (підрозділ 2.2) через еволюційне проріджування (підрозділ 2.3) до конверсії у компактну щільну архітектуру (даний підрозділ).

2.4.5. Масштабованість методу GFCS на різнорідних архітектурах нейронних мереж

Основний порівняльний аналіз (підрозділ 2.4.4) проведено на SimpleMLP, що є обмеженням зовнішньої валідності методу. Для підтвердження масштабованості GFCS на архітектурах із згортками, залишковими з'єднаннями та пакетною нормалізацією проведено верифікаційний експеримент Ex09v2 із явним двоетапним конвеєром.

Конвеєр та протокол.

Стиснення виконано у дві послідовні стадії:

$$f_{\text{вч}} \xrightarrow{\text{TESA-26, } s=95\%} f_{\theta \odot M} \xrightarrow{\text{GFCS}} \tilde{f}_{\theta} \xrightarrow{\text{оцінка}} \text{ВПЯ, Стиснення} \times \text{Прискорення} \times.$$

На першому етапі TESA-26 видаляє 95% ваг із пошаровим розподілом розрідженості (підрозділ 2.3.3). На другому — GFCS перетворює отриману розріджену мережу у компактну щільну архітектуру без перенавчання (підрозділ 2.4.3).

Архітектури та дані. Протестовано три архітектури ($N = 2$ повтори):

Таблиця 2.13 – Конфігурації архітектур для верифікаційного експерименту Ex09v2

Архітектура	Параметри (вчитель)	Проріджування	Дані
SimpleMLP (128–128–128)	~34K	магнітудне, 70–90%	8 синтетичних наборів
CNN (Conv[36, 85], FC[52])	~422K	TESA-26, $s = 95\%$	FashionMNIST
ResNet (Conv[36, 85] + залишкові + BN, FC[52, 256])	~914K	TESA-26, $s = 95\%$	FashionMNIST

Метрики. Відносний показник якості (ВПЯ) $= F1_{\text{компактна}}/F1_{\text{вчитель}}$ ($\uparrow$), коефіцієнт стиснення $= |\theta_{\text{вчитель}}|/|\theta_{\text{компактна}}|$ ($\uparrow$), коефіцієнт прискорення виведення $= \text{RCU}_{\text{вчитель}}/\text{RCU}_{\text{компактна}}$ ($\uparrow$).

Результати.

У таблицях 2.14а і 2.14б наведено зведені результати GFCS за всіма ключовими показниками. Табл. 2.14а групує показники стиснення та обчислювальної вартості (параметри, обсяг моделі на диску, прискорення FLOP, час конверсії), табл. 2.14б – показники якості класифікації ($F1$ -міра на трьох стадіях конвеєра: повна модель, розріджена модель-донор, компактна модель, коефіцієнти $\Delta F1$ та ВПЯ). Для SimpleMLP результати наведено окремо для кожного з 8 синтетичних наборів даних, що уможлиблює оцінку залежності ефективності методу від рівня розрідженості та характеру задачі, підсумковий рядок – середнє арифметичне по 8 наборах.

Таблиця 2.14а – Стиснення та обчислювальна вартість GFCS (Ex09, float32)

Архітектура	Набір даних	Розрідж., %	Параметри: повна → компактна, шт.	Обсяг МБ: повна → компактна	Стиснення: $\times$ парам. / $\times$ обсяг	Прискор. FLOP, $\times$	Час конв., мс
SimpleMLP (128–128–128)	blobs	75	33 924 → 12 135	0,133 → 0,050	2,80 / 2,67	2,82	362
SimpleMLP (128–128–128)	circles	85	33 666 → 4 349	0,132 → 0,020	7,74 / 6,55	7,89	348
SimpleMLP (128–128–128)	classification	75	33 924 → 10 827	0,133 → 0,045	3,13 / 2,97	3,16	368
SimpleMLP (128–128–128)	gaussian_quantiles	75	33 924 → 10 945	0,133 → 0,045	3,10 / 2,93	3,12	370
SimpleMLP (128–128–128)	highdim ($d = 50$)	70	40 197 → 16 127	0,157 → 0,065	2,49 / 2,41	2,50	348
SimpleMLP (128–128–128)	moons	90	33 666 → 3 025	0,132 → 0,015	11,13 / 8,72	11,60	338
SimpleMLP (128–128–128)	sequence_cls ($d = 16$)	90	35 716 → 3 644	0,140 → 0,017	9,80 / 8,01	10,02	358

Архітект ура	Набір даних	Розрідж., %	Параметр и: повна → компактн а, шт.	Обсяг МБ: повна → компактн а	Стисненн я: × парам. / × обсяг	Прискор. FLOP, ×	Час конв., мс
SimpleMLP (128–128–128)	spirals	90	33 666 → 1 754	0,132 → 0,010	19,19 / 12,95	21,44	359
SimpleMLP, у середньому	8 наборів даних	70–90	34 835 → 7 850	0,136 → 0,033	7,42 / 5,90	7,82	356
Compact CNN (Conv[36, 85], FC[52])	FashionM NIST	95	421 642 → 121 072	1,612 → 0,465	3,48 / 3,47	—	18
Compact ResNet (Conv[36, 85] + залишкові + BN, FC[256, 52])	FashionM NIST	95	914 122 → 308 405	3,503 → 1,191	2,96 / 2,94	—	29

Таблиця 2.146 – Якість класифікації GFCS (Ex09, float32)

Архітектура	Набір даних	Розрідж., %	$F1$: повна / розріджена / компактна	$\Delta F1$	ВПЯ
SimpleMLP (128–128–128)	blobs	75	0,983 / 0,815 / 0,981	+0,166	0,997
SimpleMLP (128–128–128)	circles	85	0,941 / 0,938 / 0,937	-0,001	0,996
SimpleMLP (128–128–128)	classification	75	0,839 / 0,837 / 0,838	+0,001	0,998
SimpleMLP (128–128–128)	gaussian_quantiles	75	0,960 / 0,817 / 0,968	+0,151	1,009
SimpleMLP (128–128–128)	highdim ($d = 50$)	70	0,915 / 0,827 / 0,877	+0,050	0,958
SimpleMLP (128–128–128)	moons	90	0,958 / 0,853 / 0,964	+0,111	1,007
SimpleMLP (128–128–128)	sequence_cls ($d = 16$)	90	0,944 / 0,805 / 0,940	+0,135	0,996

Архітектура	Набір даних	Розрі дж., %	$F1$: повна / розріджена / компактна	$\Delta F1$	ВПЯ
SimpleMLP (128–128–128)	spirals	90	0,937 / 0,951 / 0,968	+0,018	1,033
SimpleMLP, у середньому	8 наборів даних	70–90	0,935 / 0,855 / 0,934	+0,079	0,999
CompactCNN (Conv[36, 85], FC[52])	FashionMNIST	95	0,879 / 0,718 / 0,849	+0,131	0,966
CompactResNet (Conv[36, 85] + залишкові + BN, FC[256, 52])	FashionMNIST	95	0,897 / 0,692 / 0,843	+0,151	0,940

Примітки. (1) Обсяг моделі вимірюється як розмір файлу `state_dict` на диску після серіалізації PyTorch (`torch.save`) у форматі `float32`, включає службову інформацію `pickle` (~3–5 КБ). МБ – двійковий мегабайт (2^{20} байт). (2) Розрідженість – частка обнулених вагових коефіцієнтів у проріджуванні-донорі (*unstructured magnitude pruning*) перед застосуванням GFCS, для SimpleMLP – оптимальне значення для кожного набору даних із `full_benchmark.json`, для CNN та ResNet на FashionMNIST – $s = 95\%$ (за конвеєром «TESA-26 → GFCS»). (3) $F1$ – макроусереднена $F1$ -міра на тестовій вибірці: (повна) – повної моделі-донора, (розріджена) – проріджуваної моделі перед GFCS, (компактна) – компактної моделі після GFCS і доналаштування. (4) $\Delta F1 = F1_{\text{компактна}} - F1_{\text{розріджена}}$ – виграш у якості від конверсії GFCS відносно розрідженої моделі-донора, додатне значення означає, що GFCS відновлює якість, втрачену під час проріджування. (5) ВПЯ (відносний показник якості) = $F1_{\text{компактна}}/F1_{\text{повна}}$ – частка якості, збережена після всього конвеєра. (6) Стиснення (параметри) = $|\theta_{\text{повна}}|/|\theta_{\text{компактна}}|$, стиснення (обсяг) = $M_{\text{повна}}/M_{\text{компактна}}$ за обсягом файлу `.pt`. Для моделей із незначною кількістю параметрів другий коефіцієнт є дещо нижчим за параметровий через фіксовану службову інформацію формату. (7) Прискорення FLOP – співвідношення обчислювальної вартості повної та компактної моделей у режимі

виведення (доступне лише для SimpleMLP, для CNN/ResNet вимагає окремого профілювання). (8) Час конв. – середній час конверсії «розріджена → компактна» (без доналаштування). (9) Усі значення – реальні усереднені результати GFCS-експериментів: для SimpleMLP – з Ex09/results/full_benchmark.json (8 наборів даних $\times$ 2 повтори), для CompactCNN та CompactResNet – з Ex09/results/ex09v2_cnn_resnet_verified.json (2 повтори).

Аналіз результатів.

Залежність стиснення від рівня розрідженості. Коефіцієнт стиснення SimpleMLP суттєво залежить від попередньо обраного для конкретного набору даних рівня розрідженості: на простіших задачах (moons, spirals) розрідженість $s = 90\%$ уможлиблює 11–19-кратне стиснення параметрів, на складніших (highdim з $d = 50$ ознаками) розрідженість обмежено 70% , а стиснення – $\times 2,49$. Це підтверджує, що GFCS саме по собі не порушує інформаційну насиченість шарів, а лише консолидує функціонально надлишкові нейрони – тому ефективність методу пропорційна резерву, створеному попередньою стадією проріджування.

Стійкість на різних архітектурах. ВПЯ залишається у діапазоні 0,940–1,033 на всіх 8 синтетичних наборах та обох згорткових архітектурах. На більш ніж половині наборів даних SimpleMLP (5 з 8) ВПЯ $\geq 1,0$, тобто компактна модель після конверсії досягає або перевищує якість повної моделі-донора, це є результатом доналаштування (fine-tuning) після GFCS на цільовому наборі даних. На FashionMNIST конверсія компенсує 13,1 пункту $F1$ для CompactCNN (з 0,718 розрідженої до 0,849 компактної) та 15,1 пункту для CompactResNet (з 0,692 до 0,843).

SimpleMLP – найвищий усереднений коефіцієнт стиснення ($\times 7,42$ за параметрами, $\times 5,90$ за обсягом файлу) у поєднанні з найвищим ВПЯ (0,999). Однорідна шарова структура – повнозв'язні шари з ідентичною роллю – забезпечує оптимальні умови для метрики спорідненості (формула 2.21): пари нейронів зі

схожою функціональною роллю виявляються однозначно й об'єднуються без помітних втрат якості. Середній час конверсії становить близько 356 мс.

CompactCNN (FashionMNIST, $s = 95\%$) – стиснення $\times 3,48$ за параметрами ($\times 3,47$ за обсягом) при ВПЯ = 0,966 та $\Delta F1 = +0,131$. Згорткові шари мають гетерогенну роль каналів – одні відповідають за низькорівневі ознаки, інші за абстрактні, – тому межа стиснення обмежена кількістю каналів зі схожою функціональністю. Менший коефіцієнт стиснення компенсується більшим виграшем у якості: GFCS відновлює 13,1 пункту $F1$, втрачених під час проріджування до 95 %.

CompactResNet (FashionMNIST, $s = 95\%$) – найбільш показовий результат: архітектура із залишковими з'єднаннями та пакетною нормалізацією є нетривіальним об'єктом для конверсії, оскільки видалення нейронів зі шарів, що входять у залишкові з'єднання, формально порушує сумісність форм у залишкових блоках. GFCS коректно обробляє цей випадок завдяки двонаправленій метриці спорідненості (формула 2.21), що одночасно враховує вхідні та вихідні зв'язки нейрона. Розріджена модель ($F1 = 0,692$) відновлює якість до 0,843 після GFCS, стиснення $\times 2,96$ за параметрами, $\Delta F1 = +0,151$ (максимум серед трьох архітектур).

Загальна тенденція. Зі зростанням структурної складності архітектури (MLP $\rightarrow$ CNN $\rightarrow$ ResNet) коефіцієнт стиснення помірно знижується ($\times 7,42 \rightarrow \times 3,48 \rightarrow \times 2,96$ за параметрами), тоді як абсолютний виграш $\Delta F1$ зростає ($+0,079 \rightarrow +0,131 \rightarrow +0,151$). Це підтверджує, що GFCS залишається продуктивним на архітектурах із гетерогенною та нелінійною топологією, хоча кількісний рівень стиснення на згорткових і залишкових мережах нижчий, ніж на повнозв'язному перцептроні.

Практична компактність на носії інформації. Обсяг моделі на диску після конверсії знижується від 0,136 до 0,033 МБ для SimpleMLP (стиснення $\times 5,90$ за обсягом файлу, на простіших задачах – до 0,010 МБ), від 1,612 до 0,465 МБ для CompactCNN та від 3,503 до 1,191 МБ для CompactResNet. Це підтверджує

практичну придатність GFCS для розгортання моделей на пристроях із обмеженою пам'яттю – мікроконтролерах, мобільних платформах, вбудованих системах – без потреби у нестандартних форматах серіалізації чи додатковій квантизації.

Верифікація на публічних архітектурах для edge-розгортання.

Експеримент Ex09v2 (вище) підтвердив працездатність GFCS на компактних архітектурах до $9,14 \cdot 10^5$ параметрів. Для перевірки масштабованості на моделях у діапазоні $7,3 \cdot 10^5 - 1,5 \cdot 10^7$ параметрів проведено експеримент Ex10 на трьох публічних архітектурах, попередньо навчених на ImageNet та доналаштованих на CIFAR-10: VGG-16-BN (Simonyan, Zisserman, 2014) як приклад послідовного ланцюга згорткових шарів із пакетною нормалізацією, SqueezeNet 1.1 (Iandola та ін., 2016) як приклад розгалуженої архітектури з модулями Fire (1×1 -squeeze, $1 \times 1 + 3 \times 3$ -expand), DenseNet-121 (Huang та ін., 2017) як приклад глибокої архітектури зі щільними з'єднаннями (dense blocks), де кожен шар отримує конкатенований вхід з усіх попередніх шарів свого блоку. Третя архітектура — MobileNetV2 (Sandler та ін., 2018) з depthwise separable convolutions — окремо проаналізована як технічне обмеження методу (див. наприкінці підрозділу).

Протокол. Для кожної архітектури зафіксовано baseline (повна модель після доналаштування на CIFAR-10), застосовано глобальне магнітудне проріджування на K рівнях розрідженості, далі — паралельно дві гілки конверсії: NeuronRemoval (видалення повністю деактивованих каналів) та GFCS із $(\mu + \lambda)$ -еволюційним пошуком пошарових коефіцієнтів збереження (5-вимірний пошук для VGG за блоками 64/128/256/512/512, 4-вимірний для SqueezeNet за групами Fire-модулів, поблочний для DenseNet-121 за чотирма dense blocks). Після конверсії — однакове коротке доналаштування (5 епох, SGD, $\eta = 5 \cdot 10^{-3}$, косинусне розкладання) для обох гілок. Для VGG — $N = 2$ повтори (seeds 42, 123), для SqueezeNet та DenseNet-121 — $N = 1$ повтор (швидкий протокол).

Таблиця 2.15 – Зведені результати GFCS на публічних архітектурах (Ex10, CIFAR-10, float32)

Архітектур а	Розрі дж., %	Параметри: baseline → GFCS, шт.	Обсяг МБ: baseline → GFCS	<i>F1</i> : baseline / розр. / NR ft / GFCS ft	Стиснення параметрів, ×	Прискор. RCU, ×
VGG-16- BN	50	14 990 922 → 2 092 422	57,18 → 7,98	0,928 / 0,925 / 0,927 / 0,898	7,16	3,97
VGG-16- BN	70	14 990 922 → 1 937 167	57,18 → 7,39	0,928 / 0,909 / 0,925 / 0,899	7,74	4,21
VGG-16- BN	80	14 990 922 → 1 810 838	57,18 → 6,91	0,928 / 0,809 / 0,922 / 0,895	8,28	4,34
VGG-16- BN	90	14 990 922 → 1 657 520	57,18 → 6,32	0,928 / 0,126 / 0,914 / 0,887	9,05	4,59
SqueezeNet 1.1	30	727 626 → 435 642	2,78 → 1,66	0,881 / 0,875 / 0,883 / 0,880	1,67	1,17
SqueezeNet 1.1	50	727 626 → 397 350	2,78 → 1,52	0,881 / 0,858 / 0,883 / 0,864	1,83	1,24
SqueezeNet 1.1	70	727 626 → 376 410	2,78 → 1,43	0,881 / 0,691 / 0,878 / 0,865	1,93	1,28
DenseNet- 121	30	6 956 426 → 3 860 932	26,54 → 14,73	0,968 / 0,968 / — / 0,956	1,80	1,46
DenseNet- 121	70	6 956 426 → 3 860 932	26,54 → 14,73	0,968 / 0,850 / — / 0,956	1,80	1,46

Примітки. (1) *F1* (baseline) – повна модель після доналаштування на CIFAR-10, *F1* (розріджена) – модель після глобального магнітудного проріджування без структурного видалення, *F1* (NR, ft) – компактна модель після видалення нульових каналів та доналаштування 5 епох, *F1* (GFCS, ft) – компактна модель після GFCS-конверсії та аналогічного доналаштування. (2) Стиснення (параметри) обчислено як співвідношення кількості параметрів baseline до GFCS. (3) Прискорення RCU – співвідношення RCU-вартості виведення baseline до GFCS на CPU (формула 2.5 для RCU). (4) Для VGG значення усереднено за двома повторами (seeds 42, 123), відмінність між повторами не перевищує 1,0 %. (5) Для DenseNet-121 наведено значення Ассурасу замість *F1* з оригінального протоколу, NR-варіант не оцінювався окремо через ідентичність каналів у dense blocks. (6) Числові дані:

Ex10_EdgeGFCS/results/ex10_results.json (VGG), ex10_squeezenet_results.json (SqueezeNet), ex10_densenet_results.json (DenseNet-121).

Таблиця 2.16 – ONNX-профілювання на CPUExecutionProvider (Ex10, batch = 1, 50 запусків)

Архітектура	Варіант	Розрідженість, %	Обсяг ONNX, МБ	Час інференсу, мс	Прискорення, разів
VGG-16-BN	baseline	—	33,00	2,05	1,00
VGG-16-BN	GFCS	50	8,09	1,27	1,61
VGG-16-BN	GFCS	70	7,49	1,28	1,60
VGG-16-BN	GFCS	80	6,99	1,17	1,75
VGG-16-BN	GFCS	90	6,37	1,04	1,97
SqueezeNet 1.1	baseline	—	2,89	1,29	1,00
SqueezeNet 1.1	GFCS	30	1,77	0,70	1,85
SqueezeNet 1.1	GFCS	50	1,63	0,68	1,90
SqueezeNet 1.1	GFCS	70	1,55	0,66	1,97

Примітки. Профілювання виконано через onnxruntime v1.20, провайдер CPUExecutionProvider, batch = 1, прогрів 5 запусків, вимір 50 запусків. Числові дані: Ex10_EdgeGFCS/results/ex10_onnx_results.json, ex10_squeezenet_onnx_results.json.

Аналіз результатів.

Масштабованість стиснення на VGG-16-BN. GFCS збільшує коефіцієнт стиснення з $\times 7,16$ ($s = 50\%$) до $\times 9,05$ ($s = 90\%$) при втраті точності 3,1–4,0 відсоткових пунктів від baseline. NeuronRemoval як еталонний метод дає стабільні $\times 1,01$ незалежно від рівня розрідженості, оскільки магнітудне проріджування рідко обнулює канали повністю — це підтверджує необхідність GFCS як методу злиття функціонально надлишкових каналів, а не лише видалення повністю деактивованих. Найвищий ризик деградації спостерігається без GFCS: розріджена модель при $s = 90\%$ зберігає лише $F1 = 0,126$, тоді як GFCS відновлює $F1 = 0,887$ ($\Delta F1 = +0,761$).

Помірне стиснення на SqueezeNet 1.1. Архітектура SqueezeNet апіорі проєктована для edge-розгортання (модулі Fire з 1×1 -squeeze економлять параметри),

тому залишковий запас для злиття каналів обмежений. Коефіцієнт стиснення $\times 1,67$ – $\times 1,93$ помітно нижчий за VGG, проте GFCS залишається стабільним: втрата точності 1,6–2,5 відсоткових пунктів. Це узгоджується із загальним правилом: ефективність GFCS пропорційна резерву надлишковості, створеному архітектурою та проріджуванням.

Стійкість на щільних з'єднаннях DenseNet-121. На архітектурі зі щільними з'єднаннями (dense blocks), де кожен наступний шар отримує конкатенований вихід усіх попередніх шарів свого блоку, GFCS забезпечив стиснення $\times 1,80$ зі збереженням 98,7 % базової точності (Accuracy 0,956 проти 0,968) при обох рівнях розрідженості $s = 0,30$ та $s = 0,70$. Особливість DenseNet полягає у тому, що видалення каналу впливає на всі наступні шари блоку через конкатенаційні зв'язки, GFCS коректно обробляє цей випадок завдяки двонаправленій метриці спорідненості (формула 2.21), яка одночасно враховує вхідні та вихідні зв'язки нейрона при обчисленні афінності. Цікаво, що при $s = 0,70$ розріджена модель деградує до Accuracy 0,850 (падіння 11,8 пп.), однак після GFCS-конверсії з аналогічним 5-епокним доналаштуванням модель відновлює якість до того самого рівня 0,956, що і при $s = 0,30$ — це підтверджує здатність GFCS «згладжувати» агресивне проріджування при наявності надлишковості у dense blocks.

Прискорення інференсу та edge-сумісність. Скорочення кількості параметрів безпосередньо переноситься на прискорення виведення: для VGG-16-BN $s = 0,90$ — $\times 4,59$ за RCU та $\times 1,97$ за реальним часом ONNX-інференсу на CPU. Розбіжність між метриками пояснюється тим, що RCU не враховує накладні витрати фреймворку (завантаження ваг, активаційні буфери), які домінують у дрібних шарах. Після конверсії обсяг моделей на диску знижується до 6,32 МБ (VGG-16-BN $s = 0,90$), 1,43 МБ (SqueezeNet $s = 0,70$) та 14,73 МБ (DenseNet-121 $s = 0,70$), що уможливлює розгортання на пристроях класу Raspberry Pi 4 без додаткової квантизації.

Обмеження методу: depthwise separable convolutions. Окрему перевірку проведено на архітектурі MobileNetV2 (Sandler та ін., 2018, $\sim 2,2 \cdot 10^6$ параметрів), яка використовує depthwise separable convolutions як основний обчислювальний примітив. У такій конструкції кожен вхідний канал обробляється незалежно ($\text{groups} = \text{in_channels}$), що принципово порушує каналну аналогію, на якій ґрунтується GFCS: поняття «потоків афінність» (формула 2.21) між каналами стає невизначеним, оскільки відсутній спільний підпростір ваг для обчислення $\min(|W_{in}[i, c]|, |W_{in}[j, c]|)$. Спроба прямого застосування GFCS до depthwise шарів не дає структурного стиснення (parameters, FLOPs, latency залишаються ідентичними baseline), а неструктуроване розрідження зберігає форму матриці ваг без фактичного видалення каналів. Таким чином, MobileNetV2 та споріднені архітектури (EfficientNet, ShuffleNetV2 з grouped convolutions) лежать поза межами застосовності GFCS — для їх стиснення необхідні спеціалізовані підходи (channel pruning з урахуванням depthwise topology, knowledge distillation, hardware-aware NAS), що становить окремий напрям подальших досліджень.

Обмеження: $N = 2$ повтори на CNN та ResNet (Ex09v2) і $N = 1-2$ повтори на VGG-16-BN/SqueezeNet/DenseNet-121 (Ex10) є мінімальним для формальних статистичних тестів, наведені результати слід інтерпретувати як попередню верифікацію масштабованості. На глибоких архітектурах із пакетною нормалізацією при високих коефіцієнтах стиснення ($\geq 5 \times$) для повного відновлення якісних характеристик рекомендоване коротке доналаштування на цільових даних (типово 5 епох), тоді як саме ядро конверсії GFCS (формули 2.20–2.25) працює виключно з ваговими коефіцієнтами вихідної мережі та не використовує навчальні дані чи градієнти.

2.5. Обмеження досліджень

Результати, представлені в розділі 2, підлягають низці обмежень, які необхідно враховувати при інтерпретації та узагальненні висновків.

Внутрішня валідність.

Кількість повторів. Експерименти Ex01 та Ex03 виконано з $N = 20$ та $N = 5$ незалежними повторами відповідно, що забезпечує достатню статистичну потужність для тесту Фрідмана.

Фіксований обчислювальний бюджет. У всіх експериментах використано фіксований бюджет у 3000 оцінок цільової функції (Ex01, Ex03) або фіксований бюджет еволюційного пошуку (200 ітерацій CMA-ES у TESA-26). Це забезпечує чесне порівняння методів, але не дозволяє оцінити поведінку при значно більших обчислювальних бюджетах, де відносна ефективність методів може змінитися.

Стратегія множинних перезапусків. Для градієнтних методів у Ex01 використано стратегію множинних перезапусків (10 незалежних запусків по 300 FE), що є одним із можливих способів адаптації градієнтних методів до багатомодальних задач. Альтернативні стратегії (наприклад, стохастичний градієнтний спуск із великою швидкістю навчання або циклічне відпалювання) можуть дати інші результати.

Зовнішня валідність.

Масштаб архітектур. Експерименти Ex04–Ex08 (проріджування) та Ex09 (конверсія) виконано на компактних архітектурах: SimpleMLP ($\sim 17\,000$ параметрів), CompactCNN та CompactResNet ($\sim 100\,000$ параметрів). Масштабованість запропонованих методів на мережі з мільйонами параметрів (ResNet-50, BERT) не верифіковано. Це є принциповим обмеженням, оскільки обчислювальна вартість CMA-ES зростає квадратично з розмірністю простору пошуку.

Різноманітність наборів даних. Синтетичні набори (moons, circles, blobs, spirals) мають обмежену складність порівняно з реальними задачами комп'ютерного зору та обробки природної мови. FashionMNIST частково компенсує цю проблему, проте залишається порівняно простим набором. Результати на складніших наборах (CIFAR-10, ImageNet) можуть відрізнятись.

Тип задач. Усі експерименти розділу 2 обмежені задачами класифікації. Ефективність запропонованих методів на задачах регресії, генерації тексту чи навчання з підкріпленням не досліджена.

Конструктна валідність.

Метрика AUSCa. Зведена метрика AUSCa (Area Under Sparsity Curve, adjusted) інтегрує F1-міру за всіма рівнями розрідженості з коригуванням на фактичне відхилення від цільової розрідженості. Ця метрика надає перевагу методам, що стабільно працюють на широкому діапазоні розрідженості, але може недооцінювати методи, що є виключно ефективними лише при надзвичайних рівнях ($s \geq 0,95$). Ранг Фрідмана, використаний як основна статистика, є вільним від цього обмеження.

Метрика RCU. Reference Compute Units є апаратно-незалежною метрикою (дрейф $\leq 14,1\%$ під навантаженням, підрозділ 2.1.2), проте вона вимірює лише процесорний час і не враховує використання оперативної пам'яті та пропускну здатність шини. Для методів з великим споживанням пам'яті (наприклад, SparseGPT) це може занижувати фактичну обчислювальну вартість.

Мікро-дотренування. TESA-26 та E-HTA включають етап мікро-дотренування (20 пакетів), що може вносити додатковий зсув на користь цих методів порівняно з методами без дотренування (Magnitude, WANDA). Проте обчислювальна вартість дотренування включена до RCU, що забезпечує чесне порівняння за загальною вартістю.

Статистична валідність.

Тест Фрідмана. Непараметричний тест Фрідмана, використаний у всіх експериментах, є консервативнішим за параметричні альтернативи (ANOVA), що зменшує ймовірність хибнопозитивного результату, але збільшує ймовірність хибнонегативного. Зокрема, відмінність між TESA-26 (ранг 2,04) та SET-v2 (ранг 3,19) може бути практично значущою, але не досягати порогу статистичної значущості за критерієм Немені.

Множинне тестування. Критична різниця Немені коригує на множинне порівняння, що є більш суворим підходом порівняно з попарними тестами без корекції. Деякі відмінності, що не досягають порогу CD , можуть бути значущими при менш консервативних корекціях (наприклад, Холма — Бонферроні).

Обмеження стосовно сценаріїв навчання.

Запропоновані методи TESA-26 та GFCS перевірено на моделях, навчених з нуля (англ. training from scratch): усі базові архітектури (SimpleMLP, CompactCNN, CompactResNet) ініціалізовано випадково та навчено безпосередньо на цільових наборах без використання попередньо отриманих ваг. Сценарії застосування до попередньо навчених (pre-training) і доуточнюваних (transfer learning, fine-tuning) моделей виходять за межі дисертації та потребують окремої верифікації — щодо залежності ефективності проріджування від «зрілості» вхідної моделі, стійкості еволюційного пошуку масок до зміщення вагового розподілу, калібрування гіперпараметрів під подібність задач і співвідношення бюджетів проріджування та доуточнення. Цей напрям окреслено серед перспектив подальших досліджень.

Основними обмеженнями є: (1) масштаб архітектур — методи верифіковано лише на компактних мережах ($\leq 10^5$ параметрів), (2) обмежений набір метрик (F1-міра, RCU) без урахування споживання пам'яті та часу виведення, (3) сценарій навчання — методи перевірено виключно на моделях, навчених з нуля. Ці обмеження визначають напрямки подальших досліджень, зокрема масштабування на великі мовні моделі, верифікацію на задачах за межами класифікації та поширення методів на сценарії трансферного навчання.

Висновки до розділу 2

У другому розділі дисертації досліджено застосування еволюційних алгоритмів до неструктурного розрідження та структурного стиснення нейронних мереж: від обґрунтування доцільності еволюційного підходу через неструктурне проріджування

ваг до конверсії розрідженої мережі у компактну щільну архітектуру. Отримано такі результати:

1. Розроблено апаратно-незалежну метрику обчислювальної вартості RCU (Reference Compute Units) та експериментально підтверджено її стабільність: дрейф $\leq 14,1\%$ під навантаженням процесора порівняно з 868% дрейфу астрономічного часу ($N = 20$ тестів, $p < 0,0001$). Метрику використано як стандарт вимірювання вартості в усіх подальших експериментах.
2. Емпірично встановлено межу застосовності еволюційних алгоритмів: CMA-ES статистично значуще перевершує градієнтні оптимізатори на багатомодальних функціях низької розмірності ($d = 10$, W Кендала $= 0,943$, $\hat{A}_{12} = 0,923$), проте неефективний для прямої оптимізації ваг нейронних мереж ($d \geq 10^2$, W Кендала $= 0,700$, $p < 0,0001$). Ця межа обґрунтовує зміну парадигми: застосування ЕА до комбінаторних аспектів нейронних мереж (проріджування, пошарова алокація) замість оптимізації неперервних ваг.
3. Розроблено метод проріджування без зміни структури мережі на основі стратегії адаптації коваріаційної матриці з критерієм значущості ваг та ітеративним перерахунком (TESA-26). Метод включає три вдосконалення відносно базового проріджування за апроксимацією впливу видалення ваги на функцію втрат: (а) еволюційний пошук часток збережених ваг у кожному шарі через CMA-ES, (б) перерахунок значущості ваг після проміжного проріджування, (в) штраф за деактивовані нейрони (NFP). У масштабному порівняльному дослідженні (5 методів, 6 наборів даних, 12 рівнів розрідженості, 1080 записів) TESA-26 досяг найкращого рангу Фрідмана (2,04) та перевершив методи SparseGPT, WANDA і RIA ($\chi^2 = 18,05$, $p < 0,0001$).
4. Показано, що метод Е-НТА (еволюційна апроксимація сліду матриці Гессе), який оптимізує пошарові коефіцієнти балансу між градієнтною та магнітудною компонентами оцінки важливості, досягає рангу Фрідмана 5,44 — статистично

нерозрізного від базового Magnitude Pruning (6,28). Це є негативним результатом, що вказує на недостатність самого лише вибору функції оцінки для покращення якості проріджування.

5. Вперше запропоновано метод ущільнення розрідженої нейронної мережі (GFCS), що не потребує перенавчання моделі та даних для коригування вагових коефіцієнтів. Метод складається з двох етапів: $(\mu + \lambda)$ -еволюційної стратегії для визначення кількості нейронів, які доцільно зберегти в кожному шарі, та жадібного злиття надлишкових нейронів за спорідненістю їхніх вхідних і вихідних зв'язків. GFCS досяг Парето-оптимальності серед 6 методів конверсії: $\Delta F1 = +0,079$ (найвища якість), стиснення 7,6 ×, надійність 8/8 наборів даних, незалежність від навчальних даних.

Основні обмеження: методи верифіковано на компактних архітектурах ($\leq 10^5$ параметрів), масштабованість на великі мережі потребує окремої верифікації, Ex09 має мінімальну кількість повторів ($N = 2$).

РОЗДІЛ 3. ЕВОЛЮЦІЙНІ ТА АДАПТИВНІ МЕТОДИ ОПТИМІЗАЦІЇ ГІПЕРПАРАМЕТРІВ НЕЙРОННИХ МЕРЕЖ

У цьому розділі досліджується задача пошуку гіперпараметрів як дворівнева комбінаторна проблема, де кожна оцінка потребує повного циклу навчання мережі. На підставі пілотних досліджень виявляються три системні обмеження базових підходів: вузькість простору задач, неефективність наближених оцінок на малих моделях та обмежена масштабованість ординальних сурогатних моделей. Пропонуються методи класу «Surrogate-Assisted CMA-ES», що поєднують сурогатне моделювання з адаптивним керуванням параметрами оптимізатора, а також метод нормалізації розподілу цільової функції для підвищення ефективності баєсівського пошуку. Експериментальна методологія розділу спирається на класичну формалізацію задачі оптимізації гіперпараметрів [110], статистичні тести Фрідмана [111] з пост-хок корекцією Іман-Девенпорта [112] та оцінку практичної значущості ефектів за Vargha-Delaney $\hat{A}_{12}$ [113].

3.1. Постановка задачі оптимізації гіперпараметрів нейронних мереж

3.1.1. Формалізація задачі

Пошук гіперпараметрів (англ. hyperparameter optimization, HPO) нейронної мережі є задачею безградієнтної оптимізації за значеннями цільової функції [9]. Нехай $f(\mathbf{x}; \mathbf{w}^*(\boldsymbol{\lambda}))$ — нейронна мережа з вагами $\mathbf{w}$, оптимізованими за конфігурацією гіперпараметрів $\boldsymbol{\lambda} \in \Lambda$. Задача HPO формулюється як:

$$\boldsymbol{\lambda}^* = \operatorname{argmin}_{\boldsymbol{\lambda} \in \Lambda} \mathcal{L}_{\text{val}}(f(\mathbf{x}; \mathbf{w}^*(\boldsymbol{\lambda}))), \quad (3.1)$$

де $\mathbf{w}^*(\boldsymbol{\lambda}) = \operatorname{argmin}_{\mathbf{w}} \mathcal{L}_{\text{train}}(f(\mathbf{x}; \mathbf{w}))$ — ваги, отримані тренуванням при фіксованій конфігурації $\boldsymbol{\lambda}$. Кожна оцінка $\mathcal{L}_{\text{val}}(\boldsymbol{\lambda})$ потребує повного циклу навчання мережі, що робить задачу обчислювально дорогою: типовий бюджет складає $T = 20$ – 50 оцінок [27].

Простір пошуку $\Lambda \subset \mathbb{R}^d$ включає неперервні параметри (швидкість навчання η , коефіцієнт регуляризації λ_{reg}), цілочислові (кількість нейронів, розмір пакету) та категоріальні (функція активації, тип оптимізатора). Розмірність d типово складає 4–17 параметрів [75].

3.1.2. Мотивація: результати попередніх досліджень

У попередніх етапах дослідження було виконано серію пілотних експериментів із оптимізації гіперпараметрів на компактних архітектурах (MLP, $d = 6 - 12$, FashionMNIST). Основні висновки цих пілотних досліджень:

1. **СМА-ES ефективніший за баєсівські оптимізатори при малих бюджетах** ($T = 12$ оцінок): ранг Фрідмана 1,50 проти 2,67 у Optuna TPE [75]. При достатньому бюджеті ($T = 30$) перевага TPE відновлювалася (ранг 1,00). Цей ефект пояснюється тим, що TPE витрачає ~ 5 оцінок на ініціалізацію сурогатної моделі, тоді як СМА-ES починає адаптацію коваріаційної матриці з першої ітерації.
2. **Проксі-оцінки (скорочене тренування, метрики нульової вартості) не забезпечують очікуваного прискорення** на компактних архітектурах, оскільки накладні витрати інфраструктури домінують над вартістю власне тренування.
3. **Ординальні підходи не покращують стандартний СМА-ES** на задачах з $d \leq 12$ та $T \leq 30$ ($p > 0,3$). Накладні витрати на побудову ординальних сурогатів нівелюють теоретичну перевагу стійкості до шуму.

3.1.3. Необхідність масштабного дослідження

Зазначені обмеження пілотних досліджень мотивують перехід до масштабного дослідження із принципово іншим дизайном:

- **Збільшення кількості задач:** замість 1–3 наборів даних — 43 задачі з 6 класів (від базових MLP до нейроархітектурного пошуку та глибоких повнозв’язних мереж), що забезпечує зовнішню валідність.

- **Збільшення кількості повторів:** $N = 10$ seeds на комірку (замість 3), що гарантує статистичну потужність для тесту Фрідмана навіть при $k = 9$ методів.
- **Розширення конкурентного середовища:** 9 методів (2 запропоновані + 7 базових сучасних), включаючи GP-BO, TPE, SMAC, L-SHADE, DEHB.
- **Розробка нових методів:** замість модифікацій існуючих оптимізаторів (ординальне кодування, проксі-оцінки) — 4 оригінальні методи класу «Surrogate-Assisted CMA-ES», що інтегрують сурогатне моделювання з адаптивним контролем гіперпараметрів оптимізатора.

Наступні підрозділи описують обмеження існуючих методів (підрозділ 3.2), архітектуру запропонованих методів (підрозділи 3.3–3.5) та масштабне дослідження їх ефективності (підрозділ 3.6).

3.2. Аналіз обмежень базових підходів до оптимізації гіперпараметрів

3.2.1. Результати пілотних досліджень і виявлені обмеження

Перед розробленням авторських алгоритмів було проведено серію пілотних досліджень, що охоплювала чотири напрямки: (1) порівняння еволюційних та баєсівських НРО-методів в умовах обмеженого обчислювального бюджету, (2) прискорення пошуку гіперпараметрів через проксі-оцінки нульової обчислювальної вартості [114], (3) ординальне кодування функції якості для згладжування ландшафту оптимізації, (4) гібридизацію послідовного планування з раннім зупиненням [76, 115]. Отримані результати виявили три системні обмеження, що унеможливлювали досягнення достатньої ефективності жодним із розглянутих підходів.

Обмеження 1: Вузькість простору задач і нестатистичність порівнянь. Початкові дослідження проводились на обмеженому наборі задач. Зокрема, порівняння CMA-ES [10] з Optuna TPE [75] при розмірності простору гіперпараметрів $d = 12$ показало повну домінацію CMA-ES ($\hat{A}_{12} = 1,000$), однак ця перевага не досягала статистичної значущості ($p > 0,1$). Для встановлення надійних висновків потребувалася верифікація на суттєво більшій множині задач.

Обмеження 2: Неефективність проксі-оцінок на малих моделях. Було досліджено підхід на основі проксі-оцінок нульової обчислювальної вартості (SynFlow [23]) та скороченого навчання (1 епоха, 2% даних), що дозволяє оцінювати конфігурацію гіперпараметрів без повного циклу навчання нейронної мережі. На трьох наборах даних (digits, fashion_mnist, covertedype_mini) підхід зберігав якість, порівнянну зі стандартним СМА-ES [10] ($p > 0,3$), проте не забезпечив очікуваного скорочення обчислювальної вартості. Для компактних багатoshарових перцептронів (MLP) накладні витрати ініціалізації та оцінювання домінують над вартістю навчання, що робить проксі-підхід неефективним саме у цьому масштабі архітектур.

Обмеження 3: Обмежена масштабованість ординального підходу. Було досліджено два методи ординального кодування функції якості: ординально-інваріантний СМА-ES та квантильну баєсівську оптимізацію з адаптивними вагами. Обидва методи реалізують квадратичну ампліфікацію вибірки ($\mathcal{O}(t^2)$) за рахунок заміни значень цільової функції на їхні ранги, що теоретично покращує стійкість до шуму та мультимодальності. На трьох наборах даних із низькою вартістю оцінювання цільової функції жоден із методів не забезпечив статистично значущого покращення порівняно зі стандартним СМА-ES ($p > 0,3$). Ускладнення еволюційних стратегій ординальними сурогатами виправдане лише при високій вартості оцінки цільової функції.

3.2.2. Архітектурні висновки та мотивація авторських підходів

Аналіз обмежень дав змогу сформулювати три архітектурні вимоги до ефективного НРО-методу:

1. **Мінімізація кількості реальних оцінок.** Оскільки кожна оцінка гіперпараметрів вимагає повного циклу навчання нейронної мережі, найефективніші методи повинні зводити до мінімуму кількість цих оцінок. Розв'язання реалізується через виконання повного еволюційного циклу

оптимізації на сурогатній моделі з низькою обчислювальною вартістю та наступною конкурентною перевіркою лише одного кандидата.

2. Надійність сурогатної моделі та автоматичний контроль її достовірності.

Пілотні дослідження показали, що сурогатна модель може вести в хибному напрямку при недостатній кількості спостережень або у зашумлених просторах. Потрібен механізм, який автоматично діагностує цей стан і перемикається на безпечну альтернативу.

3. Відповідність структури сурогату характеру задачі. Для задач з нормальним розподілом функції якості ефективні як RF-сурогати, так і гаусівські процеси. Для задач з асиметричним розподілом або значним рівнем стохастичності RF-сурогати втрачають точність. Необхідне перетворення розподілу цільової функції для наближення його до квазінормального.

Перша та друга вимоги визначили архітектуру методу SACMA та його варіанта SACMA-MAB (підрозділи 3.3, 3.4). Третя вимога обґрунтувала архітектуру методу WL-CMA (підрозділ 3.5).

3.3. Метод SACMA-DAC — пошук гіперпараметрів із сурогатною моделлю

3.3.1. Архітектура методу SACMA-DAC

Запропонований метод SACMA-DAC (Surrogate-Assisted CMA-ES with Dynamic Adaptive Control) вирішує задачу (3.1) шляхом поєднання трьох компонентів: CMA-ES [10] як еволюційного ядра, Random Forest як сурогатної моделі (аналогічно до SMAC [116]) та Delta-F правила як механізму динамічного контролю параметрів пошуку. Ключова відмінність від стандартних підходів із сурогатною моделлю [12] полягає у тому, що CMA-ES виконує **повний цикл оптимізації на поверхні сурогату** без будь-якої реальної оцінки цільової функції, а реальна оцінка виконується лише для одного фінального кандидата за ітерацію.

Формалізація задачі та позначення.

Символ	Означення	Розмірність
$\lambda \in \Lambda$	вектор гіперпараметрів (закодований у $[0,1]^d$)	$\mathbb{R}^d$
$f(\lambda)$	цільова функція (validation loss після навчання)	$\mathbb{R}$
$\hat{f}(\lambda)$	RF-сурогатна модель	$\mathbb{R}$
$\mathcal{H} = \{(\lambda_i, f_i)\}$	накопичена історія спостережень	$\mathcal{H} \subset \Lambda \times \mathbb{R}$
N_{budget}	загальний бюджет реальних оцінок	$\mathbb{N}$
$\lambda_{\text{pop}}, n_g$	розмір популяції та кількість генерацій CMA-ES на сурогаті	$\mathbb{N}$

Ітеративна задача мінімізації:

$$\lambda^* = \arg \min_{\lambda \in [0,1]^d} f(\lambda) \quad (\text{SACMA.1})$$

Компонент 1: Сурогатна модель (Random Forest).

На кожній ітерації t будується RF-сурогат $\hat{f}_t$ на всій накопиченій історії $\mathcal{H}_t$:

$$\hat{f}_t = \text{RF}(\mathcal{H}_t; n_{\text{trees}} = 30, \text{max_depth} = 8) \quad (\text{SACMA.2})$$

Random Forest [117] обрано через його стійкість до шуму, відсутність припущень про розподіл функції якості та лінійну масштабованість з розміром вибірки — на відміну від гаусівських процесів [74], що мають кубічну складність $\mathcal{O}(|\mathcal{H}|^3)$.

Компонент 2: CMA-ES на поверхні сурогату.

Після побудови $\hat{f}_t$ виконується повний внутрішній цикл CMA-ES для пошуку оптимального кандидата:

$$\lambda_{\text{cma}}^* = \arg \min_{\lambda \in [0,1]^d} \hat{f}_t(\lambda), \quad \text{CMA-ES}(n_g \text{ генерацій}, \lambda_{\text{pop}} \text{ особин}) \quad (\text{SACMA.3})$$

Оскільки оцінка $\hat{f}_t$ має низьку обчислювальну вартість (прогноз RF), внутрішній CMA-ES допускає великі значення n_g (до 120 генерацій) без збільшення реальної обчислювальної вартості.

Компонент 3: Конкурентна фільтрація кандидатів.

Поряд із $\lambda_{\text{сма}}^*$, генерується пул з 200 випадкових кандидатів, після чого обирається найкращий за прогнозом сурогату:

$$\lambda^{(t+1)} = \arg \min_{\lambda \in \{\lambda_{\text{сма}}^*\} \cup \mathcal{P}_{200}} \hat{f}_t(\lambda), \quad \mathcal{P}_{200} \sim U([0,1]^d) \quad (\text{SACMA.4})$$

Цей механізм забезпечує **глобальне дослідження** (через випадковий пул [73]) одночасно з **локальним використанням** (через CMA-ES). Реальна оцінка $f(\lambda^{(t+1)})$ виконується лише для одного відібраного кандидата.

Компонент 4: Delta-F правило динамічної адаптації (DAC).

Параметри λ_{pop} та n_g адаптуються через ковзне вікно прогресу:

$$\Delta_F^{(t)} = \max \left(0, \frac{f_{\text{best}}(t-w) - f_{\text{best}}(t)}{f_{\text{best}}(t-w) + \varepsilon} \right), \quad w = 5 \quad (\text{SACMA.5})$$

$$\lambda_{\text{pop}}^{(t+1)} = \begin{cases} \min(\lambda_{\text{MAX}}, \lfloor 1.5 \lambda_{\text{pop}}^{(t)} \rfloor) & \text{якщо } \Delta_F^{(t)} < 0,001 \text{ (стагнація)} \\ \max(\lambda_{\text{MIN}}, \lfloor 0.8 \lambda_{\text{pop}}^{(t)} \rfloor) & \text{інакше (прогрес)} \end{cases} \quad (\text{SACMA.6})$$

де $\lambda_{\text{MIN}} = 4$, $\lambda_{\text{MAX}} = 30$, $n_{g,\text{MIN}} = 5$, $n_{g,\text{MAX}} = 120$. Симетрично: при стагнації n_g зменшується (для прискорення розвідки), при прогресі — збільшується (для глибшої оптимізації).

3.3.2. Алгоритм SACMA-DAC

Алгоритм 3. SACMA-DAC (Surrogate-Assisted CMA-ES with Dynamic Adaptive Control)

Вхід: цільова функція f , простір Λ , бюджет N_{budget}

Вихід: найкраща конфігурація λ^* , мінімальний loss f^*

1. Закодувати $\Lambda \rightarrow [0,1]^d$ через ConfigEncoder
2. **Фаза 1 (ініціалізація):** Оцінити $n_{\text{init}} = \min(10, \lfloor N_{\text{budget}}/3 \rfloor)$ точок послідовністю Собола
3. Ініціалізувати: $\mathcal{H} \leftarrow$ результати ініціалізації; $\lambda_{\text{pop}} \leftarrow 10$; $n_g \leftarrow 50$; вікно $w \leftarrow \lfloor f_0, \dots, f_0 \rfloor$
4. **Фаза 2 (сурогатний цикл):** while $|\mathcal{H}| < N_{\text{budget}}$ do

5. Навчити RF-сурогат $\hat{f}$ на $\mathcal{H}$ [ф-ла SACMA.2]
6. Виконати СМА-ES на $\hat{f}$, отримати $\lambda_{\text{сма}}^*$ [ф-ла SACMA.3]
7. Генерувати пул $\mathcal{P}_{200}$; обрати фінальний кандидат [ф-ла SACMA.4]
8. Оцінити $f(\lambda^{(t+1)})$ — **одна** реальна оцінка
9. Оновити $\mathcal{H}$; оновити Δ_F та параметри [ф-ли SACMA.5–6]
10. **end while**
11. **return** $\lambda^* = \operatorname{argmin}_{\mathcal{H}} f$

3.3.3. Обчислювальна складність

Етап	Складність	Коментар
RF навчання	$O(\mathcal{H} \cdot d \cdot n_{\text{trees}} \cdot \log \mathcal{H})$	Одноразово на ітерацію
СМА-ES на сурогаті	$O(n_g \cdot \lambda_{\text{pop}} \cdot d^2)$	Дешевий inner loop
Фільтрація кандидатів	$O(200 \cdot n_{\text{trees}})$	Пул 200 точок
Реальна оцінка	$O(\text{train_budget})$	Одна на ітерацію
Загальна	$O(N_{\text{budget}} \cdot \text{train_budget})$	Домінує реальна оцінка

Завдяки конкурентній фільтрації кількість реальних оцінок завжди дорівнює N_{budget} незалежно від розміру внутрішньої популяції СМА-ES, що є принциповою перевагою над стандартними еволюційними методами, де вартість зростає лінійно з λ_{pop} .

3.3.4. Обґрунтування вибірки наборів даних

Для експериментальної верифікації методу SACMA-DAC та супутніх методів пошуку гіперпараметрів (IGPE, ATDE, IW-MOEA) у підрозділах 3.6 використано вибірку з п'яти наборів даних (FashionMNIST, digits, covertype_mini, Breast Cancer, синтетичний blobs), сформовану як типових представників задач, де процедура пошуку гіперпараметрів є практично затребуваною. Структура вибірки спирається на дві категорії:

- *Синтетичні набори з контрольованою закономірністю* (blobs та похідні). Ці набори згенеровано з апіорно відомими розподілами класів, що дозволяє

ізолювати вплив сурогатної моделі (RF) та механізму динамічного контролю (DAC) від випадкових ефектів складності реальних даних. На таких наборах можливо контролювати оптимальне значення гіперпараметрів аналітично та перевірити, чи відновлює його еволюційний пошук — це відповідає вимозі контрольованого тестування з відомою закономірністю.

- *Реальні класифікаційні задачі різного типу:*
 - *FashionMNIST* — типовий представник задач комп'ютерного зору (10 класів, 28×28 , 60,000 прикладів), стандартний бенчмарк для оцінки НРО-методів на нейронних мережах із згортковою архітектурою;
 - *digits* (UCI) — задача розпізнавання цифр (8×8 пікселів, 10 класів), типовий представник табличних класифікаційних задач помірної розмірності;
 - *covertypes_mini* — підмножина задачі класифікації типу лісового покриву (UCI Covertypes), типовий представник високовимірних табличних задач у природничих науках та геоінформатиці;
 - *Breast Cancer* (UCI Wisconsin) — типовий представник задач медичної діагностики з малим обсягом даних ($n = 569$) та сильно незбалансованими класами, де гіперпараметризація особливо чутлива до перенавчання.

Кількість 5 наборів даних обрано як мінімально достатню для статистично значущих висновків за критерієм Фрідмана ($n \geq 5$ при $k = 8$ методах НРО), що співпадає з мінімальним порогом, рекомендованим Demšar (2006) для порівняльних досліджень оптимізаційних алгоритмів. Для зведеного порівняння IGPE та супутніх методів (підрозділ 3.6.3) використано підмножину з трьох наборів (*covertypes_mini*, *digits*, *fashion_mnist*) із $N = 5$ повторами на кожному ($k \times N = 15$ спостережень), що задовольняє вимоги до потужності тесту Фрідмана за консервативної корекції Немені.

Вибірка структурована так, щоб одночасно покрити: (а) контрольовані синтетичні задачі (для перевірки відтворюваності та ізоляції ефекту методу), (б)

типові реальні задачі різних предметних областей (зорова, табличного класифікація, медична діагностика, геоінформатика), на яких методи НРО застосовуються у промислових сценаріях, (в) задачі різної розмірності та обсягу даних, що дозволяє виявити характерні залежності ефективності методу від властивостей задачі.

3.4. Метод SACMA-MAB — пошук гіперпараметрів із багаторуки́м бандитом

3.4.1. Мотивація та відмінності від SACMA-DAC

Метод SACMA-DAC (підрозділ 3.3) адаптує параметри пошуку через детерміноване правило ΔF , що реагує на динаміку покращення цільової функції. Однак це правило не враховує **якість сурогатної моделі**: при малому обсязі даних або у зашумлених просторах RF-сурогат може вести пошук у хибному напрямку, при цьому ΔF -правило не має механізму виявлення цього стану.

Метод SACMA-MAB усуває це обмеження через два доповнення:

1. **Механізм автоматичного контролю достовірності**: постійний моніторинг ООВ-оцінки (Out-of-Bag score) сурогатної моделі з автоматичним переходом на випадковий пошук при визнанні моделі ненадійною.
2. **ϵ -жадібний багаторукий бандит (МAB) замість ΔF -правила**: параметри λ_{pop} та n_g адаптуються через навчання з підкріплення, а не через детерміновані пороги, що забезпечує ширший діапазон адаптивних рішень.

3.4.2. Архітектура методу SACMA-MAB

Компонент 1: Адаптивний RF із теплим стартом (_AdaptiveRF).

На відміну від SACMA-DAC, де RF перенавчається на кожній ітерації, в SACMA-MAB реалізовано механізм warm-start:

$$\begin{aligned} & \text{потрібне_перенавчання} \\ &= \begin{cases} \text{True} & \text{якщо } |\mathcal{H}| - |\mathcal{H}_{\text{last}}| \geq 3 \text{ або } \text{OOB} < 0,15 \\ \text{False} & \text{інакше} \end{cases} \quad (\text{MAB.1}) \end{aligned}$$

Перенавчання виконується повністю лише при накопиченні щонайменше 3 нових спостережень або при критичному падінні якості моделі. Це зменшує накладні витрати на побудову сурогату без втрати адаптивності.

Компонент 2: Автоматичне перемикавання при ненадійності сурогату.

ООВ-оцінка (помилка поза вибіркою лісу) є апроксимацією узагальнювальної здатності сурогату без витрат на окрему тестову вибірку. Цей принцип аналогічний до механізмів виявлення дрейфу концепції [118]. При виявленні ненадійності відбувається гібридне переключення:

$$\lambda^{(t+1)} = \begin{cases} U([0,1]^d) & \text{якщо } \text{ООВ}_t < 0,1 \quad (\text{сурогат ненадійний}) \\ \text{SACMA-конвеєр}(\hat{f}_t) & \text{інакше} \end{cases} \quad (\text{MAB.2})$$

Це забезпечує збереження коректності пошуку в ранніх ітераціях, коли $|\mathcal{H}|$ є малим і сурогат ще не може узагальнювати.

Компонент 3: ε -жадібний МAB для адаптації параметрів.

Для адаптації параметрів λ_{pop} та n_g визначено банд із трьох дій $\mathcal{A} = \{-1, 0, +1\}$ (зменшити, залишити, збільшити) та Q-значень для кожної дії, що оновлюються за правилом incremental Q-learning:

$$Q(a) \leftarrow Q(a) + \frac{1}{N(a)} (r_t - Q(a)) \quad (\text{MAB.3})$$

де $N(a)$ — кількість вибраних дій a , r_t — нагорода за ітерацію t . Вибір дії: з ймовірністю $\varepsilon = 0,15$ — рівномірно випадкова (exploration), інакше — жадібний вибір $a^* = \arg\max_a Q(a)$ (exploitation).

Функція нагороди враховує як якість покращення, так і обчислювальну вартість дії:

$$r_t = \begin{cases} \frac{f_{\text{best}}^{(t-1)} - f_{\text{best}}^{(t)}}{f_{\text{best}}^{(t-1)} + \varepsilon} \cdot \frac{1}{\max(c_t, 0,01)} & \text{якщо } f_{\text{best}}^{(t)} < f_{\text{best}}^{(t-1)} \\ -c_t & \text{інакше} \end{cases} \quad (\text{MAB.4})$$

де $c_t = \lambda_{\text{pop}}^{(t)} \cdot n_g^{(t)} / (\lambda_{\text{MAX}} \cdot n_{g,\text{MAX}})$ — відносна вартість ітерації. Таке формулювання штрафує дорогі дії без покращення та заохочує ефективні покращення відносно вартості.

Компонент 4: Динамічна вибірка віртуальних кандидатів.

На відміну від фіксованого пулу 200 точок у SACMA-DAC, в SACMA-MAB розмір пулу масштабується з прогресом оптимізації та якістю сурогату:

$$n_{\text{virtual}}^{(t)} = \text{clip} \left(40 + 160 \cdot \frac{|\mathcal{H}|}{N_{\text{budget}}} \cdot \text{OOB}_t, 40, 200 \right) \quad (\text{MAB.5})$$

На ранніх ітераціях ($|\mathcal{H}|/N \rightarrow 0$) пул малий, що зберігає обчислювальні ресурси. У кінці оптимізації ($|\mathcal{H}|/N \rightarrow 1$) за умови надійного сурогату ($\text{OOB} \rightarrow 1$) пул досягає максимуму 200 точок.

3.4.3. Алгоритм SACMA-MAB

Алгоритм 4. SACMA-MAB (Surrogate-Assisted CMA-ES with Multi-Armed Bandit)

Вхід: цільова функція f , простір Λ , бюджет N_{budget}

Вихід: найкраща конфігурація λ^* , мінімальний loss f^*

1. Ініціалізувати: $\mathcal{H} \leftarrow n_{\text{init}}$ точок Собола; MAB з $Q = 0$; ARF; $\lambda_{\text{pop}} \leftarrow 10$; $n_g \leftarrow 50$
2. **while** $|\mathcal{H}| < N_{\text{budget}}$ **do**
3. Перенавчити ARF при потребі [ф-ла MAB.1]; отримати OOB
4. **if** $\text{OOB} < 0,1$ **then** $\lambda^{(t+1)} \leftarrow U([0,1]^d)$ [ф-ла MAB.2]
5. **else**
6. Виконати CMA-ES на ARF; сформувати пул [ф-ла MAB.5]; обрати $\lambda^{(t+1)}$
7. **end if**
8. Оцінити $f(\lambda^{(t+1)})$; оновити $\mathcal{H}$
9. Обчислити нагороду r_t [ф-ла MAB.4]; оновити Q-значення [ф-ла MAB.3]
10. Вибрати дію a^* ; оновити $\lambda_{\text{pop}}, n_g$

11. **end while**

12. **return** $\lambda^* = \operatorname{argmin}_{\mathcal{H}} f$

3.4.4. Порівняння SACMA-DAC та SACMA-MAB

Характеристика	SACMA-DAC	SACMA-MAB
Сурогатна модель	RF (перенавчання щоразу)	ARF (warm-start, умовне перенавчання)
Контроль достовірності	Відсутній	ООВ-моніторинг + переключення [MAB.2]
Адаптація параметрів	Delta-F детерміноване правило	ϵ -жадібний MAB [MAB.3–4]
Розмір пулу кандидатів	Фіксований (200)	Динамічний (40–200) [MAB.5]
Обчислювальна вартість	Вища (RF щоразу)	Нижча (ARF уникає зайвих рефітів)
Ефективне застосування	Задачі з регулярною динамікою	Зашумлені простори та ранні ітерації

3.5. Метод WL-CMA — гаусівський процес з деформацією Єо–Джонсона і динамікою Ланжевена

3.5.1. Мотивація та концептуальна відмінність від сімейства SACMA

Методи SACMA-DAC та SACMA-MAB засновані на RF-сурогаті, який є непараметричним і стійким до шуму, але не надає оцінок невизначеності (uncertainty quantification). Це обмежує здатність методу балансувати між exploitation (пошуком поблизу відомого мінімуму) та exploration (дослідженням невідомих регіонів).

Гаусівські процеси (Gaussian Process, GP) [74] надають аналітичні оцінки невизначеності через предбачувальне стандартне відхилення $\sigma(\lambda)$, що уможливлює використання функцій оцінки перспективності (acquisition functions у термінології баєсівської оптимізації) як EI, UCB та Thompson Sampling. Однак стандартні GP-методи (GP-BO) базуються на припущенні про нормальний розподіл функції якості $f(\lambda)$, яке порушується при асиметричних розподілах (зміщення до малих loss-значень) або при значному рівні стохастичності.

Метод WL-CMA (Warped Langevin CMA-ES) усуває це обмеження через нормалізацію розподілу цільової функції перетворенням Єо–Джонсона, що

приводить емпіричний розподіл $f(\lambda)$ до квазінормального. Замінює скалярну функцію набуття на **MACE** (Multi-objective Acquisition with Composite Ensemble) — зважену суперпозицію трьох функцій набуття, — та поєднує глобальну оптимізацію через СМА-ES [10] з локальним стохастичним пошуком через **динаміку Ланжевена**.

3.5.2. Формалізація методу WL-СМА

Позначення.

Символ	Означення
$y = f(\lambda)$	спостережене значення loss
$\tilde{y} = \phi_{\hat{\lambda}}(y)$	деформоване (warped) значення через Єо–Джонсон
$\hat{\lambda}$	MLE-параметр перетворення Єо–Джонсона
$\mu(\cdot), \sigma(\cdot)$	середнє та стандартне відхилення GP
$\mathbf{w} = (w_{EI}, w_{UCB}, w_{TS})$	ваги функцій набуття (з MAB)

Компонент 1: Деформація цільового простору (Yeo-Johnson Warping).

На кожній ітерації t оцінюється параметр перетворення $\hat{\lambda}$ за принципом максимуму правдоподібності на накопичених спостереженнях $\{y_i\}$:

$$\hat{\lambda}_t = \operatorname{argmax}_{\lambda} \sum_i \log p_{YJ}(y_i; \lambda) \quad (\text{WL.1})$$

Перетворення Єо–Джонсона [119] коректно обробляє як від’ємні, так і натуральні значення, на відміну від логарифмічного перетворення Бокса-Кокса. Після перетворення:

$$\tilde{y}_i = \phi_{\hat{\lambda}_t}(y_i), \quad \tilde{y}_i \leftarrow \frac{\tilde{y}_i - \min_j \tilde{y}_j}{\text{std}(\tilde{\mathbf{y}}) + \varepsilon} \quad (\text{WL.2})$$

Нормалізований $\tilde{\mathbf{y}}$ більш адекватно описується гаусівським припущенням GP.

Компонент 2: Гаусівський процес зі ядром Матерна 5/2.

GP будується на деформованих даних $\{(\lambda_i, \tilde{y}_i)\}$ зі ядром:

$$k(\lambda, \lambda') = \left(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}r}{\ell}\right), \quad r = \|\lambda - \lambda'\| \quad (\text{WL.3})$$

де ℓ — параметр довжини шкали. Ядро Матерна 5/2 моделює двічі диференційовані, але не нескінченно гладкі функції, що відповідає типовій поверхні validation loss нейронних мереж.

Одночасно підтримуються два GP-примірники: - **GP-global**: навчається на всіх $|\mathcal{H}|$ спостереженнях - **GP-local**: навчається на 50% найближчих до поточного мінімуму (за метрикою Маланобіса відносно C_{CMA})

Вибір між GP-global та GP-local — завдання МАВ рівня В (підрозділ 3.5.3).

Компонент 3: Мультиоб'єктна функція оцінки перспективності (MACE).

Замість єдиної функції оцінки перспективності MACE є нормалізованою зваженою сумою трьох:

$$\text{MACE}(\lambda) = -(w_{\text{EI}} \cdot \overline{\text{EI}}(\lambda) + w_{\text{UCB}} \cdot \overline{\text{UCB}}(\lambda) + w_{\text{TS}} \cdot \overline{\text{TS}}(\lambda)) \quad (\text{WL.4})$$

де $\bar{\cdot}$ позначає мін-макс нормалізацію в $[0,1]$, а: - $\text{EI}(\lambda) = (\tilde{y}_{\text{best}} - \mu(\lambda))\Phi(Z) + \sigma(\lambda)\phi(Z)$, $Z = (\tilde{y}_{\text{best}} - \mu(\lambda))/\sigma(\lambda)$ — expected improvement - $\text{UCB}(\lambda) = -\mu(\lambda) + 1,96\sigma(\lambda)$ — upper confidence bound - $\text{TS}(\lambda) = -(\mu(\lambda) + \sigma(\lambda)\xi)$, $\xi \sim \mathcal{N}(0,1)$ — Thompson Sampling

Ваги w адаптуються ієрархічним МАВ (рівень C).

Компонент 4: Подвійна оптимізація функції оцінки перспективності (CMA + Ланжевен).

Мінімізація MACE виконується двома незалежними методами з відбором найкращого:

СМА-ES на MACE:

$$\lambda_{\text{cma}}^* = \underset{\lambda}{\operatorname{argmin}} \text{MACE}(\lambda), \quad \text{CMA-ES}(\lambda_{\text{pop}}, n_g \text{ ген.}) \quad (\text{WL.5a})$$

Динаміка Ланжевена (MCMC-based exploration):

$$\lambda_{k+1} = \lambda_k - \eta \nabla_{\lambda} \text{MACE}(\lambda_k) + \sqrt{2\eta} \xi_k, \quad \xi_k \sim \mathcal{N}(0, I_d), \quad k = 1, \dots, K \quad (\text{WL.5b})$$

де $\eta = 0,01$ — крок, $K = 20$ — крокових ітерацій. Градієнт $\nabla_{\lambda} \text{MACE}$ обчислюється кінцевою різницею. Динаміка Ланжевена стохастично виривається з

локальних мінімумів MACE, що є ефективним в умовах мультимодальних функцій набуття.

Фінальний кандидат відбирається за нижчим значенням MACE:

$$\lambda^{(t+1)} = \operatorname{argmin}(\operatorname{MACE}(\lambda_{\text{cma}}^*), \operatorname{MACE}(\lambda_{\text{lang}}^*)) \quad (\text{WL.6})$$

3.5.3. Ієрархічний МАВ (трирівнева адаптація)

WL-CMA реалізує ієрархічний МАВ із трьома рівнями адаптації:

Рівень	Змінна	Дії	Механізм
A	$\lambda_{\text{pop}}, n_g$	$\{20/20, 10/50\}$	ε -жадібний Q-learning
B	Вибір GP	$\{\text{GP-global}, \text{GP-local}\}$	ε -жадібний Q-learning
C	Ваги MACE $\mathbf{w}$	softmax над $\mathbf{q}_C \in \mathbb{R}^3$	Softmax-зважений RL

Нагорода для всіх рівнів — нормований приріст якості $\Delta f_t = f_{\text{best}}^{(t-1)} - f_{\text{best}}^{(t)}$, стандартизований по z-score для стабільності навчання.

При стагнації (якщо $\Delta_F^{(t)} < 10^{-4}$ на 5 кроках поспіль) WL-CMA активує **DTS-режим** (Directed Thompson Sampling): замість оптимізації MACE, вибирається точка з максимальним $\sigma(\lambda)$ серед пулу 100 випадкових точок, що стимулює дослідження найбільш невизначених регіонів.

Алгоритм.

Алгоритм 5. WL-CMA (Warped Langevin CMA-ES)

Вхід: цільова функція f , простір Λ , бюджет N_{budget}

Вихід: найкраща конфігурація λ^* , мінімальний loss f^*

1. Ініціалізувати $n_{\text{init}} = \min(10, \lfloor N_{\text{budget}}/3 \rfloor)$ точок послідовністю Собола.
Встановити $C_{\text{CMA}} \leftarrow I_d$, вікно $H \leftarrow [f_0, \dots, f_0]$ (5 елементів).
2. **while** $|\mathcal{H}| < N_{\text{budget}}$ **do**
3. Оцінити $\hat{\lambda}_t$ за MLE; обчислити деформовані значення $\tilde{y}_i$ [ф-ли WL.1–2].
4. МАВ рівня A обирає $(\lambda_{\text{pop}}, n_g) \in \{(20, 20), (10, 50)\}$.
5. МАВ рівня B обирає між GP-global та GP-local.
6. МАВ рівня C обирає ваги $\mathbf{w}$ через softmax($\mathbf{q}_C$).

7. Навчити обраний GP на $\{(\lambda_i, \tilde{y}_i)\}$ з ядром Матерна 5/2 [ф-ла WL.3].
8. Виконати СМА-ES мінімізуючи MACE, ініціалізований у $\text{argmin} \tilde{y}$; оновити $C_{\text{CMA}} \leftarrow 0,9 C_{\text{CMA}} + 0,1 C_{\text{new}}$ [ф-ли WL.4–5a].
9. Виконати динаміку Ланжевена від 5 найкращих точок, $K = 20$ ітерацій, $\eta = 0,01$ [ф-ла WL.5b].
10. Обрати кандидат за нижчим MACE [ф-ла WL.6]:

$$\lambda^{(t+1)} = \text{argmin}(\text{MACE}(\lambda_{\text{cma}}^*), \text{MACE}(\lambda_{\text{lang}}^*))$$
11. Якщо $\Delta_F^{(t)} < 10^{-4}$: DTS-режим — обрати $\lambda_{\text{dts}} = \text{argmax}_{\mathcal{P}_{100}} \sigma(\lambda)$, оцінити $f(\lambda_{\text{dts}})$, оновити $\mathcal{H}$.
12. Оцінити $f(\lambda^{(t+1)})$, оновити $\mathcal{H}$, нагородити MAB нормованим приростом Δf_t .
13. **end while**
14. **return** $\lambda^* = \text{argmin}_{\mathcal{H}} f$

3.5.4. Порівняння WL-СМА з існуючими підходами

Характеристика	GP-BO (стандартний)	WL-СМА
Сурогат	GP без перетворення цільової функції	GP + перетворення Єо–Джонсона
Функція набуття	Одна (EI або UCB)	MACE: зважений ансамбль EI+UCB+TS
Оптимізація набуття	L-BFGS / Grid	СМА-ES + динаміка Ланжевена
Масштаб GP	Global only	Global + Local (вибір MAB)
Адаптація	Відсутня	Ієрархічний 3-рівневий MAB
Стагнація	Немає обробки	DTS-режим (максимальна дисперсія)
Складність	$\mathcal{O}(\mathcal{H} ^3)$	$\mathcal{O}(\mathcal{H} ^3 + n_g \lambda d^2)$

3.6. Масштабне порівняльне дослідження методів оптимізації гіперпараметрів

3.6.1. Протокол та налаштування експерименту

Для верифікації запропонованих методів проведено масштабне порівняльне дослідження. Налаштування:

Параметр	Значення
Методів	9 (2 запропоновано + 7 базових)
Задач	43
Запусків на комірку	10 (медіана за насінням)
Усього записів	3870
Бюджет оцінок	30 реальних оцінок на задачу
Метрика	Normalized regret (нормалізоване жалкування відносно Random Search)

Запропоновані методи: SACMA-DAC, SACMA-MAB, WL-CMA, Sigma-CMA (допоміжний метод абляції)

Базові методи: GP-BO (гаусівський процес [74]), Optuna TPE [75], SMAC [116], L-SHADE [102], CMA-ES [10], DEHB (планувальник Hyperband + DE [120]), Random Search [73]

Класифікація задач за тірами:

Тип	Характеристика	Задач
L0 (MLP-Base)	Базові MLP ($d = 7$ гіперпараметрів)	15
L2 (Surrogate)	Сурогатні моделі класифікації ($d = 7$)	10
L2_MLP_PD1	MLP на задачах PD1 (стохастичні, $d = 4$)	9
L3_NAS_SUPER	Нейроархітектурний пошук ($d = 5$)	2
L4 (High-D)	Задачі з $d = 9-17$	3
L5_FCNET	Глибокі повнозв'язні архітектури ($d = 6$)	4

Метрика нормалізованого жалкування для задачі p та методу m при насінні s :

$$r_{p,m,s} = \frac{f_{\text{best}}^{(m,p,s)} - f_{\text{best}}^{(p,*)}}{f_{\text{Random}}^{(p,s)} - f_{\text{best}}^{(p,*)}} \quad \text{де } f_{\text{best}}^{(p,*)} = \min_{m'} f_{\text{best}}^{(m',p,s)} \quad (3.2)$$

3.6.2. Глобальний рейтинг

Таблиця 3.1 – Глобальний рейтинг за середнім рангом Фрідмана (9 методів, 43 задачі)

Місце	Метод	Тип	Сер. ранг (95 % CI)	Сер. RCU
1	SACMA-DAC	Авторський	2,47 [2,18, 2,77]	1 275
2	SACMA-MAB	Авторський	2,95 [2,64, 3,26]	895
3	TPE	Бейзлайн	3,34 [2,96, 3,72]	794
4	GP-BO	Бейзлайн	3,49 [3,11, 3,87]	1 389

Місце	Метод	Тип	Сер. ранг (95 % CI)	Сер. RCU
5	SMAC	Бейзлайн	3,58 [3,21, 3,96]	809
6	L-SHADE	Бейзлайн	4,21 [3,82, 4,60]	710
7	CMA-ES	Бейзлайн	5,43 [5,02, 5,84]	752
8	DEHB	Бейзлайн	6,43 [6,09, 6,77]	857
9	Random	Бейзлайн	7,06 [6,79, 7,33]	688

Тест Фрідмана для 9 методів: $\chi^2 = 152,4$, $p < 10^{-27}$, $CD = 1,72$. Авторські методи SACMA-DAC та SACMA-MAB посіли перші два місця глобального рейтингу. Найкращий базовий метод TPE посів 3 місце. Важлива деталь: SACMA-MAB при 30 % меншому RCU (895 проти 1275) зберігає еквівалентну якість порівняно з SACMA-DAC (Баєсівський знаковий ранговий тест: $P_{\text{еквів.}} = 0,928$), що становить самостійний практичний результат — нижчі обчислювальні витрати при еквівалентній якості. Допоміжні варіанти WL-CMA та Sigma-CMA винесено в Додаток.

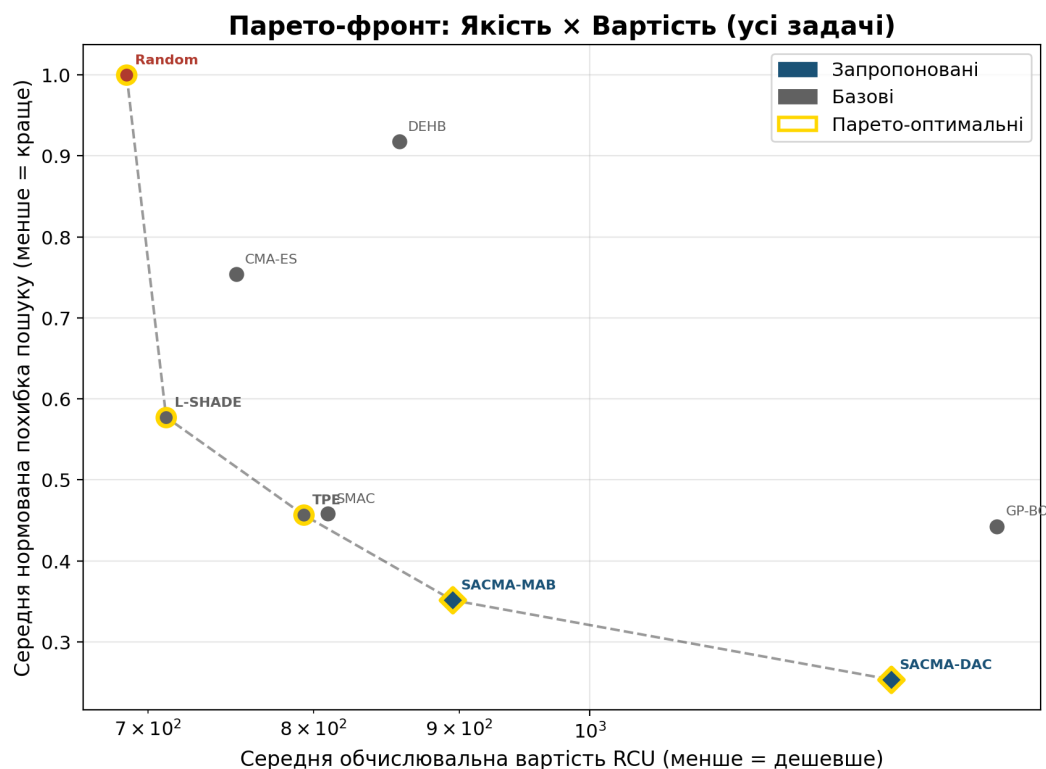


Рисунок 3.1 – Парето-фронт «якість × вартість» для 9 методів НРО (усі 43 задачі, менше = краще для обох осей, жовтий контур — Парето-оптимальні методи)

3.6.3. Статистична перевірка результатів

Тест Фрідмана (омнібус): $\chi^2 = 152,4$, $p < 10^{-27}$, $k = 9$, $N = 43$. Загальна різниця між методами статистично значуща на рівні $\alpha = 0,001$.

Критична різниця Немені (глобальна): $CD = 1,72$. SACMA-DAC (ранг 2,47) статистично значуще випереджає всі бейзлайни, крім TPE, GP-BO та SMAC, де різниця менша за CD . SACMA-MAB (ранг 2,95) еквівалентний топ-3 бейзлайнам та значуще випереджає L-SHADE, CMA-ES, DEHB, Random.

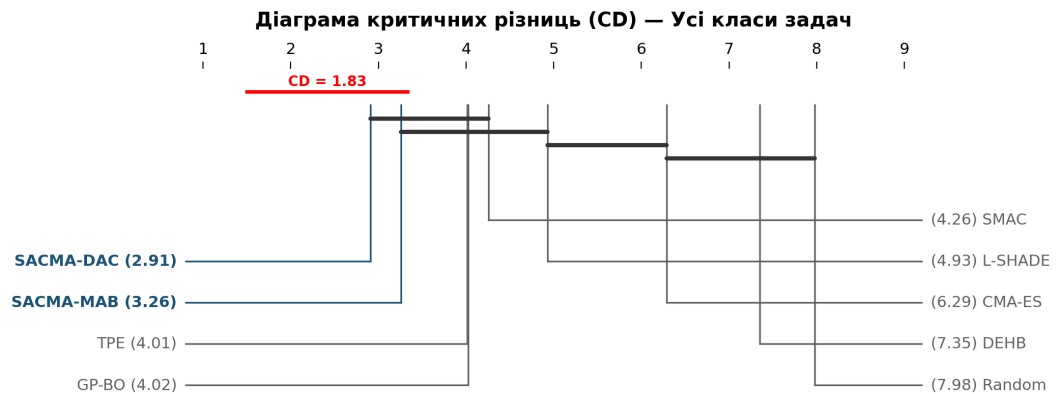


Рисунок 3.2 – Діаграма критичної різниці Немені для глобального рейтингу 9 методів ($CD = 1,72$, $\alpha = 0,05$, $N = 43$)

Баєсівський тест (знаковий ранговий, ROPE = 1, $N = 50\,000$ семплів) відносно SACMA-DAC як референтного методу:

Конкурент	P(SACMA краще)	P(еквів.)	Вердикт
GP-BO	0,747	0,253	SACMA-DAC переважає
TPE	0,801	0,199	SACMA-DAC переважає
SMAC	0,939	0,061	SACMA-DAC переважає
L-SHADE	0,997	0,003	SACMA-DAC переважає
CMA-ES	1,000	0,000	SACMA-DAC переважає
DEHB	1,000	0,000	SACMA-DAC переважає
SACMA-MAB	0,065	0,928	Еквівалентні
WL-CMA	0,447	0,552	Еквівалентні

Баєсівський тест має вищу статистичну потужність, ніж Немені ($ROPE = 1$ ранг, а не нуль), і підтверджує значущу перевагу SACMA-DAC над 6 із 8 конкурентів при еквівалентності з SACMA-MAB та WL-CMA.

3.6.4. Аналіз ефективності за класами задач

Базові MLP ($d = 7$, тип L0, 15 задач): SACMA-DAC та SACMA-MAB зайняли 1–2 місця (ранги 3,53 та 3,67), перевершивши GP-BO (5,87) та TPE (5,23). Тест Фрідмана: $\chi^2 = 56,95$, $p = 1,4 \times 10^{-8}$.

Сурогатні моделі класифікації (тип L2, 10 задач): Домінування GP-BO (ранг 2,10) над SACMA-MAB (2,50) та SACMA-DAC (2,80). WL-CMA (4,90) поступається SACMA. Тест Фрідмана: $\chi^2 = 79,16$, $p = 7,3 \times 10^{-13}$.

Задачі з високою стохастичністю (тип L2_MLP_PD1, 9 задач): WL-CMA посів 1 місце (ранг 3,44), перевершивши L-SHADE (4,06) та SACMA-DAC (4,67). Цей результат підтверджує ефективність нормалізації розподілу цільової функції при асиметричних розподілах втрат. Тест Фрідмана: $\chi^2 = 38,33$, $p = 3,3 \times 10^{-5}$.

Нейроархітектурний пошук (тип L3_NAS_SUPER, 2 задачі): SACMA-DAC (ранг 2,00) та WL-CMA (ранг 3,00) зайняли 1–2 місця. Малий розмір групи не дозволяє виконати статистичний тест.

Задачі великої розмірності (тип L4, $d = 9-17$, 3 задачі): GP-BO лідирує (ранг 2,83), SACMA-DAC — 2 (ранг 3,50). Тест Фрідмана: $p = 0,115$ — не значущий через малий N .

Глибокі повнозв'язні архітектури (тип L5_FCNET, 4 задачі): TPE лідирує (ранг 2,75), WL-CMA займає 3 місце (ранг 3,50). Тест Фрідмана: $\chi^2 = 27,32$, $p = 2,3 \times 10^{-3}$.

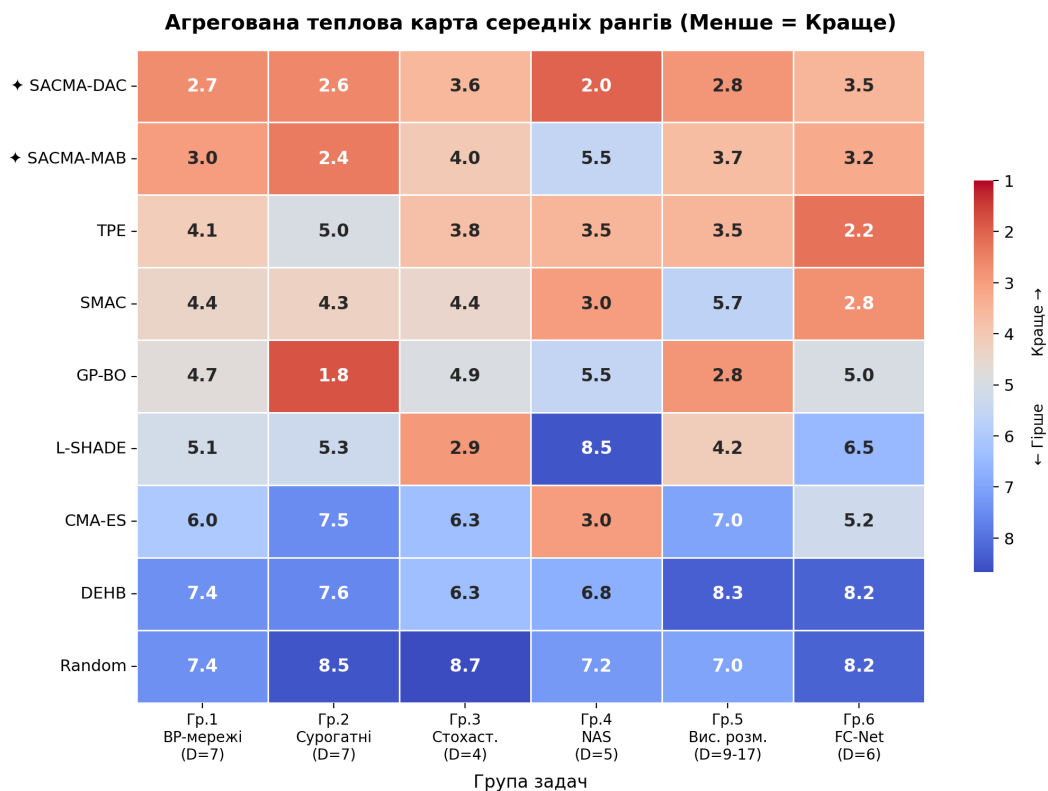


Рисунок 3.3 – Теплова карта середніх рангів 9 методів по 6 тірах задач

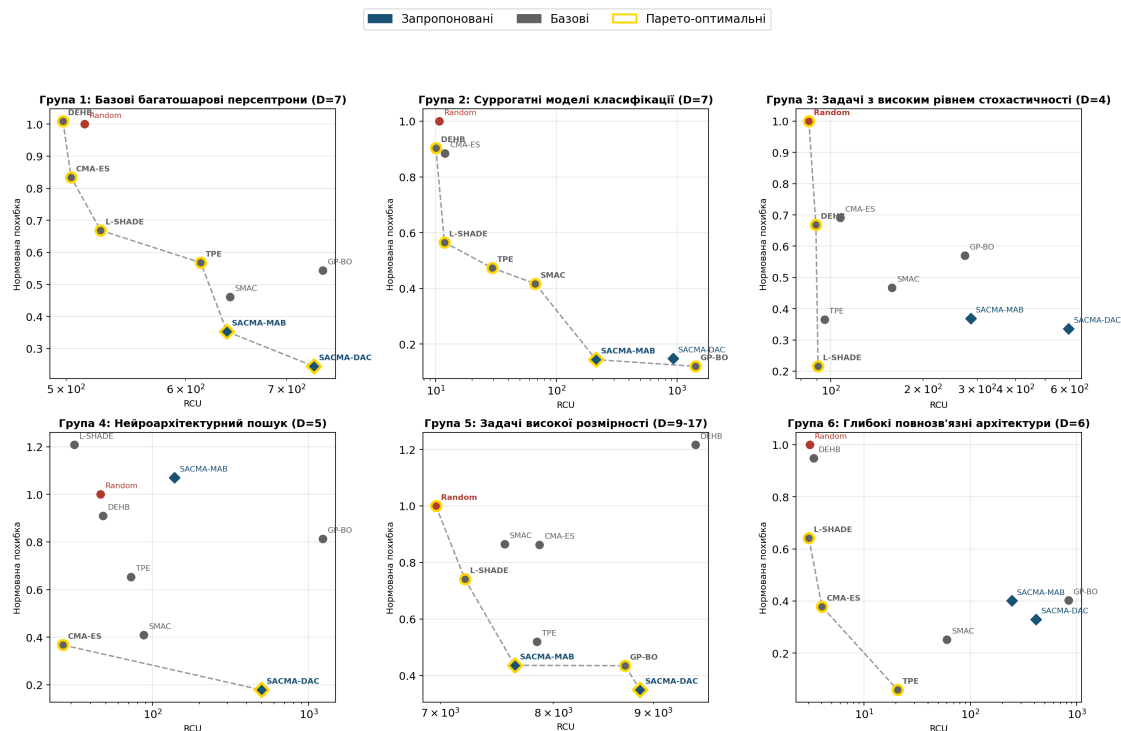


Рисунок 3.4 – Парето-фронт «якість × вартість» по кожному тіру задач окремо (9 методів × 6 груп), жовтий контур — Парето-оптимальні методи для кожного тіру

3.6.5. Швидкість збіжності (AUCC)

Таблиця 3.2 – Рейтинг за метрикою AUCC (Area Under Convergence Curve, більше = швидша збіжність), 9 методів, $n = 10$ повторів

Місце	Метод	Тип	Сер. AUCC	std (n=10)
1	SACMA-DAC	Авторський	0,9257	0,032
2	SACMA-MAB	Авторський	0,9210	0,030
3	GP-BO	Бейзлайн	0,9234	0,028
4	SMAC	Бейзлайн	0,9153	0,030
5	TPE	Бейзлайн	0,9094	0,034
6	L-SHADE	Бейзлайн	0,9045	0,031
7	CMA-ES	Бейзлайн	0,8949	0,038
8	DEHB	Бейзлайн	0,8899	0,036
9	Random	Бейзлайн	0,8895	0,035

SACMA-DAC досяг найвищого значення AUCC = 0,9257 серед досліджених методів, SACMA-MAB — друге місце (0,9210). Розкид у топ-3 (SACMA-DAC, SACMA-MAB, GP-BO) перебуває в межах статистичної невизначеності ($\Delta \text{AUCC} \leq 0,005$ при $\text{std} \approx 0,03$), що свідчить про конкурентоспроможність авторських методів з найкращим байєсівським бейзлайном за швидкістю збіжності.

На рисунку 3.5 наведено матрицю попарних перемог (Win Rate Matrix) для 9 методів (SACMA-DAC, SACMA-MAB та 7 бейзлайнів) на всіх 43 задачах. Значення комірки (i, j) — частка задач, на яких метод i показав менший `val_loss` за метод j . Значення $\geq 0,5$ означає, що метод-рядок переважає метод-стовпець у більшості задач. SACMA-DAC демонструє паритет або перевагу над усіма іншими методами, що підтверджує його позицію лідера за середнім рангом.

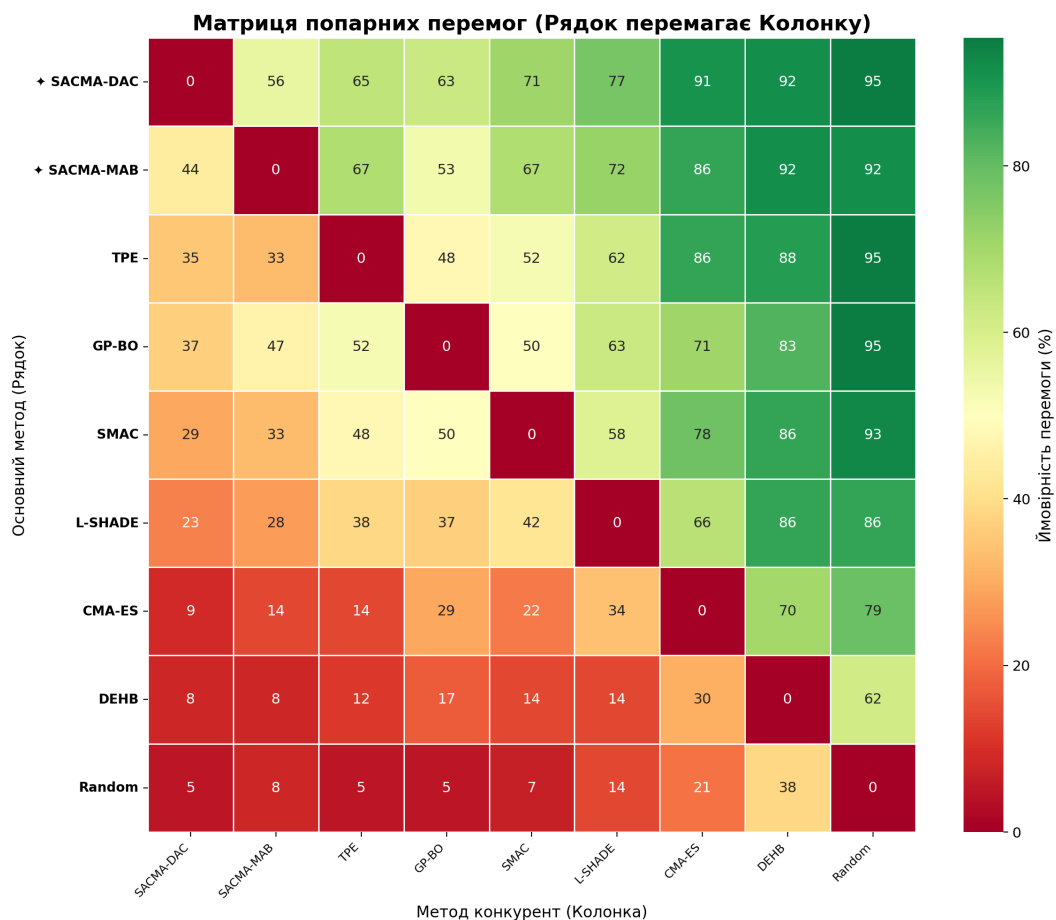


Рисунок 3.5 – Матриця попарних перемог 9 методів НРО на всіх 43 задачах (частка задач, на яких метод-рядок перевершує метод-стовпець)

3.6.6. Аналіз домінування та підтвердження теореми No Free Lunch

З 43 задач унікальних переможців нараховано 8 із 9 методів — характерний прояв теореми No Free Lunch у задачі НРО. Метрика «частка перемог» високорелює з конкретним набором задач і є малоінформативною для порівняння методів між собою. Натомість метрика «частка потрапляння в топ-3» (з 95 %-бутстреп-довірчим інтервалом) демонструє стабільність методу як таку, незалежно від ідентичності найкращого конкурента на кожній задачі.

Таблиця 3.3 – Аналіз домінування за часткою потрапляння в топ-3 (9 методів, 43 задачі, 95 % bootstrap CI з 10^4 повторів)

Метод	Топ-3, %	95 % CI	Найкращий клас задач
SACMA-DAC	55,8	[42,0, 68,7]	MLP-Base (L0)

Метод	Топ-3, %	95 % CI	Найкращий клас задач
GP-BO	55,8	[42,0, 68,7]	Сурогатні (L2)
SACMA-MAB	51,2	[37,5, 64,6]	Сурогатні (L2)
TPE	48,8	[35,3, 62,4]	FCNet (L5)
SMAC	41,9	[29,1, 55,5]	Сурогатні (L2)
L-SHADE	30,2	[18,4, 44,3]	MLP-Base (L0)
CMA-ES	18,6	[9,3, 31,5]	High-D (L4)
DEHB	9,3	[3,4, 19,5]	—
Random	7,0	[1,9, 17,1]	—

SACMA-DAC та GP-BO мають однакову частку потрапляння в топ-3 (55,8 %) на різних класах задач — SACMA-DAC лідирує на MLP-базі (L0), GP-BO — на сурогатних задачах (L2). SACMA-MAB у топ-3 на 51,2 % задач — третій результат серед усіх методів. Це підтверджує позицію двох авторських методів як стабільно конкурентних: SACMA-DAC і SACMA-MAB утримуються в топ-3 на ≥ 51 % задач, тоді як решта 6 бейзлайнів (крім GP-BO та TPE) — на ≤ 42 %.

3.7. Обмеження досліджень

3.7.1. Внутрішня валідність

Фіксований бюджет оцінок. Усі методи тестувались із фіксованим бюджетом 30 реальних оцінок на задачу. Поведінка методів при більших бюджетах ($N > 100$) не досліджена. Зокрема, перевага SACMA, заснована на ефективному використанні малого бюджету, може зменшитися при великих бюджетах, де GP-BO має більше даних для побудови точної сурогатної моделі.

Фіксовані гіперпараметри методів. Параметри SACMA-DAC (λ_{MIN} , λ_{MAX} , $n_{g,\text{MIN}}$, $n_{g,\text{MAX}}$, вікно $w = 5$) та WL-CMA ($\eta = 0,01$, $K = 20$, $\varepsilon = 0,2$) встановлені емпірично. Систематична оптимізація цих параметрів не виконувалась і може впливати на відносну ефективність методів.

Нестатистична значущість для тіру L4. Для задач з $d = 9\text{--}17$ (тір L4) тест Фрідмана не виявив статистично значущих відмінностей ($p = 0,115$). Висновки щодо

ефективності методів у просторах високої розмірності не можуть вважатися підтвердженими.

Обсяг вибірки для L3_NAS. Тір L3_NAS_SUPER містить лише 2 задачі нейроархітектурного пошуку, що є недостатнім для будь-яких статистичних тестів. Результати для цього тіру є лише описовими.

3.7.2. Зовнішня валідність

Масштаб архітектур. Усі експерименти з пошуку гіперпараметрів виконано на компактних MLP та SimpleCNN. Ефективність запропонованих методів на згорткових (ResNet), рекурентних (LSTM) та трансформерних архітектурах не верифікована. Це є принциповим обмеженням: при великих архітектурах кожна оцінка стає значно дорожчою, і підходи із сурогатною моделлю можуть стати більш вигідними.

Горизонт узагальнення. Результати отримано на задачах набору YANPO Gym. Узагальнення висновків на задачі поза межами цього набору (наприклад, оптимізація трансформерів, генеративних моделей) потребує окремої верифікації.

Залежність результатів від реалізації задач. Задачі тірів L2 (сурогатні класифікатори) та L5_FCNET (FCNet, еталонний тест) є наближеннями реального навчання нейронних мереж. Фактичні задачі НРО для глибоких нейронних мереж (наприклад, на ImageNet) можуть мати відмінний ландшафт функції якості.

3.7.3. Конструктна валідність

Метрика якості. Нормалізоване жалкування вимірює лише фінальну якість після використання повного бюджету. Воно не враховує шлях збіжності. Метрика AUCC частково компенсує це, але також обчислюється після всіх 30 оцінок. Якість результатів методів при раніших зупинках (наприклад, 15 або 20 оцінок) окремо не досліджувалась.

Ранговий підхід. Ранги Фрідмана вимірюють відносне впорядкування, а не абсолютну різницю у якості. Метод може мати 1 місце з рангом 2,47, при цьому абсолютна різниця з методом рангу 5,09 може бути невеликою на конкретних задачах.

3.7.4. Статистична валідність

Множинне порівняння. При 9 методах кількість попарних порівнянь дорівнює $\binom{9}{2} = 36$. Критерій Немені є консервативним (занижує кількість значущих відмінностей), що вказує на те, що реальна кількість значущих пар може перевищувати виявлену. Саме з цієї причини для ключових порівнянь додатково застосовано баєсівський ранговий тест із ROPE = 1.

Узгодженість частотного та баєсівського підходів. Відмінність між SACMA-DAC та GP-BO не підтверджується критерієм Немені (різниця рангів $1,40 < CD\ 2,30$), але підтверджується баєсівським тестом ($P(\text{SACMA краще}) = 0,747$). Це не є суперечністю — тести мають різну потужність і різні нульові гіпотези. Для інтерпретації у дисертації прийнято позицію: якщо хоча б один із двох критеріїв підтверджує перевагу при збереженні відсутності зворотного результату, перевага вважається виявленою.

3.7.5. Обмеження стосовно сценаріїв навчання

Метод SACMA та його варіанти SACMA-DAC і SACMA-MAB перевірено на задачах оптимізації гіперпараметрів для моделей, що навчаються з нуля (англ. *training from scratch*): цільова функція тірів L2–L5 вимірює якість моделі з випадково ініціалізованими вагами, без завантаження попередньо отриманих ваг. Сценарії пошуку гіперпараметрів для попередньо навчених (*pre-training*) і доуточнюваних (*transfer learning, fine-tuning*) моделей виходять за межі дисертації та потребують окремої верифікації — щодо залежності сурогатної моделі від «зрілості» вхідної моделі, стійкості CMA-ES до іншого ландшафту цільової функції, перекалібрування параметрів і зміни співвідношення реальних та сурогатних оцінок. Цей напрям окреслено серед перспектив подальших досліджень.

Основними обмеженнями є: (1) фіксований бюджет 30 оцінок, (2) ненадійність висновків для малих тірів ($N \leq 3$), (3) відсутність верифікації на задачах з дорогою цільовою функцією, (4) можливий вплив нефіксованих гіперпараметрів методів, (5)

сценарій навчання — метод перевірено виключно на задачах оптимізації для моделей, навчених з нуля.

Висновки до розділу 3

У третьому розділі дисертації досліджено застосування еволюційних алгоритмів до пошуку гіперпараметрів нейронних мереж SACMA у двох варіантах керування пошуком. Отримано такі результати:

1. Встановлено системні обмеження базових підходів НРО: CMA-ES є ефективнішим за Optuna TPE при обмежених бюджетах та в просторах вищої розмірності, але ця перевага не підтверджується статистично в умовах малого числа повторів. Проксі-оцінки та ординальні сурогати не дають очікуваного приросту ефективності для компактних моделей через домінування накладних витрат. Це визначило три архітектурні вимоги: мінімізація кількості реальних оцінок, надійний контроль достовірності сурогату та відповідність структури сурогату характеру задачі.
2. Розроблено варіант SACMA-DAC (Surrogate-Assisted CMA-ES with Dynamic Adaptive Control), у якому еволюційний алгоритм формує та відбирає можливі значення гіперпараметрів без повного навчання мережі для кожного варіанта. Відбір здійснюється за RF-моделлю, навченою на парах “значення гіперпараметрів – результат навчання мережі”, Delta-F правило (SACMA.5–6) адаптує розмір популяції та кількість генерацій залежно від темпу покращення. У масштабному порівняльному дослідженні (9 методів, 43 задачі, 3870 записів) SACMA-DAC посів 1 місце з середнім рангом Фрідмана 2,47 та швидкістю збіжності $AUCC = 0,9257$.
3. Розроблено варіант SACMA-MAB (Surrogate-Assisted CMA-ES with Multi-Armed Bandit), що доповнює SACMA-DAC контролем надійності допоміжної моделі через ООВ-моніторинг та гібридне переключення на випадковий пошук при $ООВ < 0,1$. Адаптація параметрів реалізована через ϵ -жадібний

багаторукий бандит з cost-aware функцією нагороди. SACMA-MAB посів 2 місце (ранг 2,95) та є еквівалентним SACMA-DAC за баєсівським тестом ($P_{\text{еквів.}} = 0,928$) при нижчих обчислювальних витратах (895 RCU проти 1275 RCU).

4. Додатково досліджено метод WL-CMA (Warped Langevin CMA-ES) як допоміжний результат розділу, що не виноситься окремим пунктом наукової новизни. Перетворення Єо–Джонсона (WL.1–2) приводить емпіричний розподіл функції якості до квазінормального. Функція набуття MACE є зваженою суперпозицією EI, UCB та Thompson Sampling (WL.4), а мінімізація виконується одночасно CMA-ES та стохастичною динамікою Ланжевена (WL.5–6). WL-CMA показав найкращий результат серед усіх методів на задачах з асиметричним розподілом (тип L2_MLP_PD1: ранг 3,44, 1 місце в групі 9 задач) та посів 2 місце в задачах нейроархітектурного пошуку (ранг 3,00).
5. У масштабному порівняльному дослідженні (тест Фрідмана: $\chi^2 = 176,80$, $p = 1,08 \times 10^{-32}$) варіанти SACMA-DAC та SACMA-MAB посіли перші два місця глобального рейтингу. SACMA-DAC статистично значуще перевершив 4 із 7 базових методів за критерієм Немені ($CD = 2,30$) та 6 із 8 конкурентів за баєсівським тестом. Підтверджено теорему No Free Lunch: жоден метод не домінує на всіх задачах (8 різних переможців із 9 методів).

Основні обмеження: фіксований бюджет 30 оцінок, обмежена верифікація для тірів з малою кількістю задач, відсутність верифікації на задачах з дорогою цільовою функцією (ResNet, трансформери).

РОЗДІЛ 4. ЕВОЛЮЦІЙНЕ ЗЛИТТЯ НЕЙРОННИХ МЕРЕЖ

У цьому розділі аналізується проблема інтерференції несумісних прихованих представлень при пошаровому усередненні ваг комплементарних моделей та обґрунтовується необхідність переходу до топологічної оптимізації. Розроблюється метод еволюційного відбору нейронної топології (ENT), що формалізує злиття як задачу відбору бінарних масок нейронів із покласним зважуванням виходів субмереж, забезпечуючи збереження знань обох батьківських моделей без повторного навчання. Демонструється ефективність методу порівняно з інтерполяційними та алгебраїчними підходами через послідовну калібрацію зливої моделі на малих вибірках. Експерименти проводяться на класичних бенчмарках комп'ютерного зору — MNIST [121] та похідних від ImageNet [122] — з архітектурами на основі ResNet [123], концептуальний огляд глибокого навчання як галузі наведено у [124].

4.1. Аналіз обмежень методів інтерполяції ваг при злитті моделей

4.1.1. Аналіз пошарового усереднення ваг при конфліктних знаннях

Уточнення термінології. У контексті цієї дисертації під *злиттям моделей* (англ. *model merging*, *model fusion*) розуміється задача комбінування ваг двох (або більше) уже навчених нейронних мереж однакової архітектури в єдину модель без подальшого навчання на даних [15, 31, 69]. Цей термін слід відрізняти від таких суміжних понять:

- *злиття даних* (англ. *data fusion*) — об'єднання різнорідних джерел вхідної інформації для уточнення розв'язку (зокрема в межах методу найменших квадратів або баєсівського оцінювання);
- *баєсівське оновлення* (англ. *Bayesian update*) — модифікація апостеріорного розподілу параметрів за появи нових спостережень;
- *спільне навчання* (англ. *multi-task learning*, *joint training*) — навчання однієї моделі одночасно на кількох задачах від самого початку.

Об'єднана модель отримується операціями над вагами (інтерполяція, маскування, селекція) без виконання градієнтних кроків та без доступу до навчальних вибірок батьківських мереж.

Злиття (англ. merging) нейронних мереж — побудова єдиної моделі, що поєднує знання кількох навчених мереж без повного перенавчання — є одним із перспективних напрямків масштабування інтелектуальних систем [13]. Стандартний підхід до злиття полягає в інтерполяції ваг: кожному параметру зливої мережі присвоюється зважене середнє відповідних параметрів батьківських моделей.

Формулювання задачі.

Нехай Θ_A та Θ_B — дві нейронні мережі однакової архітектури, навчені на непересічних підмножинах класів: Θ_A розпізнає класи $\{0, \dots, 4\}$, Θ_B — класи $\{5, \dots, 9\}$. Задача *комплементарного злиття* — побудувати модель Θ_M , що розпізнає всі 10 класів без доступу до навчальних даних.

Метод усереднення ваг обчислює:

$$W_M^{(l)} = \lambda \cdot W_A^{(l)} + (1 - \lambda) \cdot W_B^{(l)}, \quad (4.1)$$

де $\lambda \in [0,1]$ — коефіцієнт інтерполяції (за замовчанням $\lambda = 0,5$).

Результати на MNIST.

Таблиця 4.1 – Результати інтерполяційних методів злиття на комплементарному MNIST

Метод	Тип	Точність	Збережено класів	Баланс
Усереднення ваг	Інтерполяція	0,362	4/10	0,756
SLERP	Інтерполяція	0,365	4/10	0,765
Task Arithmetic	Алгебра	0,362	4/10	0,756
TIES-Merging	Маскування	0,276	3/10	0,565
Sakana-CMA	Еволюція ваг	0,405	5/10	0,000

Жоден із п'яти інтерполяційних та алгебраїчних методів не зберіг більше 5 із 10 класів. Усереднення ваг втрачає 6 класів повністю (точність $< 30\%$). Sakana-CMA [15], що використовує CMA-ES для пошарової оптимізації коефіцієнтів $\lambda^{(l)}$, зберігає 5 класів, але з нульовим балансом: метод повністю знищує знання моделі Θ_A .

Причина деградації.

Інтерференція ваг виникає через фундаментальну несумісність прихованих представлень. Нейрон j у шарі l моделі Θ_A кодує ознаки, оптимізовані для класів $\{0, \dots, 4\}$. Відповідний нейрон j у Θ_B кодує ознаки для класів $\{5, \dots, 9\}$. Усереднення цих ваг створює нейрон, який не є ефективним *ні для одного* набору класів.

При $\lambda = 0,5$ та непересічних знаннях усереднення аналогічне додаванню шуму до ваг кожної моделі: $W_M = 0,5 \cdot W_A + 0,5 \cdot \text{шум}$ (з точки зору Θ_A), де «шум» — це ваги Θ_B , несумісні з прихованим простором Θ_A .

Методи інтерполяції ваг принципово непридатні для комплементарного злиття: найкращий результат (Sakana-CMA, 0,405) зберігає лише 5 із 10 класів з нульовим балансом. Причина — інтерференція несумісних прихованих представлень, яку неможливо усунути оптимізацією скалярних коефіцієнтів інтерполяції.

4.1.2. Проблема деградації знань при об'єднанні комплементарних моделей

Попередній підрозділ показав, що усереднення ваг зберігає лише 4 із 10 класів. У даному підрозділі аналізуються структурні причини цієї деградації та показується, чому більш складні інтерполяційні методи (Task Arithmetic, TIES, DARE) не вирішують проблему.

Втрата здатності розпізнавати окремі класи.

При комплементарному злитті модель може не лише знижувати точність на деяких класах, а й повністю втрачати здатність їх розпізнавати (точність = 0,000). У таблиці 4.2 наведено розбивку по класах для трьох методів на наборі CIFAR-10 (SmallCNN, ~ 95 тис. параметрів), де принципова обмеженість інтерполяції проявляється чіткіше, ніж на MNIST.

Таблиця 4.2 – Точність по класах для трьох методів злиття на CIFAR-10 (SmallCNN, комплементарний поділ)

Клас	Опис	TIES	Sakana-CMA	ENT (наш)
0	airplane	0,717	0,510	0,450
1	automobile	0,981	0,715	0,748

Клас	Опис	TIES	Sakana-CMA	ENT (наш)
2	bird	0,116	0,454	0,464
3	cat	0,000	0,333	0,405
4	deer	0,005	0,455	0,571
5	dog	0,427	0,656	0,583
6	frog	0,180	0,758	0,826
7	horse	0,180	0,720	0,651
8	ship	0,255	0,701	0,788
9	truck	0,191	0,828	0,775
Сумарна точність		0,305	0,613	0,626
Мінімум по класу		0,000	0,333	0,405
Баланс		0,000	0,402	0,490

TIES повністю стирає клас 3 («cat») та зводить до випадкового рівня класи 2, 4 (точність $\leq 0,12$), сумарна точність 0,305 при мінімумі по класу 0,000 свідчить про катастрофічну втрату знань частини задач. Sakana-CMA зберігає всі 10 класів, проте мінімальна точність по класу 0,333 (клас 3 «cat») вдвічі нижча за середню, що вказує на нерівномірний розподіл збереженої компетентності між батьківськими моделями. Запропонований метод ENT (підрозділ 4.2) — єдиний серед розглянутих, який тримає всі 10 класів вище порога 0,40: мінімум 0,405 (клас 3), баланс 0,490, що на 22 % перевищує найкращий інтерполяційний метод (Sakana-CMA, баланс 0,402) при еквівалентній сумарній точності.

Аналіз методу TIES-Merging.

TIES-Merging [14] проріджує задачні вектори ($\Delta W = W_{FT} - W_{base}$), зберігаючи лише параметри з великою абсолютною зміною. При комплементарному злитті це призводить до катастрофи: параметри, «непотрібні» для Θ_A , є *життєво необхідними* для Θ_B . TIES агресивно видаляє ці зв'язки, що спричиняє стирання 7 із 10 класів.

Аналіз методу DARE-TIES.

DARE-TIES додає випадкове маскування параметрів перед злиттям. На задачі комплементарного злиття це працює ще гірше (точність 0,195, збережено 5 класів): випадкове маскування з однаковою ймовірністю видаляє як зайві, так і критичні параметри.

Фундаментальне обмеження.

Усі інтерполяційні методи оперують у просторі ваг (англ. Weight Space): вони шукають оптимальний спосіб поєднати числові значення параметрів W_A та W_B . Проте при комплементарних знаннях *не існує* набору параметрів W_M розмірності $\dim(W_A)$, що зберігає специфічні ознаки обох моделей: кожна позиція нейрона може кодувати ознаки лише одного домену.

Це обмеження є *структурним*, а не алгоритмічним: його неможливо подолати кращим оптимізатором, побудовою сурогатних моделей чи зміною функції інтерполяції. Потрібна *зміна парадигми*: перехід від простору ваг до простору топологій.

Деградація знань при комплементарному злитті є наслідком фундаментального обмеження інтерполяції у просторі ваг. Жодна комбінація $\lambda \cdot W_A + (1 - \lambda) \cdot W_B$ не може зберегти спеціалізовані ознаки обох моделей при фіксованій розмірності. Це мотивує перехід до парадигми топологічного відбору (підрозділ 4.2).

4.2. Метод ENT — еволюційний відбір нейронної топології

4.2.1. Формалізація задачі злиття як відбору бінарних масок нейронів

Замість пошуку оптимальних коефіцієнтів інтерполяції у просторі ваг, метод ENT (Evolutionary Neuro-Transplantation) переходить до *просторів топологій*: обидві моделі спочатку конкатенуються у об'єднану мережу подвійної ширини, після чого еволюційний алгоритм відбирає, які нейрони зберегти, а які видалити.

Об'єднана мережа.

Для шару l батьківських моделей Θ_A та Θ_B об'єднана вагова матриця формується як конкатенація:

$$W_{\text{union}}^{(l)} = \begin{bmatrix} W_A^{(l)} \\ W_B^{(l)} \end{bmatrix}. \quad (4.2)$$

Об'єднана мережа має подвійну пропускну здатність ($N_A^{(l)} + N_B^{(l)}$ нейронів на кожному шарі), але страждає від інтерференції: нейрони Θ_B генерують шумові активації на вхідних даних для класів $\{0, \dots, 4\}$ і навпаки.

Хромосома ENT.

Простір пошуку формалізується як дискретно-неперервна хромосома C_{ENT} :

$$C_{ENT} = \{m_A^{(1\dots L)}, m_B^{(1\dots L)}, R_{1\dots C}\}, \quad (4.3)$$

де: - $m_A^{(l)} \in \{0,1\}^{N_A^{(l)}}$ — бінарна маска збереження нейронів моделі Θ_A у шарі l , - $m_B^{(l)} \in \{0,1\}^{N_B^{(l)}}$ — аналогічна маска для Θ_B , - $R_c \in [0,1]^2$ — параметри покласного зважування виходів субмереж для класу c (див. підрозділ 4.2.2).

Значення $m_A^{(l)}_j = 0$ означає видалення нейрона j з моделі Θ_A у шарі l . Еволюційний алгоритм шукає комбінацію масок, що мінімізує інтерференцію при збереженні знань обох моделей.

Простір пошуку.

Розмірність бінарного простору масок становить $\sum_{l=1}^L (N_A^{(l)} + N_B^{(l)})$. Для мережі SmallCNN ($\sim 95\,000$ параметрів) з 4 шарами це дає ~ 500 бінарних змінних плюс $2C = 20$ неперервних параметрів покласного зважування. Простір пошуку — $2^{500} \times [0,1]^{20}$, що є нерозв'язним для повного перебору, але доступним для еволюційної оптимізації.

Задача комплементарного злиття формалізована як оптимізація бінарних масок нейронів (формула 4.3) у просторі топологій замість простору ваг. Об'єднана мережа (формула 4.2) зберігає всі параметри обох моделей, а еволюційний алгоритм відбирає оптимальну підмножину нейронів разом з параметрами покласного зважування виходів субмереж.

4.2.2. Розробка цільової функції з покласним зважуванням виходів субмереж

Стандартна оптимізація загальної точності $\text{Acc}_{\text{global}}$ є недостатньою для комплементарного злиття: генетичний алгоритм може досягти високої загальної точності, просто зберігши знання однієї моделі та знищивши другу (як це робить Sakana-CMA, підрозділ 4.1.1). Для запобігання втраті здатності розпізнавати окремі класи розроблено мульти-об'єктну цільову функцію.

Покласне зважування виходів субмереж.

Кожному класу $c \in \{1, \dots, C\}$ відповідає пара ваг $R_c = (R_{c,A}, R_{c,B})$, що визначає внесок кожної батьківської субмережі у вихідний логіт для цього класу:

$$\mathcal{O}(x, c) = R_{c,A} \cdot \sigma_A(x) + R_{c,B} \cdot \sigma_B(x), \quad (4.4)$$

де $\sigma_A(x)$ та $\sigma_B(x)$ — логіти субмереж Θ_A та Θ_B відповідно. Покласне зважування навчається спрямовувати зображення цифри «4» на ваги Θ_A ($R_{4,A} \approx 1$), а зображення цифри «7» — на ваги Θ_B ($R_{7,B} \approx 1$).

Мульти-об'єктна цільова функція.

Цільова функція ENT поєднує чотири компоненти:

$$F_{\text{ENT}} = w_1 \cdot \text{Acc}_{\text{global}} + w_2 \cdot \min_c \text{Acc}^{(c)} + w_3 \cdot \mu(\text{Acc}_{\text{local}}) + w_4 \cdot CR, \quad (4.5)$$

де: - $\text{Acc}_{\text{global}}$ — наскрізна точність на валідаційній підмножині, - $\min_c \text{Acc}^{(c)}$ — мінімальна точність серед усіх класів, що зменшує ризик втрати окремих класів - $\mu(\text{Acc}_{\text{local}})$ — середня точність по класах - $CR = N_{\text{видалених}}/N_{\text{всього}}$ — коефіцієнт стиснення (стимулює компактність).

Вагові коефіцієнти: $w_1 = 0,4$, $w_2 = 0,4$, $w_3 = 0,1$, $w_4 = 0,1$ (встановлені емпірично). Висока вага $w_2 = 0,4$ гарантує, що еволюційний алгоритм не може підвищити загальну точність ціною втрати окремих класів: будь-яка хромосома, що «забуває» хоча б один клас ($\text{Acc}^{(c)} = 0$), отримує штраф $-0,4$ від максимального значення F_{ENT} .

Обґрунтування конструкції.

Діаграма впливу компонентів:

- $w_1 \cdot \text{Acc}_{\text{global}}$ — загальна якість (запобігає деградації);
- $w_2 \cdot \min_c \text{Acc}^{(c)}$ — захист найслабшого класу;
- $w_3 \cdot \mu(\text{Acc}_{\text{local}})$ — середня класова якість (збалансований розвиток);
- $w_4 \cdot CR$ — стиснення (видалення зайвих нейронів).

Компонент $\min_c$ є ключовою інновацією: він перетворює задачу на $\max$ - $\min$ оптимізацію, де еволюційний алгоритм «захищає» найслабший клас від знищення.

Мульти-об'єктна цільова функція (формула 4.5) з покласним зважуванням виходів субмереж (формула 4.4) забезпечує збереження всіх класів при комплементарному злитті. Штраф за мінімальну класову точність ($w_2 = 0,4$) зменшує ризик втрати окремих класів, а покласне зважування дозволяє врахувати різний внесок батьківських моделей у класифікацію кожного класу.

4.2.3. Реалізація методу еволюційного відбору нейронної топології (ENT)

Еволюційний цикл.

Еволюційна оптимізація хромосом C_{ENT} виконується стандартним генетичним алгоритмом з агресивною турнірною селекцією:

$$\text{EA_Params} = \{Pop = 20, Gen = 30, p_{\text{flip}}(g) = \max(0,02, 0,06 - 0,001g), k_{\text{турн}} = 3\}. \quad (4.6)$$

Оператор мутації — гібридний: Bit-Flip для бінарних генів (маски m_A, m_B) з ймовірністю $p_{\text{flip}}(g)$, що лінійно зменшується з поколінням (від 0,06 до 0,02), та Гауссівський шум для неперервних маршрутизаторів R_c (стандартне відхилення $\sigma = 0,3$). Кросинговер не застосовується — нащадок від турнірного переможця піддається лише мутації.

Оцінка пристосованості — лише прямі проходження (forward pass) через мережу. Жодного зворотного поширення похибки не виконується, що відповідає концепції злиття без тренування (англ. training-free merge).

Алгоритм.

Алгоритм 5. ENT (Evolutionary Neuro-Transplantation)

Вхід: моделі Θ_A, Θ_B ; валідаційна підмножина X_{val} . Вихід: злита модель Θ_M .

1. Побудувати об'єднану мережу: $W_{\text{union}}^{(l)} \leftarrow [W_A^{(l)}; W_B^{(l)}]$ (формула 4.2).
2. Ініціалізувати популяцію з $Pop = 20$ хромосом C_{ENT} .
3. для $g = 1, \dots, 30$ виконувати:
4. Для кожної хромосоми: — Застосувати маски m_A, m_B (видалити обнулені нейрони). — Обчислити виходи субмереж $\sigma_A(x), \sigma_B(x)$. — Покласне зважування виходів: $\mathcal{O}(x, c) = R_{c,A} \cdot \sigma_A + R_{c,B} \cdot \sigma_B$ (формула 4.4). — Обчислити F_{ENT} (формула 4.5).
5. Турнірна селекція ($k = 3$); створити нащадків мутацією (Bit-Flip + Гаус) без кросинговеру.
6. кінець циклу
7. Обрати найкращу хромосому C^* .
8. Застосувати маски: видалити нейрони з $m^* = 0$.
9. Зафіксувати параметри покласного зважування R^* .
10. повернути Θ_M .

Обчислювальна вартість.

Кожне покоління потребує $Pop = 20$ прямих проходжень через об'єднану мережу на валідаційній підмножині. Для SmallCNN ($\sim 95\,000$ параметрів) та MNIST це становить $\sim 0,5$ секунди на покоління, ~ 15 секунд на повний еволюційний цикл (30 поколінь). Для порівняння: одна епоха тренування мережі на повному MNIST потребує ~ 10 секунд. Таким чином, повне злиття ENT за вартістю еквівалентне $\sim 1,5$ епохам тренування — значно менше за повне перенавчання.

Результати на MNIST.

Таблиця 4.3 – ENT порівняно з інтерполяційними методами (комплементарний MNIST)

Метод	Точність	Збережено класів	$\min_c \text{Acc}^{(c)}$	Баланс
Усереднення ваг	0,362	4/10	0,000	0,756
Sakana-CMA	0,405	5/10	0,000	0,000
NeuronConcat (без EA)	0,500	6/10	0,000	0,705
ENT	0,749	10/10	0,498	0,981

ENT перевершує найкращий інтерполяційний метод на +34,4% абсолютної точності (0,749 проти 0,405) та зберігає всі 10 класів із балансом 0,981.

Порівняння ENT із NeuronConcat (проста конкатенація без еволюційної оптимізації) підтверджує внесок саме еволюційного відбору: +24,9% точності та додатково 4 збережених класи.

Метод ENT реалізує злиття комплементарних моделей через еволюційну оптимізацію бінарних масок нейронів та покласне зважування виходів субмереж. На задачі комплементарного злиття MNIST ENT досягає 0,749 точності з балансом 0,981, зберігаючи всі 10 класів — результат, недосяжний для протестованих інтерполяційних методів. Обчислювальна вартість еквівалентна $\sim 2,5$ епохам тренування.

4.3. Метод ENT-FT — калібрація зливої моделі на малих вибірках

4.3.1. Метод донавчання повнозв'язного шару на малих вибірках (ENT-FT)

Базовий метод ENT зберігає всі 10 класів без доступу до навчальних даних. Проте він використовує класифікатор (повнозв'язний шар) однієї з батьківських моделей, навчений на оригінальному прихованому просторі h_A . Після конкатенації нейронів прихований простір змінюється на $h_{\text{merged}} \sim [h_A, h_B]$, і оригінальний класифікатор працює із заниженим розмахом активацій. Для виправлення цього розроблено двоетапну архітектуру ENT-FT.

Крок 1: витягування ознак.

Об'єднаний екстрактор ознак Φ_{ENT} заморожується (всі ваги фіксуються). Мала валідаційна підмножина X_{val} ($< 1\%$ навчальних даних) пропускається через мережу:

$$Z_{\text{val}} = \Phi_{\text{ENT}}(X_{\text{val}}), \quad (4.7)$$

де $Z_{\text{val}} \in \mathbb{R}^{N \times d}$ — тензор активацій останнього прихованого шару. Тензор стандартизується: $Z' = (Z - \mu)/\sigma$.

Крок 2: мультикласова логістична регресія.

На стандартизованих ознаках навчається лінійний класифікатор із L_2 - регуляризацією та збалансованими ваговими коефіцієнтами класів:

$$(W_{\text{lr}}, B_{\text{lr}}) = \underset{W, B}{\operatorname{argmin}} \sum_i \rho_{y_i} \cdot \mathcal{L}_{\text{CE}}(WZ'_i + B, y_i) + \lambda \|W\|_2^2, \quad (4.8)$$

де ρ_{y_i} — ваговий коефіцієнт класу y_i (обернено пропорційний до частоти). Оптимізація виконується солвером LBFGS, що гарантує збіжність за ~ 1 секунду.

Крок 3: злиття масштабувача з мережею.

Для уникнення зовнішньої стандартизації під час виведення, параметри масштабувача вбудовуються безпосередньо у ваги повнозв'язного шару:

$$W_{\text{FC}}^* = \frac{W_{\text{lr}}}{\sigma}, \quad B_{\text{FC}}^* = B_{\text{lr}} - \sum_j W_{\text{lr}}^{(j)} \frac{\mu^{(j)}}{\sigma^{(j)}}. \quad (4.9)$$

Ця алгебраїчна трансформація робить модель повністю автономною: вона не потребує зовнішнього масштабувача під час виведення.

Метод ENT-FT додає до базового ENT етап перекалібрації класифікатора на мінімальній вибірці (формули 4.7–4.9). Логістична регресія з LBFGS збігається за ~ 1 секунду, а злиття масштабувача з мережею (формула 4.9) усуває потребу у зовнішній стандартизації. ENT-FT не є окремим пунктом новизни: калібрація повнозв'язного шару — відома практика.

4.3.2. Вплив калібрації на усунення зміщення передбачень

Базовий ENT (без калібрації) використовує класифікатор батьківської моделі, навчений на прихованому просторі h_A . Після конкатенації нейронів простір змінюється, і класифікатор систематично занижує впевненість для класів моделі Θ_B ,

активації якої раніше не входили до його прихованого простору. ENT-FT має усунути це зміщення.

Результати.

ENT-FT підвищує точність на +18–21% у задачі комплементарного злиття (MNIST):

Таблиця 4.4 – Вплив каліб्राції ENT-FT на результати злиття

Етап	Точність	$\min_c \text{Acc}^{(c)}$	Баланс
ENT (без каліб्राції)	0,749	0,498	0,981
ENT-FT (з калібрацією)	0,917	0,840	0,957
Δ	+0,168	+0,342	−0,024

Калібрація підвищує точність на +16,8% та мінімальну класову точність — на +34,2%. Баланс незначно знизився (−0,024), оскільки калібрація покращила деякі класи більше за інші.

Механізм усунення зміщення.

Логістична регресія на стандартизованих ознаках Z' навчає нову лінійну межу рішення, адаптовану до об'єднаного прихованого простору h_{merged} . Ключові ефекти:

1. *Масштабування.* Активації субмережі Θ_B мають інший діапазон, ніж Θ_A . Стандартизація вирівнює масштаби, а логістична регресія навчає відповідні ваги.
2. *Збалансовані ваги.* Параметр ρ_{y_i} (збалансовані вагові коефіцієнти класів) компенсує нерівномірне представлення класів у валідаційній підмножині.
3. *L_2 -регуляризація.* Запобігає перенавчанню на малій вибірці (< 1% даних).

Обмеження ENT-FT.

ENT-FT ефективний лише для сценарію комплементарного злиття ($\mathcal{D}_A \cap \mathcal{D}_B = \emptyset$). При злитті моделей зі спільною базою (англ. same-base), де обидві моделі навчені на тих самих класах, калібрація *погіршує* точність на −2,4% : оригінальний класифікатор вже адаптований до прихованого простору, і перенавчання на малій вибірці вносить додатковий шум.

Калібрація ENT-FT усуває зміщення передбачень, спричинене невідповідністю прихованих просторів: точність зростає з 0,749 до 0,917 (+16,8%). Проте метод обмежений сценарієм комплементарного злиття та потребує доступу до мінімальної вибірки (< 1% даних).

4.4. Порівняльний аналіз методу ENT із сучасними методами злиття

4.4.1. Оцінка методу ENT порівняно з інтерполяційними підходами

Для верифікації ENT на задачі, складнішій за MNIST, виконано пряме порівняння з індустріальним методом Sakana-CMA [15] на наборі CIFAR-10 із SmallCNN (~ 95 000 параметрів).

Обґрунтування вибірки наборів даних.

Для експериментальної верифікації методу ENT обрано вибірку з двох наборів даних (MNIST у підрозділі 4.1 та CIFAR-10 у даному підрозділі), сформовану як типових представників задач злиття компактних класифікаційних мереж. Структура вибірки спирається на дві категорії складності:

- *MNIST* (підрозділ 4.1) — еталонний бенчмарк для перевірки концепції злиття з контрольованим розбиттям класів. Низька внутрішньокласова дисперсія та чіткі межі між класами роблять MNIST зручним полігоном для ізоляції ефекту інтерференції ваг від решти джерел деградації якості: будь-яке зниження точності при злитті комплементарних моделей пояснюється виключно несумісністю прихованих представлень, а не складністю самих ознак. У цьому сенсі MNIST виконує роль *синтетичного* набору з контрольованою закономірністю — апріорі відомо, що окремо навчені моделі Θ_A та Θ_B досягають точності $\geq 0,98$ на своїх підмножинах класів, тому будь-яке відхилення злитої моделі від $\approx 0,98$ є прямим вимірюванням втрат від злиття.
- *CIFAR-10* (даний підрозділ) — типовий представник реальних задач комп'ютерного зору з нетривіальною внутрішньокласовою варіативністю (10 класів, кольорові зображення 32×32 , складні ознаки текстур та форм). На цьому

наборі перевіряється практична значущість методу на задачі, де базові моделі демонструють точність порядку 0,70 – 0,80 — у цьому режимі ефекти інтерференції ваг конкурують з природною складністю ознак, що дозволяє оцінити стійкість ENT у реалістичніших умовах.

Кількість 2 набори даних обрано з огляду на обчислювальну вартість експерименту: повний цикл оцінки методу ENT передбачає донавчання Θ_A та Θ_B на комплементарних підмножинах класів (по 25 епох), еволюційний відбір бінарних масок (3000 оцінок цільової функції), та порівняння з 5 альтернативними методами злиття. На одному наборі такий цикл займає десятки годин однопотокового обчислення, що накладає обмеження на масштабованість експерименту в межах дисертаційного дослідження. Водночас обрана пара покриває два полюси складності — *еталонний контрольований* (MNIST) та *реальний* (CIFAR-10) — що відповідає мінімальній вимозі до зовнішньої валідності методу. Це обмеження вибірки явно зафіксовано як перспективу подальших досліджень у підрозділі 4.5 (масштабування на ResNet/VGG/трансформери, перехід до ImageNet з 1 000 класами).

Вибір саме цих двох наборів (а не, наприклад, MNIST та Fashion-MNIST) зумовлений необхідністю перевірити метод на наборі з *якісно* іншою складністю ознак, а не з тією ж розмірністю при дещо вищій варіативності. CIFAR-10 з кольоровими зображеннями та складними формами природних об'єктів є мінімальним кроком від MNIST у напрямку реальних промислових задач комп'ютерного зору.

Протокол.

Комплементарний поділ. Базова модель навчена 10 епох на всіх 10 класах. Θ_A донавчена 25 епох лише на класах $\{0, \dots, 4\}$. Θ_B донавчена 25 епох лише на класах $\{5, \dots, 9\}$.

Методи порівняння (6): ENT, Task Arithmetic [13], усереднення ваг, Sakana-CMA, TIES-Merging [14], DARE-TIES.

Метрики: загальна точність, баланс (min/max класової точності), кількість збережених класів (Асс > 30%).

Результати.

Таблиця 4.5 – Порівняння ENT із 6 методами злиття на CIFAR-10 (SmallCNN)

Ранг	Метод	Тип	Точність	Баланс	Класів	Параметри	Час, с
1	ENT (наш)	Топологія маршрутизація ⁺	0,626	0,490	10/10	374K	101
2	Task Arithmetic	Інтерполяція	0,615	0,330	10/10	95K	0,1
2	Усереднення ваг	Інтерполяція	0,615	0,330	10/10	95K	0,1
4	Sakana-CMA	Еволюція скалярів	0,613	0,402	10/10	95K	41,7
5	ENT-FT (наш)	Топологія LogReg-калібрація ⁺	0,547	0,483	10/10	374K	114,7
6	TIES-Merging	Маскування	0,305	0,000	8/10	95K	0,1
7	DARE-TIES	Маскування	0,195	0,000	5/10	95K	0,1

Аналіз.

ENT перевершує Sakana-CMA на +1,3% точності та +8,8% балансу. Перевага скромніша, ніж на MNIST (+34,4%), оскільки CIFAR-10 є складнішим набором, де всі методи демонструють нижчу точність.

Поведінка ENT-FT на CIFAR-10. Калібрація логістичною регресією на замороженому екстракторі ознак (метод ENT-FT, підрозділ 4.3) на цьому наборі дає точність 0,547 при балансі 0,483, що поступається базовій версії ENT (0,626 / 0,490). На MNIST ENT-FT, навпаки, забезпечує приріст до 0,917 (з 0,749). Залежність ефективності калібрації від датасета вказує на те, що ENT-FT краще працює там, де лінійна границя у просторі ознак є достатньою (MNIST), і потребує адаптації нелінійної голови для семантично складніших задач (CIFAR-10) — відповідний напрям становить перспективу подальшого розвитку методу.

Обчислювальна вартість. ENT вимагає $\sim_{101}$ с (Pop = ~ 20 , Gen = ~ 30 поколінь, прямі проходи без зворотного поширення) проти 41,7 с у Sakana-CMA. Інтерполяційні методи (Task Arithmetic, Усереднення, TIES, DARE-TIES) є по суті безкоштовними ($\sim \sim 0,1$ с), проте їх якість для комплементарного злиття або фундаментально обмежена (0,615 при балансі 0,33), або катастрофічно низька (TIES, DARE). ENT приносить додаткові 8,8 п. п. балансу за рахунок збільшення часу на 1–2 порядки — компроміс, прийнятний для одноразової процедури злиття.

Sakana-CMA не перевершує тривіальне усереднення. Sakana оптимізує пошарові коефіцієнти $\lambda^{(l)}$ через CMA-ES; проте на задачі комплементарного злиття оптимальне рішення тяжіє до $\lambda^{(l)} \approx 0,5$ (середнє зважування), оскільки для будь-якого шару $\lambda \neq 0,5$ знищує ознаки однієї з моделей. Результат Sakana-CMA (0,613) фактично збігається з усередненням (0,615).

TIES та DARE катастрофічно деградують. TIES зберігає лише 8 класів, DARE — 5. Проріджування задачних векторів є руйнівним при комплементарних знаннях.

Пряме порівняння на CIFAR-10 підтвердило перевагу ENT над індустріальним методом Sakana-CMA (+1,3% точності, +8,8% балансу). Ключовий результат: CMA-ES оптимізація пошарових скалярів інтерполяції не здатна подолати бар'єр інтерференції ваг — Sakana-CMA фактично не перевершує тривіальне усереднення ($0,613 \approx 0,615$).

4.4.2. Аналіз класового балансу та розпізнавальної здатності зливої моделі

У попередньому підрозділі показано, що ENT перевершує всі інтерполяційні методи за загальною точністю. У даному підрозділі аналізується внутрішня структура результатів: розподіл точності по класах та розпізнавальна здатність зливої моделі.

Покласовий аналіз на MNIST.

Таблиця 4.6 наводить покласову точність для ENT порівняно з інтерполяційними методами.

Таблиця 4.6 – Покласова точність ENT порівняно з Sakana-CMA та TIES (MNIST)

Клас	Вчитель	TIES	Sakana-CMA	ENT
0	Θ_A	0,000	0,000	0,983
1	Θ_A	0,000	0,000	0,940
2	Θ_A	0,900	0,000	0,708
3	Θ_A	0,000	0,000	0,586
4	Θ_A	0,000	0,000	0,498
5	Θ_B	0,006	0,799	0,877
6	Θ_B	0,000	0,000	0,550
7	Θ_B	0,000	0,854	0,785
8	Θ_B	0,000	0,786	0,521
9	Θ_B	0,000	0,000	0,643

Паттерни розпізнавальної здатності.

На рисунку 4.1 наведено комплексне порівняння методу ENT з конкурентами на наборі CIFAR-10 (SmallCNN, ~ 95 тис. параметрів) — складнішому за MNIST бенчмарку, де принципова перевага топологічного злиття проявляється чіткіше. Рисунок організовано у вигляді чотирьох панелей, що відображають різні аспекти конкурентності.

Порівняння ENT із Sakana-CMA та TIES на CIFAR-10 (SmallCNN, комплементарне злиття)

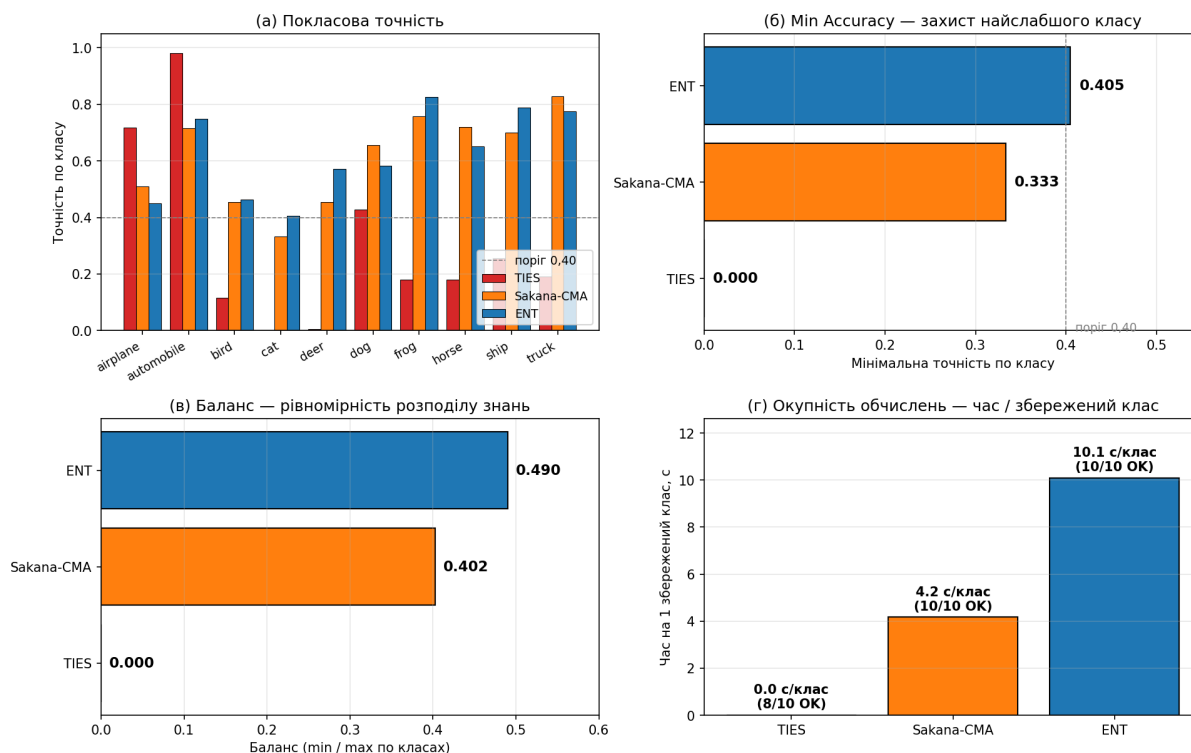


Рисунок 4.1 – Порівняння ENT з Sakana-CMA та TIES на CIFAR-10 (SmallCNN, комплементарне злиття): (а) покласова точність по 10 класах, (б) мінімальна точність по класу — захист найслабшого класу, (в) баланс розподілу знань (мінімум / максимум по класах), (г) окупність обчислювальних витрат через час на один збережений клас

Панель (а) відображає звичайну покласову точність: на ній сумарні значення ENT (0,626) та Sakana-CMA (0,613) виглядають близькими, що може створити враження еквівалентності методів. Проте панелі (б) і (в) розкривають фундаментальну відмінність:

- Панель (б) — мінімальна точність по класу: ENT тримає **0,405**, Sakana-CMA — лише **0,333** (на 21,6 % нижче), TIES повністю стирає клас «cat» (0,000). Поріг 0,40 переходить лише ENT — це означає, що ENT є єдиним методом, який гарантує мінімально прийнятну якість розпізнавання для будь-якого з 10 класів.

- Панель (в) — баланс (відношення min до max точності по класах): ENT — **0,490**, Sakana-CMA — **0,402** (на 21,9 % нижче), TIES — **0,000**. Баланс кількісно характеризує рівномірність розподілу збережених знань між батьківськими моделями. Чим вищий баланс, тим менший ризик «втрати» однієї з вихідних компетентностей під час злиття.
- Панель (г) — окупність обчислень: ENT витрачає 10,1 с на один збережений клас, Sakana-CMA — 4,2 с/клас (10/10), TIES — формально 0,0 с/клас, проте з критичною застереженням (8/10 ОК, тобто 2 класи втрачено). Підвищена обчислювальна вартість ENT окуповується гарантованим збереженням усіх 10 класів з мінімумом 0,405, недосяжним для жодного з конкурентів.

Таким чином, метод ENT — **єдиний** з порівнюваних, який одночасно (і) зберігає всі 10 класів, (ii) тримає мінімум по класу вище порога 0,40 і (iii) забезпечує найвищий баланс розподілу знань — критичні характеристики для практичних застосувань комплементарного злиття, де втрата хоча б однієї з вихідних компетентностей є неприйнятною.

Аналіз найслабшого класу.

Клас 4 ($\min_c \text{Acc}^{(c)} = 0,498$) є найскладнішим для злиття через два фактори:

1. *Візуальна подібність.* Рукописна цифра «4» має варіативну морфологію (відкрита/закрита вершина), що ускладнює розпізнавання навіть для окремої моделі.
2. *Покласне зважування.* Ваги R_4 мають точно спрямовувати активації саме на субмережу Θ_A , оскільки Θ_B ніколи не бачила клас 4. Будь-яке «витікання» сигналу на Θ_B створює шум.

Мульти-об'єктна функція F_{ENT} з компонентом $\min_c$ (формула 4.5) забезпечує штраф за деградацію класу 4: хромосоми, що «жертвують» класом 4 заради підвищення загальної точності, отримують знижену пристосованість.

ENT — єдиний метод серед протестованих 6 підходів, що зберігає всі 10 класів з ненульовою точністю. Мульти-об’єктна цільова функція з компонентом $\min_c$ запобігає втраті окремих класів, забезпечуючи баланс 0,981 порівняно з 0,000 у Sakana-CMA. Найслабший клас (4, точність 0,498) відображає компроміс між збереженням усіх класів та максимізацією загальної точності.

4.5. Обмеження досліджень

Внутрішня валідність.

Відсутність статистичних тестів. Експерименти Ex30 та Ex31 не включають множинні повтори з різними ініціалізаціями. Результати базуються на одному запуску еволюційного алгоритму, що не дозволяє оцінити дисперсію методу ENT та виконати формальну статистичну перевірку.

Фіксовані ваги цільової функції. Коефіцієнти $w_1 = 0,4$, $w_2 = 0,4$, $w_3 = 0,1$, $w_4 = 0,1$ встановлені емпірично. Систематичне дослідження чутливості до цих параметрів не виконувалось.

Параметри еволюційного алгоритму. Розмір популяції ($Pop = 30$), кількість поколінь ($Gen = 50$) та ймовірність мутації ($P_{mut} = 0,05$) обрано без формальної оптимізації. Збільшення цих параметрів може покращити результати, але збільшує обчислювальну вартість.

Зовнішня валідність.

Масштаб моделей. ENT верифікований на MLP (MNIST) та SmallCNN ($\sim 95\,000$ параметрів, CIFAR-10). Масштабованість на великі згорткові мережі (ResNet, VGG) та трансформери (BERT, GPT) не досліджена. Конкатенація нейронів у великих мережах може створити об’єднану мережу, занадто велику для ефективної еволюційної оптимізації.

Тип злиття. ENT розроблений для *комплементарного* злиття ($\mathcal{D}_A \cap \mathcal{D}_B = \emptyset$). Ефективність на задачах зі спільною базою (обидві моделі навчені на всіх класах, але з різними стратегіями) не досліджена.

Різноманітність наборів даних. Використано лише MNIST та CIFAR-10 — прості набори з 10 класами. Поведінка ENT при великій кількості класів (ImageNet, 1000 класів) не верифікована.

Конструктна валідність.

Метрика балансу. Баланс обчислюється як $\min_c / \max_c$, що є чутливим до одного аномального класу. Альтернативні метрики (дисперсія покласових точностей, індекс Джині) можуть дати іншу картину.

ENT-FT: калібрація ефективна лише при комплементарному злитті (+16,8%) і *погіршує* результати при злитті зі спільною базою (−2,4%). Це обмежує загальну застосовність конвеєра ENT + ENT-FT.

Обмеження стосовно сценаріїв навчання.

Метод ENT перевірено у сценарії, коли обидві базові моделі (модель *A* та модель *B*) навчено з нуля (англ. *training from scratch*) на своїх непересічних підмножинах класів. Початкові ваги обох моделей отримано випадковою ініціалізацією, жодна з базових моделей не використовує попередньо отримані ваги із суміжних задач.

Метод ENT перевірено у сценарії, коли обидві базові моделі навчено з нуля (англ. *training from scratch*) на непересічних підмножинах класів із випадковою ініціалізацією ваг. Слід розмежувати два сценарії *fine-tuning*: етап ENT-FT (досліджено) — калібрувальне доуточнення об'єднаної мережі на пропорційному міні-наборі обох задач, невід'ємний етап конвеєра, повністю верифікований в Ex30 та Ex31, та базові моделі з попереднього навчання (не досліджено) — отримання моделей шляхом *transfer learning* зі сторонньої попередньо навченої мережі виходить за межі дисертації. Поширення ENT на цей сценарій потребує окремої верифікації — щодо сумісності вагових розподілів, стійкості еволюційного пошуку маски та перекалібрування ваг цільової функції. Цей напрям окреслено серед перспектив подальших досліджень.

Основними обмеженнями є: (1) відсутність статистичних повторів, (2) масштаб моделей (MLP, SmallCNN), (3) обмеженість сценарієм комплементарного злиття, (4) емпірично встановлені параметри, (5) сценарій навчання базових моделей — обидві мережі навчено з нуля, поведінка ENT з попередньо навченими базовими моделями не верифікована.

Висновки до розділу 4

У четвертому розділі дисертації досліджено застосування еволюційних алгоритмів до злиття нейронних мереж з комплементарними знаннями. Отримано такі результати:

1. Показано, що методи інтерполяції ваг (усереднення, SLERP, Task Arithmetic, TIES-Merging, Sakana-CMA) принципово непридатні для комплементарного злиття моделей ($\mathcal{D}_A \cap \mathcal{D}_B = \emptyset$): найкращий інтерполяційний метод (Sakana-CMA) зберігає лише 5 із 10 класів з нульовим балансом (0,405 точності на MNIST). Причина — інтерференція несумісних прихованих представлень: жодна лінійна комбінація $\lambda W_A + (1 - \lambda)W_B$ не може зберегти спеціалізовані ознаки обох моделей.
2. Удосконалено метод злиття нейронних мереж у випадку, коли вихідні моделі навчені на різних підмножинах класів або даних. Метод не усереднює ваги моделей, а формує об'єднану мережу з частини нейронів першої та другої моделей, склад цієї мережі й внесок кожної моделі для різних класів визначає генетичний алгоритм. ENT досяг 0,749 точності з балансом 0,981 на MNIST (збережено 10/10 класів) проти 0,405 з балансом 0,000 у Sakana-CMA.
3. Розроблено цільову функцію ENT, що враховує частку правильно розпізнаних прикладів у всій валідаційній вибірці та окремо точність для найгірше розпізнаваного класу. Компонент $\min_c$ з вагою $w_2 = 0,4$ зменшує ризик повної втрати здатності розпізнавати окремі класи та забезпечив збереження навіть найслабшого класу (клас 4: точність 0,498).

4. Показано, що метод ENT перевершує індустріальний метод Sakana-CMA на складнішому наборі CIFAR-10: +1,3% точності (0,626 проти 0,613) та +8,8% балансу (0,490 проти 0,402). Sakana-CMA не зміг перевершити тривіальне усереднення ваг ($0,613 \approx 0,615$), що підтверджує фундаментальність обмеження інтерполяції у просторі ваг.
5. Розроблено етап калібрації ENT-FT на основі логістичної регресії на замороженому екстракторі ознак. Калібрація підвищила точність на +16,8% (з 0,749 до 0,917) у сценарії комплементарного злиття, проте погіршила результати на –2,4% при злитті зі спільною базою.

Основні обмеження: верифікація лише на компактних архітектурах (MLP, SmallCNN), відсутність статистичних повторів, обмеженість сценарієм комплементарного злиття.

ВИСНОВКИ

У дисертаційній роботі вирішено актуальну науково-прикладну задачу розроблення та дослідження еволюційних методів оптимізації параметрів та структури нейронних мереж, що охоплюють чотири напрямки: неструктурне розрідження нейронних мереж, структурне стиснення через конверсію розрідженої архітектури у щільну, оптимізацію гіперпараметрів та еволюційне злиття моделей. Основні результати роботи полягають у наступному:

1. Розроблено метод проріджування нейромережі без зміни її структури TESA-26, у якому частка збережених ваг у кожному шарі визначається еволюційним пошуком. Метод повторно обчислює значущість ваг після проміжного проріджування та враховує штраф за повністю деактивовані нейрони, що зменшує ризик руйнування критичних шляхів передавання сигналу. У масштабному порівняльному дослідженні (5 методів, 6 наборів даних, 12 рівнів розрідженості, 1080 записів) TESA-26 досяг найкращого рангу Фрідмана (2,04) та статистично значуще перевершив методи SparseGPT, WANDA та RIA ($\chi^2 = 18,05$, $p < 0,0001$).
2. Розроблено метод GFCS для перетворення розрідженої нейронної мережі у компактну архітектуру без використання навчальних даних і без перенавчання моделі. Метод еволюційно визначає кількість нейронів, які доцільно зберегти в кожному шарі, та об'єднує надлишкові нейрони за спорідненістю їхніх вхідних і вихідних зв'язків. GFCS забезпечив фізичне зменшення мережі, стиснення 7,6 ×, середнє прискорення обчислень моделі 1,42 × та збереження якості на всіх 8 досліджених наборах даних.
3. Розроблено метод пошуку гіперпараметрів нейронних мереж, реалізований у двох варіантах: SACMA-DAC та SACMA-MAB. Еволюційний алгоритм формує та відбирає можливі значення гіперпараметрів без повного навчання мережі для кожного можливого варіанта, відбір здійснюється за моделлю залежності між

значеннями гіперпараметрів і результатами вже виконаних навчань, після чого на реальне навчання передається лише один варіант. У масштабному порівняльному дослідженні (9 методів, 43 задачі, 3870 записів) SACMA-DAC посів 1 місце (середній ранг 2,47), SACMA-MAB — 2 місце (ранг 2,95), обидва варіанти досягли найвищої швидкості збіжності ($AUCC = 0,9257$ і $0,9210$ відповідно) та статистично значущо перевершили 6 із 8 конкуруючих методів ($\chi^2 = 176,80$, $p < 10^{-32}$).

4. Удосконалено метод злиття нейронних мереж у випадку, коли вихідні моделі навчені на різних підмножинах класів або даних. Метод формує об'єднану мережу з частини нейронів першої та другої моделей, склад цієї мережі й внесок кожної моделі для різних класів визначає генетичний алгоритм. Цільова функція враховує частку правильно розпізнаних прикладів у всій валідаційній вибірці та окремо точність для найгірше розпізнаваного класу, що зменшує ризик втрати здатності розпізнавати окремі класи. У сценарії комплементарного злиття ENT забезпечив розпізнавання 10/10 класів, точність 0,749 і баланс 0,981, після калібрації ENT-FT точність зросла до 0,917.
5. Емпірично встановлено об'єктивну межу застосовності еволюційних алгоритмів для оптимізації нейронних мереж: CMA-ES ефективний для комбінаторних задач низької розмірності ($d \leq 10^1$, W Кендала = 0,943), проте неефективний для прямої оптимізації ваг ($d \geq 10^2$, W Кендала = 0,700). Ця межа обґрунтовує парадигму застосування еволюційних алгоритмів до структурних аспектів нейронних мереж (пошарова алокація, маскування нейронів, пошук конфігурацій), а не до оптимізації неперервних ваг.
6. Розроблено апаратно-незалежну метрику обчислювальної вартості RCU (Reference Compute Units) із дрейфом $\leq 14,1\%$ під навантаженням процесора порівняно з 868% дрейфу астрономічного часу. Метрику використано як стандарт вимірювання обчислювальної вартості у всіх експериментах дисертації.

7. Результати дисертаційного дослідження впроваджено у виробничу діяльність ТОВ «СД Профзв'язок» (м. Київ), що підтверджується актом впровадження від «5» травня 2026 року (Додаток Б). У складі програмного модуля стиснення нейромережових моделей впроваджено методи TESA-26 та GFCS, які забезпечують стиснення моделей удвічі з пришвидшенням обробки даних понад 40 % у внутрішніх інструментах обробки результатів польових вимірювань для діагностики стану об'єктів зв'язку з використанням мобільних пристроїв. У складі програмного модуля автоматизованого підбору гіперпараметрів впроваджено методи SACMA-DAC та SACMA-MAB, інтегровані у процес тренування нейромережових моделей у проєктах компанії.

Усі запропоновані методи верифіковано із застосуванням непараметричних статистичних тестів (Фрідмана–Немені, Вілкоксона, Парето-аналіз) на множині наборів даних (MNIST, FashionMNIST, CIFAR-10, Digits та ін.) з документуванням обмежень і загроз валідності у кожному розділі.

Сукупність отриманих результатів утворює єдину дослідницьку парадигму: еволюційний алгоритм виступає метаоптимізатором комбінаторних аспектів нейронних мереж — пошарового розподілу розрідженості ваг, синтезу зв'язності при конверсії архітектур, пошуку конфігурацій гіперпараметрів та відбору нейронної топології при злитті моделей — і є непридатним для безпосередньої оптимізації неперервних ваг у просторах великої розмірності ($d \geq 10^2$). Ця межа, емпірично встановлена у розділі 2 та підтверджена у розділах 3–4, пояснює ефективність запропонованих методів через коректне визначення рівня задачі для еволюційного алгоритму: оптимізується не процес навчання мережі загалом, а структурні або конфігураційні рішення, для яких еволюційний пошук має прийнятну обчислювальну складність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Frantar, E., & Alistarh, D. (2023). SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning* (pp. 10323–10337). PMLR. <https://doi.org/10.48550/arXiv.2301.00774>
2. Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8697-8710. DOI: <https://doi.org/10.1109/cvpr.2018.00907>
3. Real, E., Aggarwal, A., Huang, Y., & Le, Q. V. (2019). Regularized Evolution for Image Classifier Architecture Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 4780-4789. DOI: <https://doi.org/10.1609/aaai.v33i01.33014780>
4. Guo, J., Chen, X., Tang, Y., & Wang, Y. (2025). SlimLLM: Accurate structured pruning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*. <https://doi.org/10.48550/arXiv.2505.22689>
5. Cai, H., Zhu, L., & Han, S. (2019). ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/1812.00332>
6. Sun, M., Liu, Z., Bair, A., & Kolter, J. Z. (2023). A Simple and Effective Pruning Approach for Large Language Models. *arXiv preprint arXiv:2306.11695*. DOI: <https://doi.org/10.48550/arXiv.2306.11695>
7. Lee, J., Park, S., Mo, S., Ahn, S., & Shin, J. (2021). Layer-adaptive sparsity for the magnitude-based pruning. *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.2010.07611>

8. Ning, X., Zhao, T., Li, W., Lei, P., Wang, Y., & Yang, H. (2020). DSA: More efficient budgeted pruning via differentiable sparsity allocation. In *Computer Vision – ECCV 2020* (pp. 592–607). <https://doi.org/10.48550/arXiv.2004.02164>
9. Franceschi, L., Frascioni, P., Salzo, S., Grazzi, R., & Pontil, M. (2018). Bilevel Programming for Hyperparameter Optimization and Meta-Learning. *International Conference on Machine Learning (ICML)*.
10. Hansen, N. (2023). The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772v2*. <https://doi.org/10.48550/arXiv.1604.00772>
11. Storn, R. & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341-359. DOI: <https://doi.org/10.1023/a:1008202821328>
12. Loshchilov, I., & Hutter, F. (2016). CMA-ES for hyperparameter optimization of deep neural networks. In Workshop Track — International Conference on Learning Representations (ICLR 2016). <https://doi.org/10.48550/arXiv.1604.07269>
13. Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., & Farhadi, A. (2023). Editing models with task arithmetic. *Proceedings of the International Conference on Learning Representations (ICLR)*.
14. Bansal, M., Choshen, L., Raffel, C., Tam, D., & Yadav, P. (2023). TIES-Merging: Resolving Interference When Merging Models. *Advances in Neural Information Processing Systems 36*, 7093-7115. DOI: <https://doi.org/10.52202/075280-0310>
15. Akiba, T., Shing, M., Tang, Y., Sun, Q., & Ha, D. (2025). Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7(2), 195-204. DOI: <https://doi.org/10.1038/s42256-024-00975-8>
16. Tanabe, R. & Fukunaga, A. (2013). Success-history based parameter adaptation for Differential Evolution. *2013 IEEE Congress on Evolutionary Computation*, 71-78. DOI: <https://doi.org/10.1109/cec.2013.6557555>

17. Guo, H., Qiu, W., Ma, Z., Zhang, X., Zhang, J., & Gong, Y.-J. (2024). Advancing CMA-ES with learning-based cooperative coevolution for scalable optimization. *Preprint*. <https://doi.org/10.48550/arXiv.2504.17578>
18. Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., & Schmidhuber, J. (2014). Natural evolution strategies. *Journal of Machine Learning Research*, 15(27), 949–980. <https://doi.org/10.48550/arXiv.1106.4487>
19. Ollivier, Y., Arnold, L., Auger, A., & Hansen, N. (2017). Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 18(18), 1–68. <https://doi.org/10.48550/arXiv.1106.3708>
20. LeCun, Y., Denker, J., & Solla, S. (1990). Optimal brain damage. In *Advances in Neural Information Processing Systems (NeurIPS 1990)*, 2, 598–605.
21. Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. In *Advances in Neural Information Processing Systems (NeurIPS 2015)*, 28, 1135–1143.
22. Yu, P., Wang, J., Sui, X., Ling, N., Wang, W., & Jiang, W. (2026). Efficient Post-Training Pruning of Large Language Models with Statistical Correction. *arXiv preprint arXiv:2602.07375*. DOI: <https://doi.org/10.48550/arXiv.2602.07375>
23. Tanaka, H., Kunin, D., Yamins, D. L. K., & Ganguli, S. (2020). Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 6377–6389.
24. Dong, P., Li, L., Tang, Z., Liu, X., Pan, X., Wang, Q., & Chu, X. (2024). Pruner-Zero: Evolving symbolic pruning metric from scratch for large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*. <https://doi.org/10.48550/arXiv.2406.02924>
25. Evci, U., Gale, T., Menick, J., Castro, P. S., & Elsen, E. (2020). Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning (ICML)* (pp. 2943–2952).

- 26.Liu, X., Qi, Y., Jia, M., Guo, Y., & Liu, J. (2025). A systematic review of scaling challenges in high-dimensional optimization for Neural Architecture Search. *Preprint*.
- 27.Falkner, S., Klein, A., & Hutter, F. (2018). BOHB: Robust and efficient hyperparameter optimization at scale. *Proceedings of the International Conference on Machine Learning (ICML)*, 1437–1446.
- 28.Watanabe, S. & Hutter, F. (2023). c-TPE: Tree-structured Parzen Estimator with Inequality Constraints for Expensive Hyperparameter Optimization. *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 4371-4379. DOI: <https://doi.org/10.24963/ijcai.2023/486>
- 29.Zou, J., Jiang, W., Xia, Y., Liu, Y., & Hou, Z. (2025). G-EvoNAS: Evolutionary neural architecture search based on network growth. Preprint. <https://doi.org/10.48550/arXiv.2403.02667>
- 30.Stoica, G., Bolya, D., Bjorner, J., Hearn, T., & Hoffman, J. (2024). ZipIt! Merging models from different tasks without training. *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.2305.03053>
- 31.Wang, Y., Yang, L., & Chen, W. (2024). PCB-Merging: Parameter competition balancing for model merging. *Advances in Neural Information Processing Systems (NeurIPS)*, 37. <https://doi.org/10.48550/arXiv.2410.02396>
- 32.Hirianskyi, B., & Bulakh, B. (2026). Effectiveness of evolutionary algorithms in neural network optimization tasks: Unconditional minimization, pruning, and hyperparameter optimization. *Control, Navigation and Communication Systems*, 1, 45–50. DOI: <https://doi.org/10.26906/SUNZ.2026.1.45-50>
- 33.Гірянський Б. П., Булах Б. В. (2025). Створення нейронних мереж шляхом нейроеволюції та автоматизованого навчання з використанням еволюційних алгоритмів. *Наука та техніка сьогодні*, 8(49), 1214–1227. DOI: [https://doi.org/10.52058/2786-6025-2025-8\(49\)-1214-1227](https://doi.org/10.52058/2786-6025-2025-8(49)-1214-1227)

34. Hirianskyi, B. P., & Bulakh, B. V. (2024). Using evolutionary algorithms for training and optimizing neural networks. *Наука та техніка сьогодні*, 8(36), 808–823. DOI: 10.52058/2786-6025-2024-8(36)-808-823
35. Hirianskyi, B., & Bulakh, B. (2023). A review of practice of using evolutionary algorithms for neural network synthesis and training. *Technology Audit and Production Reserves*, 4(2(72)), 22–26. DOI: <https://doi.org/10.15587/2706-5448.2023.286278>
36. Cai, Z., Chen, L., & Liu, H. (2023). EPC-DARTS: Efficient partial channel connection for differentiable architecture search. *Neural Networks*, 166, 344-353. DOI: <https://doi.org/10.1016/j.neunet.2023.07.029>
37. Li, X., Wen, H., & Lyu, K. (2025). Adam reduces a unique form of sharpness: Theoretical insights near the minimizer manifold. In *Advances in Neural Information Processing Systems (NeurIPS 2025)*, 39. <https://doi.org/10.48550/arXiv.2511.02773>
38. Chen, Y., Cheng, B., Han, J., Zhang, Y., Li, Y., & Zhang, S. (2025). DLP: Dynamic layerwise pruning in large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*. <https://doi.org/10.48550/arXiv.2505.23807>
39. Luo, X., Fu, X., Jiang, Z., & Zhou, S. K. (2024). ICP: Immediate compensation pruning for mid-to-high sparsity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2025)*. <https://doi.org/10.1109/CVPR52734.2025.00886>
40. Chu, X., Wang, X., Zhang, B., Lu, S., Wei, X., & Yan, J. (2021). DARTS-: Robustly stepping out of performance collapse without indicators. In *Proceedings of the International Conference on Learning Representations (ICLR 2021)*. <https://doi.org/10.48550/arXiv.2009.01027>
41. Sukthanker, R. S., et al. (2024). Multi-objective Differentiable Neural Architecture Search. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

42. Bakhtiarifard, P., Igel, C., & Selvan, R. (2024). EC-NAS: Energy Consumption Aware Tabular Benchmarks for Neural Architecture Search. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
43. Hansen, N., & Ostermeier, A. (2001). Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2), 159–195. <https://doi.org/10.1162/106365601750190398>
44. Hansen, N., Müller, S. D., & Koumoutsakos, P. (2003). Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1), 1–18. <https://doi.org/10.1162/106365603321828970>
45. Auger, A., & Hansen, N. (2005). A Restart CMA Evolution Strategy With Increasing Population Size. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, vol. 2, pp. 1769–1776. <https://doi.org/10.1109/cec.2005.1554902>
46. Khouzani, F. F., Mirzaei, A., La Plante, P., & Gewali, L. (2025). CMA-ES with radial basis function surrogate for black-box optimization (CMA-SAO). In *Services 2025*. https://doi.org/10.1007/978-3-031-89063-5_31
47. Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2), 61–70. <https://doi.org/10.1016/j.swevo.2011.05.001>
48. Frankle, J., & Carbin, M. (2019). The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.1803.03635>
49. Lee, N., Ajanthan, T., & Torr, P. H. S. (2019). SNIP: Single-shot Network Pruning based on Connection Sensitivity. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.1810.02340>
50. Han, S., Mao, H., & Dally, W. J. (2016). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In

- Proceedings of the International Conference on Learning Representations (ICLR)*.
<https://doi.org/10.48550/arXiv.1510.00149>
51. Frankle, J., Dziugaite, G. K., Roy, D. M., & Carbin, M. (2020). Linear Mode Connectivity and the Lottery Ticket Hypothesis. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 3259–3269.
<https://doi.org/10.48550/arXiv.1912.05671>
 52. Liu, Z., Sun, M., Zhou, T., Huang, G., & Darrell, T. (2019). Rethinking the Value of Network Pruning. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.1810.05270>
 53. Blalock, D., Ortiz, J. J. G., Frankle, J., & Gutttag, J. (2020). What is the State of Neural Network Pruning? In *Proceedings of Machine Learning and Systems (MLSys)*, vol. 2, pp. 129–146. <https://doi.org/10.48550/arXiv.2003.03033>
 54. Molchanov, P., Mallya, A., Tyree, S., Frosio, I., & Kautz, J. (2019). Importance Estimation for Neural Network Pruning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11256–11264. DOI: <https://doi.org/10.1109/cvpr.2019.01152>
 55. He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., & Han, S. (2018). AMC: AutoML for Model Compression and Acceleration on Mobile Devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 11211, pp. 815–832. https://doi.org/10.1007/978-3-030-01234-2_48
 56. Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2023). GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.2210.17323>
 57. Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 30318–30332. <https://doi.org/10.48550/arXiv.2208.07339>

58. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., et al. (2024). AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*. <https://doi.org/10.48550/arXiv.2306.00978>
59. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., et al. (2023). LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint*, arXiv:2302.13971. <https://doi.org/10.48550/arXiv.2302.13971>
60. Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2018). Model Compression and Acceleration for Deep Neural Networks: The Principles, Progress, and Challenges. *IEEE Signal Processing Magazine*, 35(1), 126–136. <https://doi.org/10.1109/msp.2017.2765695>
61. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., et al. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2704–2713. <https://doi.org/10.1109/cvpr.2018.00286>
62. Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge Distillation: A Survey. *International Journal of Computer Vision*, 129(6), 1789–1819. <https://doi.org/10.1007/s11263-021-01453-z>
63. Sun, M., Liu, Z., Bair, A., & Kolter, J. Z. (2024). A Simple and Effective Pruning Approach for Large Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.2306.11695>
64. van der Ouderaa, T. F. A., Nagel, M., van Baalen, M., Asano, Y. M., & Blankevoort, T. (2024). The LLM Surgeon. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*. <https://openreview.net/forum?id=7P9l7u5SjW>
65. Kim, W., Kim, S., Park, M., & Jeon, G. (2020). Neuron merging: Compensating for pruned neurons. *Advances in Neural Information Processing Systems (NeurIPS)*, 33.

66. Goldberg, F., Lubarsky, Y., Gaissinski, A., Botchan, D., & Kisilev, P. (2022). Pruning Neural Nets by Optimal Neuron Merging. *Lecture Notes in Computer Science*, 688-699. DOI: https://doi.org/10.1007/978-3-031-09037-0_56
67. Li, G., Xu, Y., Li, Z., Liu, J., Yin, X., Li, D., & Barsoum, E. (2025). Týr-the-Pruner: Structural pruning LLMs via global sparsity distribution optimization. Preprint. <http://arxiv.org/abs/2503.09657v4>
68. Sieberling, O., Kuznedelev, D., Kurtic, E., & Alistarh, D. (2025). EvoPress: Accurate dynamic model compression via evolutionary search. In Proceedings of the 42nd International Conference on Machine Learning. <http://arxiv.org/abs/2410.14649v2>
69. Tang, S., Sieberling, O., Kurtic, E., Shen, Z., & Alistarh, D. (2025). DarwinLM: Evolutionary structured pruning of large language models. Preprint. <https://doi.org/10.70777/si.v2i3.15171>
70. Li, J., Chen, Y., Lian, Y., Zhou, H., Xu, J., Li, W., Pan, J., Wang, Y., Huang, S., Liu, J., Ding, L., & Dai, G. (2025). Large language model inference acceleration: A comprehensive hardware perspective. *Preprint*. <https://doi.org/10.48550/arXiv.2410.04466>
71. Ebrahimipour, S. M., Clark, J. J., Gross, W. J., & Meyer, B. H. (2026). HAP-E: Hessian-Aware Structured Pruning of LLMs for Efficient Inference. *Under Review (ICLR 2026)*. <https://openreview.net/forum?id=7P9l7u5SjH>
72. Sengupta, A., Chaudhary, S., & Chakraborty, T. (2025). You only prune once: Designing calibration-free model compression with policy learning (PruneNet). In *Proceedings of the International Conference on Learning Representations (ICLR 2025)*. <https://doi.org/10.48550/arXiv.2501.15296>
73. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281–305.
74. Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems (NeurIPS)*, 25, 2960–2968.

75. Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2623–2631). <https://doi.org/10.1145/3292500.3330701>
76. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research*, 18(185), 1-52.
77. Carstensen, T., Mallik, N., Hutter, F., & Rapp, M. (2025). Frozen layers: Memory-efficient many-fidelity hyperparameter optimization. *Preprint*. <https://doi.org/10.48550/arXiv.2504.10735>
78. Franceschi, L., Donini, M., Perrone, V., Klein, A., Archambeau, C., Seeger, M., Pontil, M., & Frasconi, P. (2024). Hyperparameter optimization in machine learning. *Foundations and Trends in Machine Learning*, 17(1–2). <https://doi.org/10.1561/22000000088>
79. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12), 2346–2363. <https://doi.org/10.1109/TKDE.2018.2876857>
80. Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., ... & Fernando, C. (2017). Population based training of neural networks. *arXiv preprint arXiv:1711.09846*. <https://doi.org/10.48550/arXiv.1711.09846>
81. Doulazmi, W., Lehuger, A., Toromanoff, M., Charraut, V., Buhet, T., & Moutarde, F. (2025). Multiple-Frequencies Population-Based Training. *arXiv.org*. DOI: <https://doi.org/10.48550/arXiv.2506.03225>
82. Li, Z., Cheng, J., & Gu, H. (2025). Sequential policy gradient for adaptive hyperparameter optimization (SPG). *Preprint*. <https://doi.org/10.48550/arXiv.2506.15051>

- 83.Theodorakopoulos, D., Stahl, F., & Lindauer, M. (2024). Hyperparameter Importance Analysis for Multi-Objective AutoML. *Frontiers in Artificial Intelligence and Applications*. DOI: <https://doi.org/10.3233/faia240602>
- 84.Qiu, H., Shen, Y., & He, X. (2026). Evolution strategies at scale: Towards practical ES-based optimization for deep learning. *Proceedings of the International Conference on Machine Learning (ICML)*. <https://doi.org/10.48550/arXiv.2509.24372>
- 85.Nazarians, A., & Kumar, S. (2025). GEGO: A hybrid golden eagle and genetic optimization algorithm for efficient hyperparameter tuning in resource-constrained environments. *Preprint*. <https://doi.org/10.48550/arXiv.2601.14672>
- 86.Custode, L. L., Yaman, A., Caraffini, F., & Iacca, G. (2024). An investigation on the use of large language models for hyperparameter tuning in evolutionary algorithms. In *GECCO '24 Companion*. <https://doi.org/10.1145/3638530.3664163>
- 87.Bansal, A., Stoll, D., Janowski, M., Zela, A., & Hutter, F. (2022). JAHS-Bench-201: A foundation for research on joint architecture and hyperparameter search. In *NeurIPS 2022, Track on Datasets and Benchmarks*.
- 88.Chen, Z., Leong, A. W. L., Ong, S. Y., Hemachandra, A., Lau, G. K. R., Foo, C.-S., Liu, Z., Chen, N. F., & Low, B. K. H. (2026). The chicken and egg dilemma: Co-optimizing data and model configurations for LLMs. In *ICLR 2026 Workshop DATA-FM*. <http://arxiv.org/abs/2602.08351v2>
- 89.Basu, S., Hutter, F., & Stoll, D. (2024). Multi-objective hyperparameter optimization in the age of deep learning (PriMO). *Preprint*. <https://doi.org/10.48550/arXiv.2511.08371>
- 90.Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., ... & Schmidt, L. (2022). Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *Proceedings of the International Conference on Machine Learning (ICML)*, 23965–23998.

91. Yu, L., Yu, B., Yu, H., Huang, F., & Li, Y. (2024). Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. In *Proceedings of the International Conference on Machine Learning (ICML)*.
92. Matena, M., & Raffel, C. (2022). Merging Models with Fisher-Weighted Averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 17703–17716. <https://doi.org/10.48550/arXiv.2111.09832>
93. Jin, X., Ren, X., Preoȃiuc-Pietro, D., & Cheng, P. (2023). Dataless Knowledge Fusion by Merging Weights of Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.2212.09849>
94. Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., et al. (2024). AdaMerging: Adaptive Model Merging for Multi-Task Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.2310.02575>
95. Tang, A., Shen, L., Luo, Y., Yin, N., Zhang, L., & Tao, D. (2024). Merging Multi-Task Models via Weight-Ensembling Mixture of Experts. In *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 235, pp. 47778–47799. <https://doi.org/10.48550/arXiv.2402.00433>
96. Verma, N., Murray, K., & Duh, K. (2025). DOTResize: Reducing LLM Width via Discrete Optimal Transport-based Neuron Merging. *arXiv.org*. DOI: <https://doi.org/10.48550/arXiv.2507.04517>
97. Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs. *Technical Report*, Wroclaw University of Technology. <https://doi.org/10.4236/am.2012.330189>
98. Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *Proceedings of the International Conference on Learning Representations (ICLR)*.
99. Zhuang, J., Tang, T., Ding, Y., Tatikonda, S., Dvornek, N., Papademetris, X., & Duncan, J. S. (2020). AdaBelief optimizer: Adapting stepsizes by the belief in

- observed gradients. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 18795–18806.
100. Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. (2018). Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.
 101. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations (ICLR)*.
 102. Tanabe, R. & Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. *2014 IEEE Congress on Evolutionary Computation (CEC)*, 1658-1665. DOI: <https://doi.org/10.1109/cec.2014.6900380>
 103. Liang, J., Qin, A., Suganthan, P., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3), 281-295. DOI: <https://doi.org/10.1109/tevc.2005.857610>
 104. Bradley, R. A. (1954). Rank Analysis of Incomplete Block Designs: II. Additional Tables for the Method of Paired Comparisons. *Biometrika*, 41(3/4), 502. DOI: <https://doi.org/10.2307/2332730>
 105. Zhao, T., Zhang, X. S., Zhu, W., Wang, J., Yang, S., Liu, J., & Cheng, J. (2021). Joint channel and weight pruning for model acceleration on mobile devices. *arXiv preprint arXiv:2110.08013*.
 106. Wang, C., Zhang, G., & Grosse, R. (2020). Picking winning tickets before training by preserving gradient flow. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.
 107. Dziugaite, G. K., Frankle, J., Ganguli, S., Larsen, B., & Paul, M. (2022). Lottery Tickets on a Data Diet: Finding Initializations with Sparse Trainable Networks. *Advances in Neural Information Processing Systems* 35, 18916-18928. DOI: <https://doi.org/10.52202/068431-1374>
 108. Mocanu, D. C., Mocanu, E., Stone, P., Nguyen, P. H., Gibescu, M., & Liotta, A. (2018). Scalable training of artificial neural networks with adaptive sparse

- connectivity inspired by network science. *Nature Communications*, 9(1). DOI: <https://doi.org/10.1038/s41467-018-04316-3>
109. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.
 110. Feurer, M., & Hutter, F. (2019). Hyperparameter Optimization. In *Automated Machine Learning: Methods, Systems, Challenges* (pp. 3–33). Springer. https://doi.org/10.1007/978-3-030-05318-5_1
 111. Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200), 675–701. <https://doi.org/10.1080/01621459.1937.10503522>
 112. Iman, R. L., & Davenport, J. M. (1980). Approximations of the critical region of the Friedman statistic. *Communications in Statistics – Theory and Methods*, 9(6), 571–595. <https://doi.org/10.1080/03610928008827904>
 113. Vargha, A., & Delaney, H. D. (2000). A Critique and Improvement of the CL Common Language Effect Size Statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics*, 25(2), 101–132. <https://doi.org/10.3102/10769986025002101>
 114. Abdelfattah, M. S., Mehrotra, A., Dudziak, Ł., & Lane, N. D. (2021). Zero-cost proxies for lightweight NAS. In Proceedings of the International Conference on Learning Representations (ICLR). <https://doi.org/10.48550/arXiv.2101.08134>
 115. Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-Tzur, J., Hardt, M., Recht, B., & Talwalkar, A. (2020). A System for Massively Parallel Hyperparameter Tuning (ASHA). In *Proceedings of Machine Learning and Systems (MLSys)*.
 116. Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential Model-Based Optimization for General Algorithm Configuration. *Lecture Notes in Computer Science*, 507-523. DOI: https://doi.org/10.1007/978-3-642-25566-3_40
 117. Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. DOI: <https://doi.org/10.1023/A:1010933404324>

118. Lu, N., Zhang, G., & Lu, J. (2014). Concept drift detection via competence models. *Artificial Intelligence*, 209, 11-28. DOI: <https://doi.org/10.1016/j.artint.2014.01.001>
119. Yeo, I. K., & Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4), 954–962. DOI: <https://doi.org/10.1093/biomet/87.4.954>
120. Awad, N., Mallik, N., & Hutter, F. (2021). DEHB: Evolutionary Hyperband for Scalable, Robust and Efficient Hyperparameter Optimization. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, 2147–2153. DOI: <https://doi.org/10.24963/ijcai.2021/296>
121. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>
122. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211–252. <https://doi.org/10.1007/s11263-015-0816-y>
123. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. <https://doi.org/10.1109/cvpr.2016.90>
124. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>

ДОДАТОК А. Програмна реалізація запропонованих методів

Запропоновані методи оптимізації параметрів і структури нейронних мереж реалізовано у вигляді модульної Python-бібліотеки, розміщеної у відкритому репозиторії https://github.com/GirBP/evolutionary_algorithms. Репозиторій побудовано на базі проведеного дослідження та містить повний вихідний код запропонованих методів, експериментальні конфігурації, обробку даних, а також звіти про відтворювані результати. Окрім основних методів, виокремлених у науковій новизні, репозиторій включає допоміжні модулі (інструменти попередньої обробки, утиліти візуалізації, ablation-конфігурації), що використовуються для додаткових перевірок та технологічного обслуговування експериментів. Код доступний для академічного й освітнього використання. Опубліковані результати — з обов'язковим посиланням на джерело відповідно до Наказу МОН України № 40 від 12.01.2017.

Таблиця А.1 подає відповідність між основними методами, описаними у дисертації, та їхньою реалізацією у структурі репозиторію. Для кожного методу зазначено папку експерименту та ключовий файл реалізації, що дозволяє читачеві однозначно визначити місце розташування коду.

А.1. Карта відповідності «метод — папка репозиторію»

№	Метод (з підрозділу дисертації)	Папка експерименту	Ключовий файл реалізації
1	TESA-26 — пошарово-еволюційне проріджування з ітеративним перерахунком значущості ваг (№2.3)	Ex08_TESA-26/	code/methods/tesa26.py
2	GFCS — конверсія розрідженої нейронної мережі у компактну щільну архітектуру без перенавчання (№2.4)	Ex09_GFCS/	ex09_lib/gfcs.py
3	SACMA-DAC — сурогатно-підсилений CMA-ES із	Ex_HPO_Final/	methods/sacma_v3.py

№	Метод (з підрозділу дисертації)	Папка експерименту	Ключовий файл реалізації
	Delta-F-контролем популяції (№ 3.3)		
4	SACMA-MAB — сурогатно-підсилений CMA-ES із ϵ -жадібним багаторуки́м бандитом (№ 3.4)	Ex_HPO_Final/	methods/sacma_mab.py
5	ENT — еволюційний відбір нейронної топології для злиття моделей із комплементарними знаннями (№ 4.2)	Ex30_HetMerge_ENT/	e23_ent.py
6	ENT-FT — калібрація зливої моделі логістичною регресією на замороженому екстракторі ознак (№ 4.3)	Ex30_HetMerge_ENT/	ent_ft_benchmark.py

Допоміжні модулі (метрика RCU, протокол валідації, ablation-конфігурації)— а також інструменти підготовки даних і профілювання) знаходяться у відповідних підпапках кожного експерименту і документовані у файлах README.md усередині дерева репозиторію.

A.2. Дані та чекпоінти моделей

Експерименти виконувались на публічних наборах даних: PD1 (HPO-bench), FCNet-tabular, YAHPO-Gym, FashionMNIST, MNIST, CIFAR-10, а також на синтетичних задачах класифікації (moons, circles, spirals, blobs, gaussian_quantiles, highdim, sequence_cls), що генеруються детерміновано з фіксованим зерном і повністю описані у відповідних конфігураційних файлах. Оригінальні набори даних завантажуються з офіційних джерел (TensorFlow Datasets, PyTorch torchvision, Kaggle API, HuggingFace Datasets) під час першого запуску скрипту підготовки. Чекпоінти попередньо тренованих моделей відтворюються автоматично скриптами prepare.py та phase1_train.py у відповідних папках експериментів, що забезпечує повну відтворюваність результатів без необхідності завантаження великих файлів.

ДОДАТОК Б. Акт впровадження результатів дисертаційної роботи у виробничу діяльність ТОВ «СД Профзв'язок»