

Дзюба Вадим Віталійович

аспірант кафедри технологій цифрового розвитку

Державний університет інформаційно-комунікаційних технологій, Україна, Київ

ORCID ID: 0009-0003-8715-7245

dziuba.vadym@gmail.com

Аронов Андрій Олексійович

кандидат технічних наук, доцент кафедри технологій цифрового розвитку

Державний університет інформаційно-комунікаційних технологій, Україна, Київ

ORCID ID: 0009-0000-7868-8341

a.aronov@duikt.edu.ua

ВІДТВОРЮВАНИЙ КАРКАС І СТАТИСТИЧНЕ ОЦІНЮВАННЯ НЕЙРОКЕРОВАНОГО КОНСТРУКТИВНОГО ПОШУКУ

Анотація. У роботах із нейрокерованої комбінаторної оптимізації якість навченого методу часто оцінюють за окремим вдалим запуском. За такого підходу складно відокремити властивості програмної реалізації від випадкових коливань результату. У статті описано програмний каркас, який дає змогу повторювати обчислювальні експерименти та перевіряти перевагу методу статистично. Як приклад використано конструктивне згортання білкових послідовностей у гідрофобно полярній моделі на квадратній та кубічній ґратках. Каркас об'єднує шість методів спільним інтерфейсом, розділяє обчислювальне ядро і веб-сервіс та задає локальні генератори випадкових чисел явним параметром *seed*. Для оцінювання застосовано 27 еталонних задач і по 20 запусків на пару задач і методу. Частку від відомого оптимуму і частку успішних запусків розраховано за однаковими правилами. Середні енергії двох варіантів старту алгоритму Монте Карло порівняно на десяти задачах у парному дизайні; для середніх наведено 95% перцентильні бутстреп інтервали, а для парних різниць застосовано двобічний критерій знакових рангів Вілкоксона. За даними експерименту, алгоритм Монте Карло отримав середню частку оптимуму 80,26%, жадібний метод 63,85%, випадковий вибір 30,46%, а автономний навчений гібрид 25,49%. Навчений початковий стан не дав статистично підтвердженого покращення Монте Карло на жодній із десяти задач: найменше значення *p* становило 0,164. Водночас обмеження конструктивного алгоритму забезпечували структурну коректність завершених конфомацій незалежно від якості політики. Цей результат показує, чому коректність розв'язку та його енергію слід аналізувати окремо. Практичний внесок статті становить узгоджений протокол порівняння, за яким негативний результат щодо навченого компонента можна повідомити без вибіркового добору запусків. Межі висновків визначають наявний набір задач, обчислювальні бюджети й обмежений обсяг абляційних експериментів; перенесення на інші класи задач потребує окремої перевірки.

Ключові слова: відтворюваність, конструктивний пошук, гідрофобно полярна модель, *Wave Function Collapse*, бутстреп, критерій Вілкоксона, Монте Карло.

Vadym Dziuba

PhD student, Department of Digital Development Technologies

State University of Information and Communication Technology, Kyiv, Ukraine

ORCID ID: 0009-0003-8715-7245

dziuba.vadym@gmail.com

Andrii Aronov

Candidate of Technical Sciences,

Associate Professor Department of Digital Development Technologies

State University of Information and Communication Technology, Kyiv, Ukraine

ORCID ID: 0009-0000-7868-8341

a.aronov@duikt.edu.ua

A REPRODUCIBLE FRAMEWORK AND STATISTICAL EVALUATION OF NEURAL-GUIDED CONSTRUCTIVE SEARCH

Abstract. Studies of neural combinatorial optimization sometimes assess a trained method using its best run. Such reporting does not distinguish a repeatable improvement from ordinary variation between random starts. This paper describes a software framework and an evaluation protocol designed to make computational experiments repeatable and comparisons statistically interpretable. The illustrative task is constructive folding of protein sequences in the hydrophobic polar model on square and cubic lattices. Six methods share one callable interface. The computational core is separated from an optional web service, and each randomized procedure receives an explicit seed for a local random number generator. The main experiment covers 27 benchmark instances with 20 runs for every instance and method. All methods are assessed using the same definition of the proportion of the known optimum and the success rate. On ten instances, Metropolis Monte Carlo initialized from a trained constructive policy is compared with Monte Carlo initialized from a random conformation using paired seeds and an equal iteration budget. Group means are accompanied by 95% percentile bootstrap intervals, while paired differences are tested with a two sided Wilcoxon signed rank

© 2026 Дзюба В.В., Аронов А.О. Цей матеріал ліцензовано за умовами CC BY 4.0.

<https://creativecommons.org/licenses/by/4.0/>

test. In the reported main experiment, Monte Carlo achieved a mean proportion of the optimum of 80.26%, greedy search 63.85%, random selection 30.46%, and the autonomous trained hybrid 25.49%. A trained initial state did not produce a statistically supported improvement over random initialization on any of the ten tested instances; the smallest p value was 0.164. Constructive constraints still preserved the structural validity of completed conformations regardless of policy quality. This distinction between feasible output and solution quality is central to interpreting the result. The practical contribution is a consistent protocol that permits a negative finding about the trained component to be reported without selecting favorable runs. The conclusions are limited to the supplied benchmark set and computational budgets. Ablation results use fewer seeds and should be treated as exploratory; application to other problem classes requires further validation.

Keywords: reproducibility, constructive search, hydrophobic polar model, Wave Function Collapse, bootstrap, Wilcoxon signed rank test, Monte Carlo.

1. Вступ

Навчена евристика може змінювати порядок вибору ходів у дискретному пошуку. Її оцінювання потребує двох перевірок: чи повторюються результати за відомих умов і чи зберігається перевага за різних випадкових запусків. Огляд відтворюваності досліджень машинного навчання вказує на неповний опис налаштувань, вибіркоче звітування та залежність від середовища виконання [1]. Для комбінаторної оптимізації додається ризик нерівного порівняння з класичними методами [2].

У статті описано архітектуру каркаса та спосіб статистичного оцінювання на прикладі згортання білкових послідовностей у гідрофобно полярній моделі. Приклад важливий тим, що навчена політика відповідає за якість вибору ходу, а конструктивні обмеження відповідають за допустимість завершеної конформації. Отже, ці властивості можна перевірити окремо.

2. Постановка проблеми

Для послідовності з гідрофобних Н і полярних Р залишків потрібно побудувати самоунікний шлях на двовимірній або тривимірній ґратці. Якість конформації визначається енергією незв'язаних уздовж ланцюга контактів Н–Н: менше значення енергії є кращим. Конструктивний алгоритм обирає напрям на кожному кроці; перевірка обмежень вилучає зайняті та недопустимі позиції. Навчена політика може змінити ймовірності допустимих ходів, проте сама її наявність не гарантує меншої енергії.

Потрібен протокол, у якому методи використовують однакові задачі, визначення метрик і набори початкових значень генератора. Окремо слід перевірити, чи поліпшує навчений початковий стан локальний пошук. Показник найкращого запуску для цього недостатній: він відображає максимум серед проб і може не характеризувати типовий результат.

3. Аналіз останніх досліджень і публікацій

Семмельрок та співавтори систематизують перешкоди відтворенню експериментів машинного навчання [1]. Гармендіа, Себеріо й Мендібуру аналізують застосовність нейронної комбінаторної оптимізації та наголошують на зіставленні з належними класичними базовими методами [2]. Карл та співавтори аргументують значення коректно описаних негативних результатів [3]. Робота Лю та співавторів розглядає допустимість розв'язків у навчній комбінаторній оптимізації [4]. Разом ці джерела обґрунтовують потребу перевіряти відтворюваність, якість і допустимість за різними критеріями.

Попереднє дослідження згортання білків методами машинного навчання [5] і дисертація про поєднання Wave Function Collapse з нейронною мережею [6] задають прикладну основу цієї роботи. Еталонні задачі гідрофобно полярної моделі описано в [7], [8]. Конструктивне поширення обмежень споріднене з алгоритмом Wave Function Collapse [12], його аналізом як задачі задоволення обмежень [13] та класичним поняттям узгодженості обмежень [14]. Підходи до стохастичної політики й локального пошуку спираються відповідно на [15], [16] і [17].

Невирішеною для цього прикладного випадку залишається узгоджена оцінка: одна програма повинна забезпечувати повторювані запуски різних методів, а звіт має відокремлювати найкращий результат від оцінки середньої ефективності й статистичного висновку. Статистичну основу становлять критерій Вілкоксона [9], поправка на зв'язані ранги [10] та бутстреп [11].

4. Мета і задачі дослідження

Мета дослідження полягає у створенні відтворюваного програмного каркаса та протоколу оцінювання методів конструктивного пошуку з навченою політикою. Для її досягнення визначено чотири задачі: уніфікувати виклик різних алгоритмів; зробити випадковість явною та перевірюваною; встановити спільні метрики й парну схему порівняння; застосувати протокол до еталонних задач згортання та описати як позитивні, так і негативні результати.

5. Результати дослідження

Архітектура каркаса. Пакет experiments/ реалізовано мовою Python із numpy як залежністю обчислювального ядра. Геометрія ґратки, перевірка обмежень, обчислення енергії, нейронна мережа та статистичні процедури відокремлені від командного інтерфейсу й опційного вебсервісу. У таблиці 1 деталізовано призначення основних шарів програмного каркасу, а на рисунку 1 показано основні шари програмного каркасу.

Таблиця 1

Структура пакета	
Модуль	Функція
wfc_fold/lattice.py, energy.py, constraints.py	Геометрія ґратки, енергія й перевірка самоунікності
wfc_fold/features.py, nn.py, policy_value.py	Ознаки стану та мережі політики і значення на NumPy
wfc_fold/wfc_engine.py, policy_folder.py	Конструктивний пошук із відкатом та варіант із навченою політикою
wfc_fold/expert_data.py, train_imitation.py, train_reinforce.py	Підготовка даних, імітаційне навчання та REINFORCE
wfc_fold/api.py; baselines/; hybrid/	Програмний фасад, базові методи й гібридний метод
eval/; experiment.py; warmstart.py	Метрики, пакетні запуски та парне порівняння
scripts/verify.py; service/	Перевірка детермінізму та опційний вебсервіс



Рис. 1. Шари програмного каркаса

Шість методів, random, greedy, mc, wfc, hybrid і hybrid_rl, реалізують спільний інтерфейс method(seq, dim, seed) → FoldResult. Оцінювальний код викликає їх однаково та отримує координати, енергію й ознаку завершення. Окрема група залежностей api містить FastAPI, uvicorn та httpx; пакетне оцінювання не потребує її встановлення. Програмний фасад HPFolder дає доступ до навчання, завантаження ваг і пошуку конформації. Вебінтерфейс додатково перевіряє вхідну послідовність, розмірність ґратки та повертає помилку HTTP 422 за некоректного запиту.

Відтворюваність обчислень. Ініціалізація ваг, вибір ходів, порядок навчальних прикладів, пропозиції локального пошуку та бутстреп вибірки використовують локальні генератори numpy.random.default_rng. Випадковий стан передається через seed, тому запуск не залежить від глобального стану numpy.random. Ваги мереж зберігаються як масиви numpy. Службовий скрипт scripts/verify.py повторює запуск за тим самим seed і перевіряє точну рівність координат та енергії; невідповідність дає ненульовий код завершення.

Аналітичний градієнт мережі перевіряють центральною скінченною різницею для координати x_k з кроком $\varepsilon = 10^{-6}$:

$$\frac{\partial f}{\partial x_k} \approx \frac{f(x + \varepsilon e_k) - f(x - \varepsilon e_k)}{2\varepsilon} \quad (1)$$

Допуск числової звірки становить 10^{-4} . Для обмеження великих кроків навчання застосовують кліпінг градієнта g за глобальною нормою з порогом $\tau = 10$:

$$\text{clip}(g) = \{g, \quad \|g\|_2 \leq \tau, \tau g / \|g\|_2, \quad \|g\|_2 > \tau\} \quad (2)$$

Операція (2) обмежує величину градієнта під час імітаційного навчання та REINFORCE. Вона сама по собі не підтверджує якість навченої політики: для такого висновку потрібне окреме оцінювання енергії.

Метрики та статистичний протокол. Набір складається з 27 задач: 12 тривимірних послідовностей s1–s12 довжиною 20–88 залишків, десять тривимірних задач A1–A10 із 27 залишками та п'ять двовимірних задач [7], [8]. Для кожної пари задачі й методу використовують seed від 0 до 19. Метод *mc* виконує 2000 ітерацій на запуск. У таблиці 2 наведені показники оцінювання.

Таблиця 2

Показники оцінювання	
Показник	Означення
E_{best}	Найменша енергія серед завершених запусків серії
pct	Частка від відомого оптимуму E^* для найкращої енергії, %
sr	Частка запусків, що завершилися з енергією не вищою за E^*
ΔE	Різниця середньої енергії випадкового та навченого старту

Для порівняння якості розв'язків використовується частка від відомого оптимуму. Оскільки в НР-моделі оптимальна та знайдена енергії є від'ємними, їх відношення залишається додатним. Для найкращої енергії E_{best} , отриманої в серії запусків, показник обчислюється як:

$$pct = 100 \cdot \frac{E_{best}}{E^*} \quad (3)$$

Значення $pct = 100\%$ відповідає досягненню відомого оптимуму, а менші значення характеризують ступінь відхилення найкращого знайденого розв'язку від нього. Якщо жоден запуск не завершився побудовою допустимої конформації, для агрегованого показника приймається нульове значення.

Окремо враховується не лише найкращий результат серії, а й стабільність його досягнення. Частка завершених запусків, у яких досягнуто відомого оптимуму:

$$sr = \frac{1}{N} \sum_{r=1}^N 1 \left[complete_r \wedge E_r \leq E^* \right] \quad (4)$$

Таким чином, $sr = 1$ означає досягнення критерію успіху в усіх запусках, тоді як $sr = 0$ вказує, що його не було досягнуто в жодному з них.

Для парного порівняння Монте Карло з випадковим і навченим початковими станами додатково використовується різниця середніх енергій:

$$\Delta E = \bar{E}_{random} - \bar{E}_{warm} \quad (5)$$

За прийнятим означенням додатне значення ΔE відповідає нижчій, тобто кращій, середній енергії для навченого старту. Від'ємне значення вказує на перевагу випадкового старту, а значення, близьке до нуля – на малу різницю між середніми. Показник ΔE використовується як описова характеристика величини зміни, тоді як статистичний висновок про парні відмінності робиться окремо за критерієм Вілкоксона.

Додатне ΔE означає меншу, тобто кращу, середню енергію для навченого старту. Порівняння стартових станів охоплює s1–s10: для кожної задачі навчений генератор готують один раз, а потім запускають два варіанти Монте Карло з однаковими 20 seed і бюджетом 1000 ітерацій. Якщо генератор не завершує побудову стартової конформації, у журналі запуску фіксують перехід до випадкового старту.

Для кожної групи середню енергію супроводжують 95 % перцентильним бутстреп інтервалом за 10 000 вибірок із поверненням [11]. Ці інтервали характеризують окремі групові середні, а не довірчий інтервал парної різниці. Парну гіпотезу перевіряють двобічним критерієм знакових рангів Вілкоксона [9]. Після вилучення нульових різниць ранжують модулі решти різниць та обчислюють:

$$W = \min(W_+, W_-) \quad (6)$$

Тут W_+ і W_- є сумами рангів додатних і від'ємних парних різниць. Для p значення використовують нормальну апроксимацію з поправкою на зв'язані ранги [10]:

$$\mu = \frac{n(n+1)}{4}, \quad \sigma^2 = \frac{n(n+1)(2n+1) - \frac{1}{2} \sum_j (t_j^3 - t_j)}{24} \quad (7)$$

Величина t_j позначає розмір групи рівних за модулем різниць; далі застосовують поправку на неперервність і двобічне p значення через math.erfc . Бутстреп і тест реалізовано на numpy без SciPy. Інтерпретація значущості ґрунтується на порозі $\alpha = 0,05$. За такого дизайну непідтверджений ефект не слід тлумачити як доведену еквівалентність методів.

Порівняння методів. У таблиці 3 підсумовано результати основного експерименту на 27 задачах. Значення частки оптимуму усереднено між задачами, а оцінка найкращого запуску не замінює середню енергію типового запуску.

Таблиця 3

Результати роботи методів

Метод	Середня частка оптимуму, %	Кількість задач із досягнутим точним оптимумом	Спостереження
random	30,46	0	Випадковий конструктивний вибір
greedy	63,85	0	Вищий показник, ніж у mc лише на s10
mc	80,26	4	2000 ітерацій локального пошуку
wfc	30,20	0	На 22 тривимірних задачах збігається з random
hybrid	25,49	0	Не перевершує greedy і mc на жодній задачі
hybrid_rl	19,12	0	На чотирьох задачах енергія дорівнює нулю

У таблиці 3 нуль означає відсутність точного досягнення E і не означає нульову енергію або відсутність завершених розв'язків.

Найкращий середній показник отримав mc, 80,26% від відомого оптимуму. Порівняння слід читати з урахуванням бюджету: його 2000 ітерацій значно перевищують один конструктивний прохід інших методів. Тому таблиця встановлює результат за заданими бюджетами, але сама по собі не доводить, що mc кращий за однакового часу обчислень. Автономний навчений гібрид досяг 25,49%, а після REINFORCE лише 19,12%. На s5, s8, s12 і A10 останній метод давав енергію нуль за завершених запусків. Це вказує на погану якість конформацій, а не на порушення структурних обмежень.

Для перевірки користі навченого старту таблиця 4 подає середні енергії та результати парного тесту. W/T/L означає кількість вигравів, нічий і програвів навченого старту серед 20 пар.

Парне порівняння Монте Карло за випадкового та навченого старту

Задача	Випадковий старт: середнє [95 % ДІ]	Навчений старт: середнє [95 % ДІ]	ΔE	W/T/L	W	p
s1	-7,35 [-7,65; -7,00]	-7,60 [-8,10; -7,20]	+0,25	6/10/4	18,0	0,331
s2	-7,20 [-7,65; -6,75]	-7,05 [-7,50; -6,65]	-0,15	3/12/5	13,0	0,516
s3	-5,15 [-5,55; -4,85]	-4,95 [-5,25; -4,65]	-0,20	2/14/4	7,5	0,590
s4	-10,55 [-11,25; -9,90]	-10,40 [-11,10; -9,75]	-0,15	7/3/10	63,0	0,525
s5	-19,20 [-19,85; -18,55]	-20,05 [-21,00; -19,10]	+0,85	11/3/6	47,0	0,164
s6	-16,50 [-17,15; -15,90]	-16,10 [-17,05; -15,15]	-0,40	8/3/9	64,0	0,566
s7	-15,35 [-16,30; -14,50]	-15,50 [-16,40; -14,65]	+0,15	11/2/7	72,5	0,582
s8	-21,20 [-22,20; -20,20]	-20,85 [-21,85; -19,80]	-0,35	5/5/10	48,5	0,530
s9	-32,15 [-33,60; -30,75]	-33,85 [-35,10; -32,60]	+1,70	13/1/6	61,5	0,183
s10	-29,95 [-31,65; -28,40]	-29,65 [-31,00; -28,35]	-0,30	12/1/7	91,5	0,904

Найменше p значення дорівнює 0,164; отже, жоден із десяти тестів не перетинає поріг 0,05. Знак ΔE додатний для чотирьох задач і від'ємний для шести. Наприклад, на s1 найкращий із 20 запусків із навченим стартом досяг відомого оптимуму -11, проте середня енергія змінилася лише з -7,35 до -7,60, а p = 0,331. Цей приклад показує, як один найкращий запуск може перебільшувати враження про системну перевагу.

Абляції та обмеження інтерпретації. Додаткова серія абляцій досліджує розмір прихованого шару 16, 32 і 64 нейрони, 0, 100 і 300 епізодів REINFORCE та бюджет локального пошуку 500, 1000 і 2000 ітерацій. Для s1 і s4 використано лише по п'ять seed на конфігурацію. Рисунок 2 показує напрям змін середньої енергії, але такого обсягу недостатньо для сильного статистичного висновку.

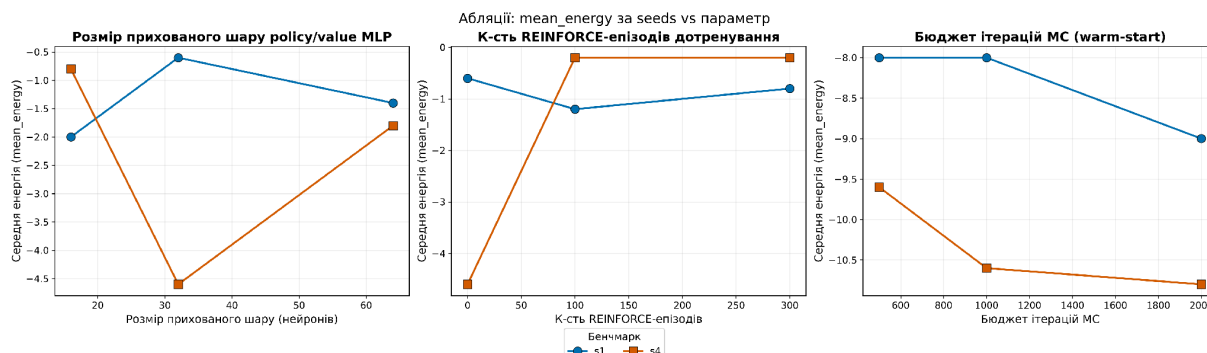


Рис. 2. Абляції розміру мережі, числа епізодів REINFORCE та бюджету Монте Карло

Оптимальний розмір прихованого шару в цих двох задачах відрізняється. На s4 варіант без донавчання REINFORCE дає кращу середню енергію за варіанти зі 100 і 300 епізодами. Збільшення числа ітерацій Монте Карло в обох задачах не погіршує середню енергію, але приріст після 1000 ітерацій менший. Ці спостереження слід перевіряти на ширшій вибірці, перш ніж робити загальні висновки про гіперпараметри.

Протокол оцінює конкретну реалізацію методів за визначених у роботі обчислювальних бюджетів і не претендує на узагальнення щодо всіх нейрокерованих підходів до конструктивного пошуку. Експериментальний набір обмежений послідовностями довжиною до 88 залишків, а абляційне дослідження проведено лише на двох задачах і з п'ятьма початковими значеннями генератора для кожної конфігурації. Тому результати абляцій слід розглядати як попередні. Крім того, основне порівняння виконано за фіксованими алгоритмічними бюджетами, а не за вирівняним часом виконання або кількістю елементарних обчислень. Перевірка стійкості висновків за іншими бюджетами та на ширшому наборі задач залишається напрямом подальших досліджень.

6. Висновки та перспективи подальших досліджень

Побудований каркас задає спільний виклик шести методів, явні локальні джерела випадковості й однакові правила оцінювання. На 27 задачах гідрофобно полярної моделі за встановленими бюджетами Монте Карло досяг середньої частки оптимуму 80,26 %, тоді як автономний навчений гібрид отримав 25,49 %. На десяти

задачах навчений старт не продемонстрував статистично підтвердженої переваги за парним тестом. Водночас конструктивні обмеження зберігають допустимість завершених конформацій незалежно від енергетичної якості.

Подальші дослідження передбачають випробування протоколу на інших дискретних задачах, розширення набору еталонних послідовностей та порівняння за вирівняним часом і кількістю обчислень. Доцільно також перевірити селективне застосування навченого старту, яке обирає задачі за попередньо визначеним правилом, а ефект оцінює на окремій тестовій вибірці.

Внесок авторів

Дзюба В.В. здійснив концептуалізацію дослідження, взяв участь у формуванні методології дослідження, провів обчислювальні експерименти, виконав формальний аналіз результатів, підготував дані та візуалізації. Аронов А.О. виконав розроблення методології та програмного забезпечення, провів аналіз та інтерпретацію результатів. Обидва автори ознайомилися з остаточною версією рукопису та схвалили її до публікації.

Декларація про штучний інтелект

Під час підготовки рукопису засоби штучного інтелекту використовувалися для мовного редагування, покращення стилістичної зв'язності тексту та підготовки англійської анотації. Наукова постановка задачі, програмна реалізація, експериментальні дані, статистичний аналіз, інтерпретація результатів і остаточна відповідальність за зміст належать авторам.

Конфлікт інтересів

Автори заявляють про відсутність конфлікту інтересів та підтверджують, що під час підготовки цієї роботи не існувало жодних комерційних, фінансових чи інших взаємовідносин, які могли б бути розцінені як такі, що здатні вплинути на результати дослідження або їх інтерпретацію. Робота виконана відповідно до принципів академічної доброчесності, етичних норм проведення наукових досліджень та вимог редакційної політики щодо запобігання конфлікту інтересів.

Список використаної літератури

1. Semmelrock, H., Ross-Hellauer, T., Kopeinik, S., Theiler, D., Haberl, A., Thalmann, S., & Kowald, D. (2025). Reproducibility in machine-learning-based research: Overview, barriers, and drivers. *AI Magazine*, 46(2), e70002. <https://doi.org/10.1002/aaai.70002>
2. Garmendia, A. I., Ceberio, J., & Mendiburu, A. (2024). Applicability of neural combinatorial optimization: A critical view. *ACM Transactions on Evolutionary Learning and Optimization*, 4(3), Article 3647644. <https://doi.org/10.1145/3647644>
3. Karl, F., Kemeter, L. M., Dax, G., & Sierak, P. (2024). Position: Embracing negative results in machine learning. *arXiv*. <https://arxiv.org/abs/2406.03980>
4. Liu, Y., Zhou, C., Chen, Y., Zhang, S., Lin, X., & Wang, X. (2026). Hard constraints meet soft generation: Guaranteed feasibility for LLM-based combinatorial optimization. *arXiv*. <https://arxiv.org/abs/2602.01090>
5. Дзюба, В. В., Колодюк, А. В., Олейніков, І. А., & Бугайов, Д. М. (2025). Моделювання згортання білків з використанням методів машинного навчання. *Зв'язок*, 3(175), 101–107. <https://doi.org/10.31673/2412-9070.2025.029944>
6. Дзюба, В. В. (2026). Метод прогнозування згортання білків на основі гібридної моделі WFC і нейронних мереж [Дисертація доктора філософії, ДУИКТ].
7. Boumedine, N., & Bouroubi, S. (2021). Protein folding simulations in the hydrophobic-polar model using a hybrid cuckoo search algorithm. *arXiv*. <https://arxiv.org/abs/2105.13226>
8. Khandoker, S. A., Inack, E. M., & Hibat-Allah, M. (2025). Lattice protein folding with variational annealing. *arXiv*. <https://arxiv.org/abs/2502.20632>
9. Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80–83.
10. Cureton, E. E. (1967). The normal approximation to the signed-rank sampling distribution when zero differences are present. *Journal of the American Statistical Association*, 62(319), 1068–1069. <https://doi.org/10.1080/01621459.1967.10500917>
11. Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1–26.
12. Gumin, M. (2016). WaveFunctionCollapse [Computer software]. GitHub. <https://github.com/mxgmn/WaveFunctionCollapse>
13. Karth, I., & Smith, A. M. (2017). WaveFunctionCollapse is constraint solving in the wild. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*. <https://doi.org/10.1145/3102071.3110566>
14. Mackworth, A. K. (1977). Consistency in networks of relations. *Artificial Intelligence*, 8(1), 99–118.
15. Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8, 229–256. <https://doi.org/10.1007/BF00992696>
16. Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.
17. Madras, N., & Sokal, A. D. (1988). The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50, 109–186. <https://doi.org/10.1007/BF01022990>

References

1. Semmelrock, H., Ross-Hellauer, T., Kopeinik, S., Theiler, D., Haberl, A., Thalmann, S., & Kowald, D. (2025). Reproducibility in machine-learning-based research: Overview, barriers, and drivers. *AI Magazine*, 46(2), e70002. <https://doi.org/10.1002/aaai.70002>

2. Garmendia, A. I., Ceberio, J., & Mendiburu, A. (2024). Applicability of neural combinatorial optimization: A critical view. *ACM Transactions on Evolutionary Learning and Optimization*, 4(3), Article 3647644. <https://doi.org/10.1145/3647644>
3. Karl, F., Kemeter, L. M., Dax, G., & Sierak, P. (2024). Position: Embracing negative results in machine learning. *arXiv*. <https://arxiv.org/abs/2406.03980>
4. Liu, Y., Zhou, C., Chen, Y., Zhang, S., Lin, X., & Wang, X. (2026). Hard constraints meet soft generation: Guaranteed feasibility for LLM-based combinatorial optimization. *arXiv*. <https://arxiv.org/abs/2602.01090>
5. Dziuba, V. V., Kolodiuk, A. V., Oleinikov, I. A., & Buhaiov, D. M. (2025). Modeling protein folding using machine learning methods [in Ukrainian]. *Zviazok*, 3(175), 101–107. <https://doi.org/10.31673/2412-9070.2025.029944>
6. Dziuba, V. V. (2026). A protein folding prediction method based on a hybrid WFC model and neural networks [Doctoral dissertation, State University of Information and Communication Technology; in Ukrainian].
7. Boumedine, N., & Bouroubi, S. (2021). Protein folding simulations in the hydrophobic-polar model using a hybrid cuckoo search algorithm. *arXiv*. <https://arxiv.org/abs/2105.13226>
8. Khandoker, S. A., Inack, E. M., & Hibat-Allah, M. (2025). Lattice protein folding with variational annealing. *arXiv*. <https://arxiv.org/abs/2502.20632>
9. Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80–83.
10. Cureton, E. E. (1967). The normal approximation to the signed-rank sampling distribution when zero differences are present. *Journal of the American Statistical Association*, 62(319), 1068–1069. <https://doi.org/10.1080/01621459.1967.10500917>
11. Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1–26.
12. Gumin, M. (2016). WaveFunctionCollapse [Computer software]. GitHub. <https://github.com/mxgmn/WaveFunctionCollapse>
13. Karth, I., & Smith, A. M. (2017). WaveFunctionCollapse is constraint solving in the wild. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*. <https://doi.org/10.1145/3102071.3110566>
14. Mackworth, A. K. (1977). Consistency in networks of relations. *Artificial Intelligence*, 8(1), 99–118.
15. Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8, 229–256. <https://doi.org/10.1007/BF00992696>
16. Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.
17. Madras, N., & Sokal, A. D. (1988). The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50, 109–186. <https://doi.org/10.1007/BF01022990>

Надійшла до редакції: 05.07.2026

Прийнята до друку: 15.09.2026

Опубліковано: 30.09.2026