https://doi.org/10.31649/1997-9266-2025-182-5-129-139

УДК 004.054:[004.032.26+004.85]

Л. Р. Кулик[1]
О. Б. Мокін[1]

МАСШТАБУВАННЯ ПРОГНОЗУВАННЯ ВІДЕО ЗА ДОПОМОГОЮ ПРОСТОРОВО-ЧАСОВИХ ПАТЧІВ

[1]Вінницький національний технічний університет

Запропоновано нову архітектуру для обробки відеоданих, Vision Byte Latent Transformer (V-BLT), яка адаптує принципи успішних байт-рівневих мовних моделей до зорової модальності. На відміну від стандартних підходів, що використовують пакування фіксованого розміру (patching), які є обчислювально неефективними через рівномірний розподіл ресурсів незалежно від складності візуального контенту, V-BLT працює безпосередньо з потоком байтів відео. Це дозволяє уникнути втрати інформації, пов'язаної з попередньою токенізацією, та підвищити гнучкість обробки. Ключовими внесками роботи є розробка концепції просторово-часових латентних патчів, впровадження N-вимірних ротаційних позиційних вкладень для збереження когерентності даних у розгорнутому потоці байтів, та застосування багаторівневої трансформерної архітектури для ієрархічної обробки даних. Для валідації гіпотези та тестування моделі розроблено новий синтетичний набір даних з 2D та 3D фігурами, що обертаються, який дозволяє проводити контрольовану оцінку здатності моделі до просторово-часового мислення. Експериментально продемонстровано, що V-BLT ефективно прогнозує майбутні кадри, досягаючи високих показників за метриками MSE, SSIM та PSNR в порівнянні з ViViT та UNet3D, при цьому демонструючи вищу ефективність розрахунків. Розроблена архітектура згідно з дизайном має можливість генерувати піксельні карти ентропії, які візуалізують невизначеність прогнозу та корелюють з динамічно складними регіонами сцени. Це відкриває шлях до реалізації динамічного, залежного від контенту, розподілу обчислювальних ресурсів «на ходу», що є перспективним напрямком для створення ефективніших та масштабованих фундаментних моделей для відеоаналітики.

Ключові слова: машинне навчання, нейронні мережі, обробка природної мови, трансформери, комп'ютерний зір, згорткова нейронна мережа, варіаційний автоенкодер, синтетичні дані, оптимізація, штучні нейронні мережі.

Вступ

Розробка та застосування моделей глибокого навчання для аналізу відео є однією з ключових задач сучасного штучного інтелекту, що охоплює широкий спектр застосувань, від систем комп'ютерного зору в автономних транспортних засобах до автоматизованого аналізу медіаконтенту. Проте однією з головних перешкод на шляху створення ефективних відеомоделей є величезний обсяг даних, що міститься в необроблених відеопослідовностях. Стандартний підхід, популяризований архітектурою Vision Transformer (ViT) [2] та її адаптаціями для відео, такими як Video Vision Transformer (ViViT) [1] та Video Swin Transformer [3], полягає у розбитті відео на сітку непересічних просторово-часових «кубів» або патчів. Хоча цей підхід довів свою ефективність, він має фундаментальне обмеження: він є формою прямої та негнучкої «токенізації» візуальних даних. Усі регіони відео, незалежно від їхнього змісту, чи то статичний, низькоінформативний фон, чи динамічна сцена з рухомими об'єктами, обробляються з однаковими обчислювальними витратами. Така стратегія є неефективною, оскільки значна частина обчислювальних ресурсів витрачається на обробку надлишкової або легко передбачуваної інформації.

Аналогічна проблема неефективної токенізації тривалий час існувала в галузі обробки природної мови (NLP), де традиційні моделі, що основувалися на фіксовані токени (слова або їхні частини), стикалися з труднощами у разі обробки рідкісних слів, складних морфологічних структур та багатомовних текстів. Проривним рішенням у цій сфері стала архітектура Byte Latent Transformer (BLT) [4], яка продемонструвала можливість ефективного навчання безпосередньо на послідовно-

стях байтів. Ключова ідея BLT полягає в динамічному групуванні байтів у «латентні патчі», розмір яких залежить від інформаційної щільності, що вимірюється через ентропію. Прості та легко передбачувані послідовності, такі як пробіли чи поширені слова та вирази, об'єднуються у великі патчі, що вимагають менших обчислювальних затрат. Натомість складні та непередбачувані фрагменти тексту формують малі патчі, на які модель концентрує більше уваги та обчислювальної потужності. Цей принцип адаптивного розподілу ресурсів дозволив досягти високої продуктивності та ефективності, що підтверджується також іншими байт-рівневими моделями, наприклад, ByT5 [5].

У цій роботі висувається гіпотеза, що принципи, закладені в архітектурі BLT, можуть бути успішно адаптовані для обробки відеоданих, пропонуючи нове рішення проблеми обчислювальної надлишковості. На відміну від одновимірного потоку байтів тексту, пропонується працювати з тривимірною сіткою байтів відео (час, висота, ширина). Для цього розроблено нову архітектуру — Vision Byte Latent Transformer (V-BLT). Основна ідея полягає у відмові від фіксованої сітки патчів на користь *навчених просторово-часових патчів*. Модель навчається самостійно групувати байти відео в динамічні об'єми (кубоїди), розмір яких залежить від візуальної складності та динаміки сцени, що дозволяє значно підвищити ефективність обробки відеопотоків.

Аналіз наявних підходів до обробки даних у часі

Для розуміння контексту та новизни запропонованої архітектури V-BLT, необхідно розглянути ключові технології, що є її ідейними попередниками, а також домінувальні на сьогодні підходи в галузі обробки часових даних. Ці технології можна умовно поділити на три основні напрямки: байт-рівневі мовні моделі, трансформерні архітектури для відео та сучасні генеративні відеомоделі.

Архітектура Byte Latent Transformer [4] запропонована як ефективна альтернатива традиційним мовним моделям, що ґрунтуються на фіксовані токенізатори. Основна інновація BLT полягає у введенні динамічного, залежного від даних, методу групування байтів у «латентні патчі». Цей процес керується ентропією: модель самостійно визначає, які послідовності байтів є інформаційно насиченими, а які — надлишковими. Структурно модель складається з трьох ключових компонентів, як показано на рис. 1.

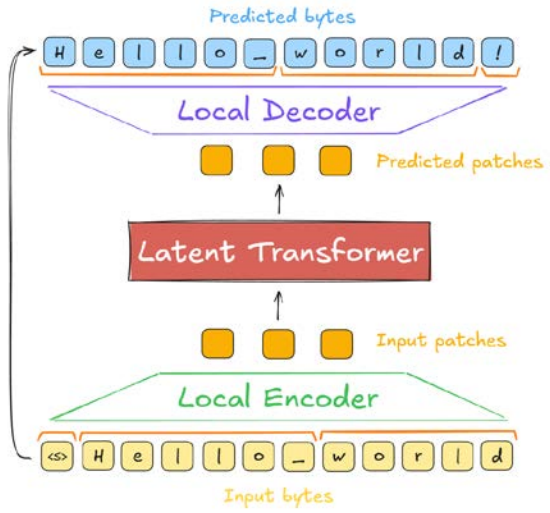

Рис. 1. Ілюстрація концептуального принципу роботи BLT

1. Локальний кодер (Local Encoder) — легкий трансформер, що обробляє послідовні байтові входи та агрегує їх у початкові представлення патчів.

2. Глобальний латентний трансформер (Global Latent Transformer) — обчислювально інтенсивна модель, що працює на рівні латентних патчів. Вона виконує основне завдання прогнозування наступних патчів в послідовності.

3. Локальний декодер (Local Decoder) — декодує вихідні дані глобального трансформера назад у послідовність байтів.

Вхідний потік байтів тексту (наприклад, "Hello World!") обробляється локальним кодером, який динамічно групує байти в латентні патчі на основі їхньої ентропії. Глобальний трансформер обробляє ці патчі, після чого локальний декодер відновлює прогнозовану послідовність байтів. Така ієрархічна структура дозволяє ефективно розподіляти обчислювальні ресурси, застосовуючи потужний трансформер лише до стиснених, інформаційно значущих одиниць (патчів), а не до кожного окремого байта. Такий підхід є ключовим для досягнення обчислювальної ефективності у разі роботи з великими обсягами даних, що забезпечує значне прискорення як під час навчання, так і під час інференсу, зберігаючи при цьому доступ до повної інформації на рівні байтів.

Щодо архітектури трансформерів, то їхня епоха у комп'ютерному зорі розпочалася з моделі ViT, яка продемонструвала, що зображення можна ефективно обробляти, розбивши їх на сітку фіксованих патчів та подавши отриману послідовність на вхід стандартного трансформера. Цей підхід успішно розширений для обробки відео. Моделі, такі як ViViT та Video Swin Transformer, адаптували цю парадигму для просторово-часових даних. Зазвичай відеопотік розбивається на

послідовність непересічних тривимірних «кубів» (патчів), які потім лінеаризуються та подаються на вхід трансформера. Механізм уваги (self-attention) в таких моделях може бути реалізований порізному: від спільної просторово-часової уваги до розділеної, де увага спочатку застосовується в просторі (всередині кадру), а потім у часі (між кадрами). Приклад такого підходу показано на рис. 2.

Вхідний відеопотік розбивається на сітку тривимірних патчів. Ці патчі лінеаризуються, до них додаються позиційні вкладення, після чого отримана послідовність векторів обробляється трансформер-енкодером для розв'язання задачі, наприклад, класифікації дії.

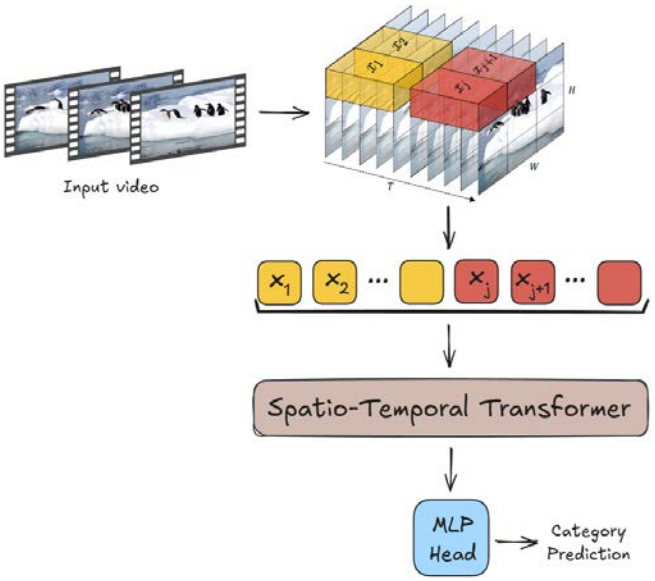

Рис. 2. Схема роботи відеомоделі ViViT

Основною перевагою трансформерних відеомоделей є їхня здатність моделювати довготривалі залежності між віддаленими регіонами відео. Однак їхній головний недолік полягає у квадратичній обчислювальній складності механізму уваги відносно кількості патчів. Через це обробка довгих відео високої роздільної здатності є надзвичайно ресурсомісткою. Модель Video Swin Transformer [3] частково вирішує цю проблему за допомогою локальних вікон уваги, що зміщуються, але фундаментальне обмеження, пов'язане з фіксованим пакуванням, залишається.

Окрім трансформерів, для генерації та аналізу відео використовуються й інші архітектури. Класичним підходом є використання тривимірних згорткових нейронних мереж (3D CNN) [8] та архітектур типу UNet-3D [26], які розширюють концепцію 2D згорток на часовий вимір. Моделі на основі векторно-квантизованого варіаційного автокодувальника (VQ-VAE), такі як VideoGPT [9], працюють у стислому латентному просторі, що дозволяє значно зменшити обчислювальну складність. Спочатку навчається автокодувальник для стиснення кадрів у дискретні латентні коди, а потім використовують трансформер для авторегресійного моделювання послідовності цих кодів. Їхній простір застосування достатньо широкий, включно з біометричною верифікацією [27].

Найбільшого прогресу в якості генерації досягли дифузійні моделі [10], [11]. Вони працюють за принципом поступового додавання шуму до даних з подальшим навчанням моделі відновлювати оригінальні дані з цього шуму. Цей підхід дозволяє генерувати надзвичайно реалістичні та когерентні відеопослідовності. Проте, як і більшість сучасних методів, вони часто покладаються на попередньо навчені моделі для стиснення даних (наприклад, Latent Diffusion Models [11]) або працюють з даними, що вже пройшли певну обробку.

На відміну від цих підходів, запропонована архітектура V-BLT є авторегресійною моделлю, яка унікальна своєю здатністю працювати безпосередньо з необробленими байтами відео, не потребуючи попередньої токенізації, стиснення або будь-яких інших евристичних кроків попередньої обробки.

Архітектура моделі Vision Byte Latent Transformer

Запропонована архітектура Vision Byte Latent Transformer наслідує ієрархічний принцип, закладений в оригінальній моделі BLT [4], адаптуючи його для роботи з багатовимірними просторово-часовими даними. Модель складається з трьох основних компонентів: локального кодера, глобального латентного трансформера та локального декодера, які послідовно обробляють відеопотік від необроблених байтів до прогнозованих кадрів.

Загальна структура V-BLT розроблена для ефективної обробки відеоданих шляхом їх ієрархічного перетворення. На відміну від традиційних підходів, що вимагають попередньої токенізації, V-BLT працює безпосередньо з послідовністю байтів, що представляють відео. Потік даних в архітектурі, як зображено на рис. 3, організовано таким чином:

1. Послідовність кадрів відео перетворюється на одновимірний потік байтів;

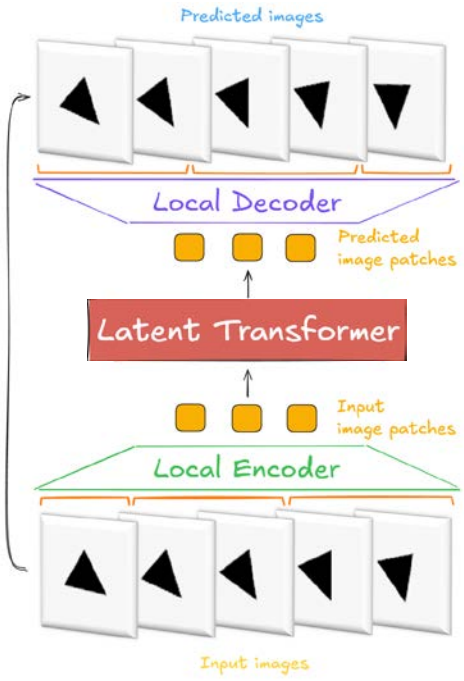

Рис. 3. Загальна архітектура V-BLT

2. Локальний кодер (Local Encoder), що є невеликим трансформером, приймає на вхід потік відео-байтів і агрегує їх у компактні латентні представлення, які називаються просторово-часовими патчами;

3. Глобальний латентний трансформер (Global Latent Transformer) є основним обчислювальним блоком моделі, працює на рівні латентних патчів. Він аналізує послідовність патчів і авторегресійно прогнозує представлення наступних патчів;

4. Локальний декодер (Local Decoder) є останнім модулем, який приймає прогнозовані латентні представлення та «розпаковує» їх назад у послідовність відео-байтів, формуючи прогнозовані кадри відео.

Вхідний відеопотік перетворюється на потік байтів, який обробляється локальним кодером для створення латентних патчів. Глобальний трансформер прогнозує наступні патчі, а локальний декодер відновлює з них прогнозований кадр. Така багаторівнева структура дозволяє моделі ефективно розподіляти обчислювальні ресурси, використовуючи потужний глобальний трансформер для аналізу високорівневих семантичних одиниць (патчів), а не для обробки кожного окремого байта.

Одним з ключових викликів у роботі з розгорнутим потоком байтів є збереження інформації про їхнє початкове просторово-часове розташування. Вхідне відео, що має структуру (T,H,W,C), де T — час, H — висота, W — ширина, а C — кількість каналів, лінеаризується в одновимірну послідовність. Без спеціального механізму кодування позицій модель не зможе розрізнити: два байти були сусідами в просторі, чи в часі.

Для вирішення цієї проблеми запропоновано розширення оригінальної імплементації *методу ротаційних позиційних вкладень* (Rotary Positional Embeddings, RoPE) [12] на *N*-мірний випадок (RoPE-ND [25]). На відміну від класичного RoPE, запропонований варіант дозволяє одночасно кодувати декілька координат, таких як час t, висота y та ширина x.

Нехай вектор вкладення x має розмірність d. Розбиваємо його на $d/2$ пар (x_{2m-1}, x_{2m}). Для кодування позиції (t,y,x) використовуються комплексні числа. Кожна пара розглядається як комплексне число $x_{2m-1} + i \cdot x_{2m}$. Позиційне кодування застосовується шляхом множення цього числа на комплексне число обертання $e^{i\theta_{m,t,y,x}}$, де кут θ залежить від позиції та індексу пари m. Кути обертання розраховуються таким чином, щоб забезпечити унікальне кодування для кожної координати. Для трьох вимірів розмірність ознак d повинна бути кратною шести ($2\times3=6$), щоб забезпечити коректне накладання повороту на кожен з трьох просторово-часових вимірів (в експериментах використовувалось значення 384).

Частоти ω_k, для $k \in [1, d/6]$ визначаються за формулою

$$\omega_k = \frac{1}{\beta^{2k/(d/3)}}, \tag{1}$$

де β — це гіперпараметр (впливає на частоту обертання, в оригінальній імплементації має значення 10000 [12]). Тоді кути для кожної координати визначаються так:

$$\theta_{k,t} = t \cdot \omega_k; \tag{2}$$

$$\theta_{k,y} = y \cdot \omega_{k+d/6}; \tag{3}$$

$$\theta_{k,x} = x \cdot \omega_{k+d/3}. \tag{4}$$

Фінальне обертання для вектора x на позиції (t,y,x) є добутком трьох обертань:

$$\dot{x} = x \cdot e^{i\left(\theta_{k,t} + \theta_{k,y} + \theta_{k,x}\right)}. \quad (5)$$

Такий підхід дозволяє моделі зберігати точну просторово-часову інформацію, що є критично важливим для розуміння динаміки об'єктів у відео.

На поточному етапі дослідження для доказу концепції (proof-of-concept) використовується *статична схема пакування*, де відеопотік розбивається на фіксовані просторово-часові куби байтів розміром $4 \times 4 \times 4$. Це є достатнім для валідації концепції та дозволяє перевірити основні гіпотези архітектури. В майбутньому планується перехід до динамічного пакування на основі значень карт ентропії. Загально, розроблена архітектура містить такі функціональні елементи:

– локальний кодер використовує кілька згорткових шарів для початкової агрегації локальної інформації, що дозволяє зменшити просторову розмірність, а потім неглибокі трансформерні блоки для створення вкладень патчів. Це дозволяє ефективно обробляти локальні залежності;

– глобальний латентний трансформер є основним прогнозувальним модулем. Для забезпечення коректної роботи з відеопослідовностями, в ньому застосовується спеціальна *схема маскування уваги*. Увага є *повнозв'язною* (full attention) для всіх токенів в межах одного 2D кадру, що дозволяє моделі аналізувати просторові зв'язки. Водночас, увага є *каузальною* (causal) вздовж часової осі, тобто токени з кадру t не можуть «бачити» токени з майбутніх кадрів $t+1$, $t+2$, і т.д. Це забезпечує авторегресійний характер моделі;

– локальний декодер виконує зворотну операцію. Він використовує транспоновані згортки (transposed convolutions) для підвищення просторової роздільної здатності та механізм перехресної уваги (cross-attention) для інтеграції інформації з виходу глобального трансформера, що дозволяє відновити деталізований прогнозований кадр на основі інформації про попередні кадри.

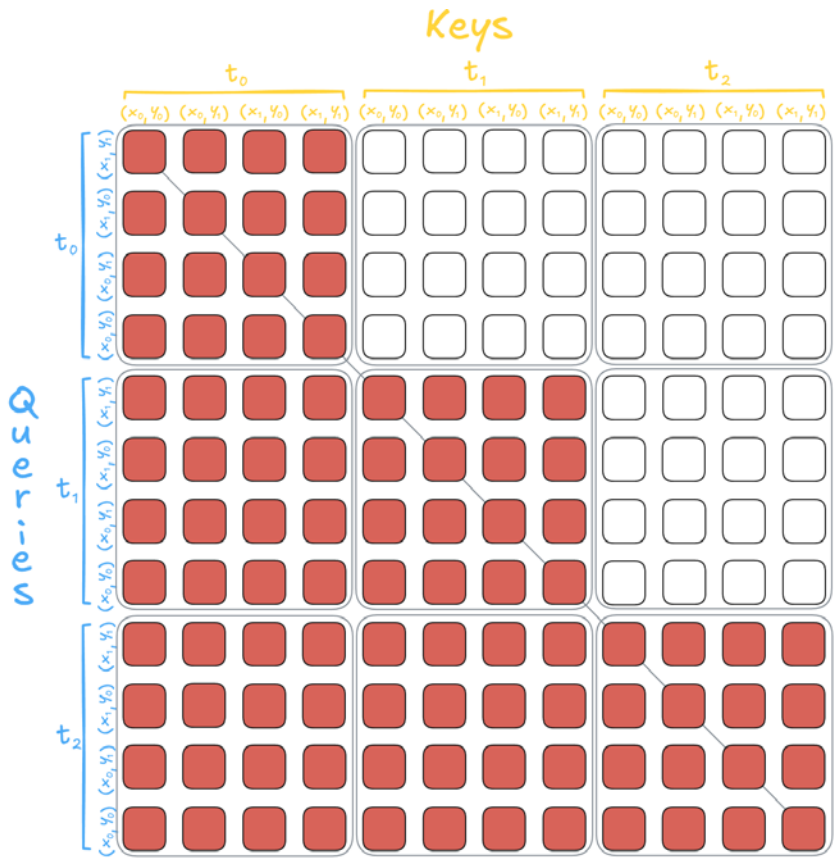

Рис. 4. Схема просторово-часового маскування в Global Transformer

На рис. 4 на прикладі трьох зображень (t_0, t_1, t_2) з розмірністю 2×2 візуалізовано дозволені зв'язки, використовуючи матрицю уваги. Для візуалізації, послідовність з трьох кадрів розгортається в одновимірний вектор довжиною 12 (3 кадри по 4 пікселі). Відповідно, матриця уваги має розмірність 12×12. Діагональні блоки, що відповідають увазі всередині одного кадру, є повними. Зв'язки з минулими кадрами дозволені (нижній червоний трикутник), а з майбутніми — замасковані (верхній білий трикутник). Важливо зазначити, що така схема уваги, хоч і виглядає складною, залишається обчислювально ефективною, оскільки застосовується на рівні латентних патчів, кількість яких значно менша за кількість вихідних байтів (пікселів).

На виході локального декодера знаходиться фінальний класифікаційний шар з активацією softmax, який для кожного пікселя прогнозованого кадру видає розподіл ймовірностей за 256 можливими значеннями байта. Це дозволяє безпосередньо розраховувати ентропію $H(p)$ для кожного пікселя за формулою

$$H(p) = -\sum_{i=0}^{255} p_i \log_2 p_i, \quad (6)$$

де p_i — ймовірність значення байта. Отримана *карта ентропії* є візуальним індикатором невизначеності моделі: висока ентропія відповідає регіонам з високою візуальною складністю, швидким рухом або оклюзією. Ця карта є ключовим сигналом, який у майбутньому може бути використаний для реалізації динамічного пакування, де розмір патчів адаптивно змінюватиметься залежно від складності контенту.

Організація та проведення експериментів

Для оцінки ефективності запропонованої архітектури V-BLT розроблено та реалізовано комплекс експериментів, спрямованих на перевірку її здатності до моделювання та прогнозування просторово-часових послідовностей. Ключовим елементом експериментальної бази є використання спеціалізованих синтетичних наборів даних, що дозволяє проводити аналіз в контрольованому середовищі, мінімізуючи вплив стохастичних факторів, притаманних реальним відеоданим.

Використання реальних відео-датасетів на етапі початкової розробки архітектур є часто неефективним через їхню високу складність та значні обчислювальні вимоги [13]. Для вирішення цієї проблеми, як запропоновано та попередньо реалізовано в попередній роботі [6], створено генератор синтетичних даних. Цей підхід дозволяє створювати нескінченний потік унікальних послідовностей з чітко визначеними правилами руху, що є ідеальним середовищем для ізольованої оцінки фундаментальних властивостей моделі, таких як здатність до відтворення, узагальнення та прогнозування динаміки.

Синтетичний набір даних реалізовано за допомогою мови програмування Python [14] та бібліотеки для роботи з графічними об’єктами PyVista [15], розроблений застосунок є публічно доступним на GitHub авторів [16]. Для цього дослідження використано два типи синтетичних даних: набір даних 2D-фігур та 3D-полікубів, що обертаються, генерація яких детально описана нижче.

Набір даних 2D-фігур призначений для базової перевірки здатності моделі розуміти прості 2D-трансформації. Приклад згенерованої послідовності для 2D-фігури показано на рис. 5.

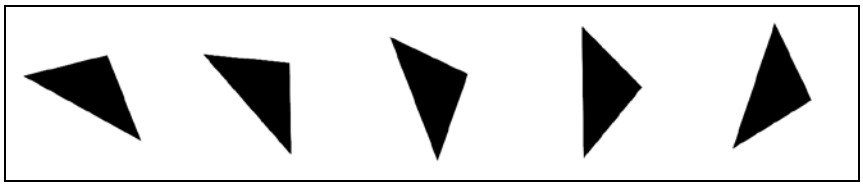

Рис. 5. Приклад згенерованої послідовності для 2D-фігури

Кожна послідовність генерується процедурно за таким алгоритмом:

1. Генерація фігури. Створюється випадковий трикутник. Його форма визначається трьома параметрами: довжиною основи, висотою та зміщенням вершини відносно основи. Ці параметри вибираються випадковим чином у заданих діапазонах, що забезпечує варіативність форм;

2. Ініціалізація сцени. Згенерована фігура розміщується у центрі 2D-сцени. Початкова орієнтація фігури є випадковою — до неї застосовується обертання на випадковий кут навколо перпендикулярної до площини осі зображення;

3. Генерація послідовності. Для кожної послідовності визначається постійна кутова швидкість обертання. На кожному кроці часу фігура обертається на цей фіксований кут. Камера залишається нерухомою, фіксуючи об’єкт з однієї точки. Таким чином, генерується послідовність кадрів, що демонструє плавне обертання.

В свою чергу набір даних 3D-полікубів призначений для тестування моделі на більш складних задачах, що вимагають розуміння тривимірної структури. Приклад згенерованої послідовності для 3D-полікуба показано на рис. 6.

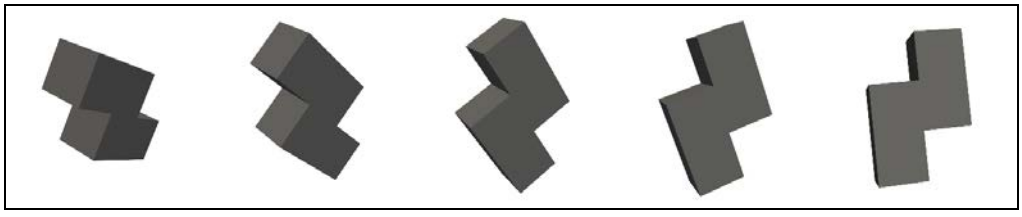

Рис. 6. Приклад згенерованої послідовності для 3D-полікуба

Генерація послідовностей відбувається таким чином:

1. Генерація фігури. Спочатку випадковим чином вибирається кількість кубів n у діапазоні від зазначених мінімальної до максимальної (у наших експериментах від 2 до 6). Далі, з використанням алгоритмів комбінаторного переліку, що базуються на роботах з теорії поліформ [7], вибирається одна з канонічних форм полікуба розміром n. Це гарантує отримання структурно унікальні та нетривіальні об'єкти.

2. Ініціалізація сцени. Згенерований полікуб розміщується в центрі 3D-сцени. Його початкова орієнтація є випадковою: до фігури застосовуються обертання на випадкові кути навколо усіх осей.

3. Генерація послідовності. Для кожної послідовності генерується випадковий вектор кутової швидкості, де кожне значення вибирається із зазначеного діапазону швидкості обертання. На кожному кроці часу до фігури застосовується обертання, визначене цим вектором. Це створює складну траєкторію руху в тривимірному просторі. Камера, як і в 2D-випадку, залишається статичною, спостерігаючи за об'єктом. Це вимагає від моделі розуміння об'ємної структури, сталості об'єкта та ефектів самозатінення.

Для проведення експериментів реалізовано модель V-BLT з використанням вищеописаних архітектурних параметрів. Модель навчалася на задачі прогнозування наступного кадру: на вхід подавалася послідовність з 12 кадрів розмірністю 64×64 з метою спрогнозувати кожен наступний кадр в послідовності на основі попередніх (causal training) [17]. Навчання проводилося з використанням оптимізатора AdamW [18] та функції втрат на основі піксельної перехресної ентропії (pixel-wise cross-entropy) [19], оскільки модель прогнозує розподіл ймовірностей для кожного байта зображення.

Для розробки та тренування моделі використовувалась бібліотека PyTorch [20], відповідний код та нюанси тренування також є публічно доступними на GitHub авторів [21].

Для кількісної оцінки якості прогнозування вибрано набір стандартних метрик, які широко використовуються в задачах обробки зображень та відео [28]. Ці метрики дозволяють оцінити різні аспекти точності моделі: від піксельної схожості до структурної цілісності. Перелік та призначення метрик:

– середньоквадратична похибка (MSE) — оцінює середню різницю між значеннями пікселів у прогнозованому та справжньому зображеннях. Що нижче значення, то краще;

– індекс структурної схожості (SSIM) — вимірює структурну схожість між двома зображеннями, враховуючи яскравість, контрастність та структуру;

– пікове співвідношення сигнал/шум (PSNR) — оцінює якість відновленого зображення шляхом порівняння максимального можливого значення сигналу зі значенням шуму, що спотворює зображення. Більші значення вказують на вищу якість.

Ці метрики розраховувалися для кожного прогнозованого кадру та усереднювалися по всій тестовій вибірці для отримання фінальних значень.

Результати експериментів та їх аналіз

Для оцінки ефективності запропонованої архітектури V-BLT проведено комплексний аналіз її продуктивності на задачах прогнозування просторово-часових послідовностей. Експерименти проводилися окремо на 2D та 3D синтетичних наборах даних, що дозволило отримати як кількісні, так і якісні дані для оцінки роботи моделі на задачах різного рівня складності. Для порівняльного аналізу використано моделі ViViT [1] та UNet-3D [26].

Перш ніж аналізувати якість прогнозування, важливо оцінити обчислювальну ефективність запропонованої архітектури порівняно з базовими моделями. У табл. 1 подано порівняльний аналіз розміру моделей та обчислювальних витрат на одну послідовність.

Таблиця 1

Порівняльний аналіз розміру моделей до обчислювальних витрат

Модель	Розмір (Mb)	Обчислювальні витрати (GFLOPs)
V-BLT	6,5	**51,2**
ViViT	6,5	59,3
UNet-3D	24,0	70,4

З табл. 1 випливає, що модель V-BLT за однакової кількості параметрів з ViViT вимагає на

14 % менше обчислювальних ресурсів. Це підтверджує основну гіпотезу про те, що ієрархічна структура на рівні патчів є ефективнішою, ніж стандартний підхід трансформерів. У порівнянні з архітектурою UNet-3D, V-BLT є значно компактнішою та потребує на 28 % менше обчислень.

Для кількісної оцінки якості прогнозування розраховано значення зазначених вище метрик на тестових вибірках з обох наборів даних на усіх моделях. Результати зведені в табл. 2.

Таблиця 2

Результати кількісної оцінки моделі V-BLT

Набір даних	Модель	MSE (↓)	SSIM (↑)	PSNR (↑)
2D-фігури	V-BLT	**0,039**	**0,98**	**32,3**
	ViViT	0,181	0,87	20,7
	UNet-3D	0,125	0,88	21,8
3D-полікуби	V-BLT	**0,11**	**0,91**	23,4
	ViViT	0,19	0,83	**24,8**
	UNet-3D	0,2	0,75	20,3

Аналіз даних з табл. 2 виявляє ключові переваги архітектури V-BLT у якості прогнозування, особливо в контексті її обчислювальної ефективності. В порівнянні з іншими моделями V-BLT демонструє вищу якість прогнозування як для 2D-фігур так і для 3D-полікубів. Це підкреслює головну перевагу розробленої архітектури: досягнення кращої якості прогнозування за менших обчислювальних затрат, через що архітектура є масштабованішою та ефективнішою. Це є ключовим фактором для масштабування моделі на задачі з високою роздільною здатністю.

Для глибшого розуміння поведінки моделі проведено якісний аналіз згенерованих послідовностей. На рис. 7 та 8 показано приклад прогнозування для простої 2D та складної 3D-послідовності, де на першому рядку — очікувані кадри в послідовності, на другому — спрогнозовані кадри моделлю V-BLT, а на третьому — карти ентропії (чорний колір вказує на високу ентропію, а білий на низьку), що відповідають прогнозованим кадрам. Ключовим аспектом архітектури V-BLT є її здатність працювати з необробленими байтами, що дозволяє розраховувати піксельну ентропію для кожного прогнозу. Ентропія в цьому контексті є мірою невизначеності моделі: що вища ентропія, то менш «впевнена» модель у своєму прогнозі для цього пікселя.

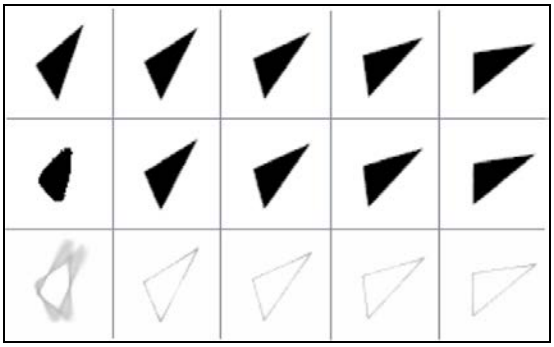

Рис. 7. Приклад роботи моделі V-BLT для 2D-фігури

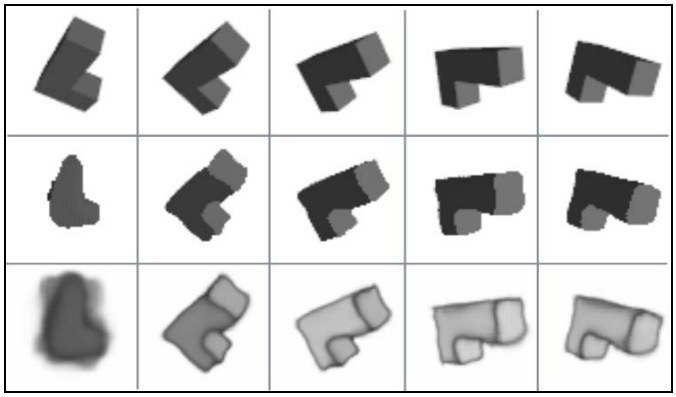

Рис. 8. Приклад роботи моделі V-BLT для 3D-полікуба

На рис. 7 можна спостерігати, що модель демонструє високу точність у прогнозуванні обертання простої 2D-фігури. Спрогнозовані кадри (другий рядок) майже ідентичні очікуваним (перший рядок), за винятком першого прогнозованого кадру, який має помітні артефакти, що пояснюється недостатнім контекстом для прогнозування (детальніше розглянуто нижче).

У свою чергу, на рис. 8 показано результати для складнішої 3D-задачі. Модель V-BLT успішно прогнозує не лише загальне положення та орієнтацію об’єкта, але й відтворює його тонку структуру, включно з правильним затіненням та збереженням форми окремих кубів. Навіть на віддалених кроках прогнозування модель зберігає структурну цілісність об’єкта, що підтверджує її здатність до моделювання довготривалих залежностей.

Аналіз карт ентропії для обох рисунків дозволяє дійти важливого висновку: найвищі значення ентропії систематично концентруються на контурах об’єкта, що обертається, особливо на тих його частинах, які зазнають найбільших змін (наприклад, з’являються через оклюзії або змінюють свій кут відносно камери). Статичний фон, навпаки, має майже нульову ентропію, оскільки його про-

гнозування є тривіальною задачею. Також важливо зазначити, що перший та певною мірою другий кадр мають підвищену ентропію (що особливо помітно в випадку 3D-полікуба), причиною є недостатній контекст інформації про попередні стани фігури, відповідно модель знаходиться в позиції максимальної невизначенності, що продукує невалідний результат передбачення. Таким чином, карти ентропії слугують не лише інструментом аналізу, але й потенційним керувальним сигналом. Цей результат є прямим емпіричним підтвердженням основної гіпотези роботи: візуально складні та динамічні регіони сцени та обмежений контекст дійсно відповідають вищій інформаційній невизначеності. Це також обґрунтовує доцільність майбутнього переходу до динамічного пакування, де модель зможе адаптивно виділяти більше обчислювальних ресурсів саме на такі високоентропійні ділянки.

Отримані результати є обнадійливими та підтверджують життєздатність запропонованого підходу. Водночас, варто зазначити, що експерименти проводилися в контрольованому середовищі синтетичних даних, що є стандартною практикою для початкової валідації нових архітектур. Подальшим логічним кроком є масштабування досліджень на реальні відео-датасети, такі як Kinetics-400 [22] або UCF101 [23], які містять значно більшу варіативність сцен, об’єктів та динаміки. Це дозволить оцінити здатність моделі до узагальнення на дані з високою стохастичністю та складними текстурами.

До того ж, для повної оцінки переваг запропонованої архітектури планується проведення порівняльного аналізу V-BLT з іншими сучасними моделями для прогнозування відео. Для цього може бути використана відкрита платформа для бенчмаркінгу, наприклад, OpenSTL [24], що дозволить провести стандартизоване та об’єктивне порівняння на широкому спектрі наборів даних та метрик.

Висновки

У роботі запропоновано та досліджено нову архітектуру для обробки відеоданих, V-BLT, яка є адаптацією байт-рівневої парадигми, що раніше використовувалася в мовних моделях, до зорової модальності. На відміну від традиційних підходів, що ґрунтуються на фіксованому пакуванні (patching), запропонована модель працює безпосередньо з необробленим потоком байтів відео, що дозволяє уникнути втрати інформації та підвищити гнучкість обробки.

Ключовими елементами архітектури є ієрархічна структура, що складається з локальних та глобального трансформерів, а також використання N-мірних ротаційних позиційних вкладень, які дозволяють зберігати просторово-часову когерентність даних. Експериментальна перевірка проводилася на спеціально розроблених синтетичних наборах даних з 2D та 3D рухомими фігурами. Результати кількісного аналізу продемонстрували високу точність прогнозування майбутніх кадрів, що підтверджується як експериментально на основі метрик так і емпірично на прикладах роботи моделі. Експериментальна перевірка продемонструвала, що розроблена архітектура досягає конкурентної якості прогнозування порівняно з вибраними ViViT та UNet-3D, при цьому вимагаючи менших обчислювальних ресурсів.

Важливим результатом роботи є демонстрація здатності моделі генерувати змістовні піксельні карти ентропії. Показано, що висока ентропія корелює з візуально складними та динамічними регіонами сцени, такими як контури рухомих об’єктів, а також з кадрами, для яких модель має обмежений контекст. Це емпірично підтверджує основну гіпотезу роботи: архітектура V-BLT є не лише точною, але й обчислювально ефективною, а згенеровані карти ентропії відкривають прямий шлях до реалізації повністю динамічного, залежного від контенту розподілу обчислювальних ресурсів.

Таким чином, виконаний огляд та запропоновані рішення можуть бути корисними для дослідників, що працюють над створенням ефективних та масштабованих моделей для обробки відео. Майбутні дослідження можуть бути спрямовані на реалізацію динамічного пакування на основі карт ентропії, а також на масштабування та тестування архітектури V-BLT на складних реальних наборах даних для подальшого вдосконалення її можливостей.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] A. Arnab, et al., “ViViT: A Video Vision Transformer,” in *ArXiv e-prints*, 2021. [Online]. Available: https://arxiv.org/abs/2103.15691 . Accessed: September 26, 2025.

[2] A. Dosovitskiy, et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in *ArXiv e-prints*, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929 . Accessed: September 26, 2025.

[3] Z. Liu, et al., “Video Swin Transformer,” in *ArXiv e-prints*, 2022. [Online]. Available: https://arxiv.org/abs/2106.13230 .

Accessed: September 26, 2025.

[4] A. Pagnoni, R. et al., “Byte Latent Transformer: Patches Scale Better than Tokens,” in *ArXiv e-prints*, 2024. [Online]. Available: https://arxiv.org/abs/2412.09871 . Accessed: September 26, 2025.

[5] L. Xue, A. Barua, et al., “ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models,” *ArXiv e-prints*, 2021. [Online]. Available: https://arxiv.org/abs/2105.13626 . Accessed: September 26, 2025.

[6] Л. Р. Кулик, і О. Б. Мокін, «Створення синтетичного набору даних для оцінювання архітектур нейромережевих моделей,» в *Матеріали LIV науково-технічної конференції підрозділів ВНТУ*, Вінниця, 24-27 березня 2025 р.

[7] G. Aleksandrowicz, and G. Barequet, “Counting polycubes without the dimensionality curse,” *Discrete Mathematics*, vol. 309, no. 13, pp. 4576-4583, 2009. https://doi.org/10.1016/j.disc.2009.02.023 . Accessed: September 26, 2025.

[8] D. Tran, et al., “A Closer Look at Spatiotemporal Convolutions for Action Recognition,” in *ArXiv e-prints*, 2018. [Online]. Available: https://arxiv.org/abs/1711.11248 . Accessed: September 26, 2025.

[9] W. Yan, et al., “VideoGPT: Video Generation using VQ-VAE and Transformers,” in *ArXiv e-prints*, 2021. [Online]. Available: https://arxiv.org/abs/2104.10157 . Accessed: September 26, 2025.

[10] J. Ho, et al., “Video Diffusion Models,” in *ArXiv e-prints*, 2022. [Online]. Available: https://arxiv.org/abs/2204.03458 . Accessed: September 26, 2025.

[11] A. Blattmann, et al., “Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models,” in *ArXiv e-prints*, 2023. [Online]. Available: https://arxiv.org/abs/2304.08818 . Accessed: September 26, 2025.

[12] J. Su, et al., “RoFormer: Enhanced Transformer with Rotary Position Embedding,” in *arXiv e-prints*, 2021. [Online]. Available: https://arxiv.org/abs/2104.09864 . Accessed: September 26, 2025.

[13] A. F. Bobick, and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257-267, 2001.

[14] *Python Software Foundation, Python Language Reference, version 3.12*. [Online]. Available: https://www.python.org . Accessed: September 26, 2025.

[15] C. Sullivan, and B. E. A. Larson, *PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)*. [Online]. Available: https://pyvista.org . Accessed: September 26, 2025.

[16] *Simple Shape Dataset Toolbox GitHub*. [Online]. Available: https://github.com/leo27heady/simple-shape-dataset-toolbox. Accessed: September 26, 2025.

[17] A. Vaswani, et al., “Attention Is All You Need,” in *ArXiv e-prints*, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762 . Accessed: September 26, 2025.

[18] I. Loshchilov, and F. Hutter, “Decoupled Weight Decay Regularization,” in *ArXiv e-prints*, 2017. [Online]. Available: https://arxiv.org/abs/1711.05101 . Accessed: September 26, 2025.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[20] A. Paszke, et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024-8035.

[21] *Vision Byte Latent Transformer GitHub*. [Online]. Available: https://github.com/leo27heady/visionBLT . Accessed: September 26, 2025.

[22] W. Kay, et al., “The Kinetics Human Action Video Dataset,” in *ArXiv e-prints*, 2017. [Online]. Available: https://arxiv.org/abs/1705.06950 . Accessed: September 26, 2025.

[23] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild,” in *ArXiv e-prints*, 2012. [Online]. Available: https://arxiv.org/abs/1212.0402 . Accessed: September 26, 2025.

[24] Tan C, et al., “OpenSTL: A Comprehensive Benchmark of Spatio-Temporal Predictive Learning,” in *arXiv e-prints*, 2022. [Online]. Available: https://arxiv.org/abs/2306.11249 . Accessed: September 26, 2025.

[25] *Rope-Nd GitHub*. [Online]. Available: https://github.com/limefax/rope-nd . Accessed: September 26, 2025.

[26] Ozgun Cicek, et al., “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,” in *ArXiv e-prints*, 2016. [Online]. Available: https://arxiv.org/abs/1606.06650 . Accessed: September 26, 2025.

[27] M. Havrylovych, and V. Danylov, “Research on hybrid transformer-based autoencoders for user biometric verification,” *System Research and Information Technologies*, no. 3, pp. 42-53, 2023. [Online]. Available: https://doi.org/10.20535/SRIT.2308-8893.2023.3.03 . Accessed: September 26, 2025.

[28] Vasyl Lytvyn, et al., “Detection of Similarity Between Images Based on Contrastive Language-Image Pre-Training Neural Network,” *Machine Learning Workshop at CoLInS, 2024*. [Online]. Available: https://doi.org/10.31110/COLINS/2024-1/008 . Accessed: September 26, 2025.

Рекомендована кафедрою системного аналізу та інформаційних технологій ВНТУ

Стаття надійшла до редакції 6.10.2025

Кулик Леонід Русланович — аспірант кафедри системного аналізу та інформаційних технологій, e-mail: leonidkulik2707@gmail.com ;

Мокін Олександр Борисович — д-р техн. наук, професор, професор кафедри системного аналізу та інформаційних технологій, e-mail: abmokin@gmail.com .

Вінницький національний технічний університет, Вінниця

L. R. Kulyk[1]
O. B. Mokin[1]

Scaling Video Prediction with Spatio-Temporal Patches

[1]Vinnytsia National Technical University

The article presents a new architecture for video data processing, the Vision Byte Latent Transformer (V-BLT), which adapts the principles of successful byte-level language models to the visual modality. Unlike standard approaches that use fixed-size patching, which are computationally inefficient due to the uniform resource allocation regardless of visual content complexity, V-BLT operates directly on the video byte stream. This allows for avoiding information loss associated with prior tokenization and enhances processing flexibility. The key contributions include the concept of spatiotemporal latent patches, the implementation of N-dimensional Rotary Positional Embeddings to preserve data coherence in the flattened byte stream, and a multi-level transformer architecture for hierarchical processing. To validate the hypothesis and test the model, a new synthetic dataset with rotating 2D and 3D shapes was developed for a controlled evaluation of the model's spatiotemporal reasoning capabilities. It is experimentally demonstrated that V-BLT effectively predicts future frames, achieving high scores on MSE, SSIM, and PSNR metrics comparing to ViViT and UNet3D with better computational efficiency. The developed architecture according to the design has the ability to generate per-pixel entropy maps that visualize prediction uncertainty and correlate with dynamically complex regions of the scene. This paves the way for the implementation of dynamic, content-dependent, on-the-fly allocation of computational resources, representing a promising direction for creating more efficient and scalable foundation models for video analytics.

Keywords: machine learning, neural networks, natural language processing (NLP), transformers, computer vision (CV), convolutional neural networks (CNN), variational autoencoder (VAE), synthetic data, optimization, artificial neutral networks.

Kulyk Leonid R. — Post-Graduate Student of the Chair of System Analysis and Information Technologies, e-mail: leonidkulik2707@gmail.com ;

Mokin Oleksandr B. — Dr. Sc. (Eng.), Professor, Professor of the Chair of System Analysis and Information Technologies, e-mail: abmokin@gmail.com