УДК 004.8

DOI: 10.15407/fmmit2025.40.130

Логіко-графова інтерпретація сцени в трансформерному ядрі автономного дрона

Назарій Лопух[1], Петро Жук[2], Адріан Торський[3]

[1]к. т. н., ст. н. с. Інститут прикладних проблем механіки і математики ім. Я. С. Підстригача НАН України, вул. Наукова, 3б, Львів, 79060, e-mail: lopuh.nazar@gmail.com

[2] к. т. н., н. с. Інститут прикладних проблем механіки і математики ім. Я. С. Підстригача НАН України, вул. Наукова, 3б, Львів, 79060, e-mail: cipt@litech.net

[3] к. т. н., ст. н. с. Інститут прикладних проблем механіки і математики ім. Я. С. Підстригача НАН України, вул. Наукова, 3б, Львів, 79060, e-mail: adrian@cmm.lviv.ua

У статті представлено концепцію інтелектуального дрона, в якому реалізовано семантичне ядро для побудови логіко-графових представлень середовища на основі сенсорних даних. Основна мета полягає у створенні адаптивної системи, здатної не лише розпізнавати об'єкти, але й формувати причинно-наслідкові та просторові зв'язки між ними для логічного висновування. Запропонований підхід поєднує методи підкріплювального навчання з графовою семантикою в межах трансформерної архітектури, забезпечуючи автономність і гнучкість поведінки в умовах часткової спостережуваності.

Ключові слова:автономна навігація; семантичне моделювання; трансформерна нейромережа; логіко-графова інтерпретація; граф знань; адаптивне управління.

Вступ.У сучасних дослідженнях у сфері автономної робототехніки зростає зацікавлення у створенні інтелектуальних систем, здатних не лише сприймати зовнішнє середовище, а й будувати семантичні інтерпретації на його основі [1-4, 7-8]. Саме ця властивість визначає принципову відмінність між традиційними сенсорними платформами та системами з елементами штучного розуму. Представлений нами проєкт інтелектуального дрона вирізняється впровадженням семантичного ядра, що керує інтерпретацією сцени на основі логіко-графового представлення, інтегрованого у трансформерну архітектуру.

Така система спроможна не лише класифікувати об'єкти у полі зору, а й встановлювати причинно-наслідкові, просторові та функціональні зв'язки між ними. Внаслідок цього дрон отримує здатність до логічного висновування, абстрактного планування та адаптивної поведінки в умовах невизначеності.

У даній статті викладено концептуальну основу проєкту, розглянуто реалізований алгоритм, порівняно нашу систему з існуючими аналогами, а також показано ключові переваги пропонованої моделі.

1. Формулювання задачі

Формалізована задача, яка лежить в основі функціонування інтелектуального дрона, полягає у побудові адаптивного агенту A, що,

взаємодіючи з частково спостережуваним динамічним середовищем E, генерує оптимальну політику прийняття рішень $\pi : S \rightarrow U$, де S — простір сенсорних станів, а U — множина допустимих дій. Сенсорні дані $x(t) = \{I_t, \omega_t, g_t\}$, які надходять у дискретні моменти часу $t \in N$, слугують вхідним потоком до семантичногопроцесора, що будує граф знань$G_t = (V_t, E_t)$ з урахуванням поточних об'єктів, їх властивостей та зв'язків.

Мета полягає у знаходженні таких параметрів θ, які мінімізують узагальнену функцію втрат:

$$L(\theta) = E_{x(t)}[-R(\pi_\theta(x(t))) + \lambda D(G_t)],$$

де $R(u)$— очікувана винагорода за дію u, $D(G_t)$—функціонал невизначеності або конфліктності в структурі знань, а λ — ваговий коефіцієнт регуляризації.

Таким чином, задача зводиться до поєднання стратегій підкріплювального навчання та логіко- семантичного контролю на основі графових структур у трансформерному середовищі.

2. Наукове обґрунтування

Класичні нейромережі комп'ютерного зору (*CNN, ViT*) працюють на рівні візуальних ознак, однак ігнорують вищі рівні абстракції — логіку, причинність, контекстуальність. Для досягнення семантичної інтерпретації сцени, ми вводимо семантичне ядро: окремий модуль, який перетворює вхідні дані (зображення, карти ознак, попередні гіпотези) у граф знань, що репрезентує факти, гіпотези, об'єкти, події та зв'язки між ними у вигляді формальних логічних тверджень.

Формально, кожна сцена S проєктується у граф $G = (V, E)$, де:

- V — множина об'єктів та подій (наприклад: трактор, колія, розворот)
- E— множина зв'язків (наприклад: спричинює, випливає з, розташований поруч)

Цей граф не є статичним — він розвивається в часі, приймає гіпотези, оновлює структури знань за допомогою логічного резонування.

Архітектура реалізації. Граф знань

Кожна сцена парситься у вигляді графу, реалізованого через *networkx.DiGraph*, з додатковими тегами для логічних категорій:

- вузли мають тип (*object, evidence, hypothesis*)
- ребра мають тип відношення (*causes, supports, observedFrom, inferredFrom*)

Семантична обробка

Для генерації та оновлення графа використовуються:

- прості логічні правила (на основі фреймів: якщо є *A і B*, то гіпотеза *H*)
- попередньо навчені класифікатори для виявлення об'єктів (наприклад, на *ViT* або*YOLO*-базі)
- історія графів (дедуктивний апдейт з урахуванням попередніх кадрів)

3. Обґрунтування вибору трансформерів

Аналіз архітектур показав, що серед доступних підходів найкраще завданням відповідає використання моделі типу *Perceiver IO,* яка забезпечує здатність до масштабованої обробки мультиканальних вхідних даних при збереженні обчислювальної ефективності. Формально, Perceiver працює як функція

$$y = f_\theta(inputs) = \mathrm{f}_\theta\left(\left\{x_i^{vision}, x_j^{GPS}, x_k^{IMU}\right\}\right)$$

перетворюється через загальний латентний простір фіксованої розмірності. Такий підхід дозволяє поєднати різнотипні сигнали без необхідності проектування спеціалізованих каналів обробки.

На етапі комп'ютерного зору використовується попередньо натренований *Vision Transformer (ViT-B/16),* перенавчений для задачі семантичної сегментації і класифікації об'єктів на зображенні з подальшою передачею ознак до Perceiver-блоку [9]. Завдяки властивостям ViT розділяти вхідне зображення на фіксовані патчі розміром $P \times P$ та агрегувати інформацію через механізм багатоголової уваги:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V,$$

забезпечується гнучка фільтрація зорових патернів.

Для формування політики дій було протестовано кілька моделей, включно з *RNN* та *CNN-LSTM* архітектурами, однак трансформер на базі Decision Transformer продемонстрував стабільнішу динаміку траєкторії під час моделювання задачі в симуляції. Його ключова перевага — представлення задачі керування як автопідказки в послідовному форматі (return-to-go + state + action), що дозволяє перетворити її на задачу умовного моделювання:

$$\pi(a_t|R_t,s_t,a_{t-1},\dots) = \mathrm{Transformer}(R_{\le t},s_{\le t},a_{<t})$$

Для практичної реалізації обрано комбінацію *ViT + Perceiver IO + Decision* Transformer, що дає змогу повністю відмовитись від класичних модулів SLAM чи rule-based планування.

3.1 Інтерпретований трансформер

Vision Transformer (*ViT*), що обробляє сцену як послідовність візуальних патчів, відзначається гнучкістю, але страждає від надмірної уваги до візуально сильних, але семантично неважливих зон [5-7]. У нашій системі ми запроваджуємо механізм семантично маскованої уваги, який змушує трансформер концентрувати увагу лише на тих парах ознак, які дозволені логічними зв'язками з побудованого семантичного графа (з пункту 1).

Інтуїтивно: модель повинна мати право "уважно дивитися" лише на ті фрагменти сцени, які мають сенс у контексті. Наприклад, якщо виявлено колію, то увагу потрібносконцентрувати на потенційних тракторах і зонах руху, а не на неважливому фоні (трава, хмари).

Ми модифікуємо стандартний *self-attention* блок у такий спосіб:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)V,$$

де *M* – семантична маска, яка задає дозволені пари патчів згідно з графом з попереднього етапу.

3.2 Інтеграція графа у трансформер

- До кожного патча зображення прив'язується індекс логічного об’єкта.
- Із семантичного графа формується матриця маскування attention.
- Ця маска подається в кожен шар трансформера, перешкоджаючи неінформативним з'єднанням.

Інтелектуальні системи обробки сцен стикаються з фундаментальною проблемою: формальна модель повинна не лише виявляти об’єкти на зображенні, а й встановлювати між ними логічні, причинно-наслідкові та часові зв’язки. У межах нашого проєкту «Інтелектуальний дрон» ця задача вирішується за допомогою семантичного ядра, яке перетворює вхідну сенсорну інформацію у структурований граф знань. У цій статті ми пропонуємо розширення базової архітектури: включення мультиграфів та локальних систем знань.

3.3 Формальна структура базового графа знань

У початковій реалізації сцена *S* проектується у граф знань *G = (V, E)*, де *V* — множина об'єктів та подій, а *E* — множина зв'язків між ними. Ми використовуємо орієнтований граф (*networkx.DiGraph*) з розміткою типів вузлів (*object, evidence, hypothesis*) та типів ребер (*causes, supports, observedFrom, inferredFrom*).

Проте у багатьох випадках між тими самими вузлами виникають кілька одночасних зв’язків різного типу (наприклад, одночасно «спричинює» і «спостерігається разом»). Для цього необхідна більш потужна структура — мультиграф.

У контексті сценічної інтерпретації мультиграф дозволяє одночасно фіксувати множинні семантичні відношення між об’єктами.

Це розширення надає більшу гнучкість для reasoning-модуля, оскільки дозволяє локальні контексти враховувати кілька гіпотез паралельно.

Для забезпечення масштабованості і контекстуалізації знання необхідно враховувати, що кожен фрагмент сцени має власну локальну семантику. Ми вводимо поняття локальних систем знань — підграфів, які описують окремі ділянки сцени або окремі часові фрейми. Такі підграфи мають власні онтології і локальні правила логічного висновку.

Семантична інтеграція: від локального до глобального

У фінальній системі інтеграція знань відбувається через об'єднання локальних підграфів у єдину структуру. Це забезпечує модульність, розширюваність і ефективність reasoning-процедур. Конфлікти між локальними інтерпретаціями вирішуються через пріоритети довіри, історичну згоду або байєсівське оновлення ваги гіпотез.

Запропоноване розширення архітектури семантичного ядра інтелектуального дрона дозволяє перейти від плоского графа до структурованого мультиграфа з локальними зонами знань. Це відкриває шлях до глибшого

розуміння сцен, побудови складних логічних ланцюгів, і, зрештою, до реального reasoning на основі інтегрованої інформації.

У майбутній роботі планується формалізація онтологій для типових сцен, використання графових трансформерів для генерації гіпотез та вивчення алгебраїчної когерентності між локальними

4. Розширений алгоритм функціонування системи

Нехай $x(t) = \{I_t, \omega_t, g_t\}$ — повний сенсорний стан у момент часу t, де I_t— RGB-зображення, ω_t—орієнтація з IMU, а g_t— координати GPS. Метою є знаходження оптимальної дії u(t), що максимізує функціонал досягнення мети:

$$L(\theta)=E_{x(t)}[R(u(t))-\lambda D(G_t)],$$

де R—функція винагороди, $D(G_t)$ —міра невизначеності у графі знань.

Алгоритм виконується ітеративно з оновленням інформаційної бази та адаптивним reasoning:

1.Сенсорний вхід $x(t)$ проходить попередню обробку та декомпозицію у множину патчів$\{p_i\}$.

2.Векторні уявлення патчів надходять у модуль первинного аналізу (ViT або Perceiver), що працює з attention-маскою M, сформованою на основі G_{t-1}.

3.Одночасно генерується або уточнюється граф G_t на основі об'єктного детектора, просторової локалізації та внутрішньої онтології.

4.Вхід G_t інтегрується у *DecisionTransformer*, що моделює послідовність дійна основі цільової функції та історії траєкторії.

5.У випадку логічного конфлікту (наприклад, коли одне джерело вказує на наявність об'єкта, а інше— на його відсутність), запускається локальна система reasoning для перегляду структурних зв'язків і пріоритетів.

6.Виведена дія $u(t)$ передається на виконавчі модулі дрона, якіреалізують рух, зйомку або іншу дію.

7.Після дії виконується зворотний зв'язок, що оновлює не лише стан дрона, а й граф G_{t+1}, ураховуючи зміни у довірі до джерел.

Цей цикл є автономною когнітивною петлею, в якій трансформер виступає не лише як інструмент обробки вхідних даних, а як елемент моделі світу.

5. Засоби реалізації на Python

Вибір мовного середовища зумовлений вимогами до гнучкості, розширюваності та сумісності з бібліотеками глибокого навчання. Python — домінуюча мова в розробці нейромереж, що забезпечує необхідну інтеграцію з фреймворками машинного навчання, пакетами комп’ютерного зору, оптимізації inference і апаратними прискорювачами. У проєкті використано чітко визначену структуру інструментального стеку.

Основним фреймворком глибокого навчання є PyTorch, що забезпечує динамічну побудову обчислювальних графів і оптимізацію через torch.jit та torch.fx. Навчання трансформерів здійснюється з використанням бібліотеки HuggingFace Transformers, яка надає доступ до моделей ViT, Perceiver, Decision

Transformer у відкритому форматі з можливістю перенавчання. Для обробки потоків сенсорних даних використано ROS2 як середовище розподіленої обробки повідомлень, що дозволяє інтегрувати камери, IMU, GPS і керування у вигляді топіків.

У контексті inference на пристроях із обмеженими ресурсами здійснюється компіляція моделей у формат ONNX, з подальшою оптимізацією за допомогою NVIDIA TensorRT.

Для потокової обробки відео, побудови графів і семантичних карт використано OpenCV у зв'язці з PyTorch. Наприклад, формування входу для ViT здійснюється через перетворення кадру з камери в набір патчів розміром 16×16 пікселів.

У таблиці1 наведено обрані бібліотеки, їхню роль і причини вибору.

Табл.1 Бібліотеки

Бібліотека	**Призначення**	**Обґрунтування вибору**
torch, torchvision	Тренування й розгортання моделей	Гнучкість, сумісність з edge-optimize toolkits
transformers	Інтеграція готових трансформерів	Підтримка ViT, Perceiver, GPT-like моделей
onnx, onnxruntime	Конвертація моделей	Інференс без залежності від PyTorch
tensorrt	Апаратне прискорення	Глибока інтеграція з Jetson-архітектурою
ros2	Сенсорна синхронізація	Стандарт де-факто для робототехнічних систем
opencv-python	Обробка відеопотоку	Стабільність, швидкість

6. Порівняння з існуючими системами

Сучасні автономні системи управління безпілотниками зазвичай базуються на поєднанні згорткових нейронних мереж (CNN) для візуального аналізу та рекурентних архітектур (LSTM або GRU) для моделювання послідовностей. У більш розвинених підходах, як-от Decision Transformer або Perceiver IO, застосовуються моделі трансформерного типу, однак їхнє використання зводиться переважно до безструктурної обробки сенсорного потоку. Семантичне представлення середовища в таких підходах або відсутнє зовсім, або використовується лише на стадії навчання у вигляді допоміжної розмітки.

На відміну від них, наша система інтегрує граф знань як активну складову процесу уваги, прийняття рішень і reasoning. Це дозволяє не лише враховувати наявність об'єктів у сцені, а й встановлювати складні логічні зв’язки між ними: причинність, спільну локалізацію, функціональні ролі та часові залежності. У

структурі трансформера attention-маска генерується з урахуванням топології семантичного графа, що значно зменшує кількість надлишкових зв'язків і фокусує модель на релевантних аспектах сцени.

Експериментальна перевірка проводилася у середовищах CARLA та AirSim із використанням симульованих урбаністичних сценаріїв. Модель, що базується на ViT з модулем reasoning, показала підвищення точності навігаційних рішень на 14,2% у порівнянні з класичним ViT та на 9,5% у порівнянні з Perceiver IO. У сценаріях з частковою перешкодженістю сцени (перешкоди, дим, несподівані об'єкти) наша система зберігала ефективність завдяки гнучкому оновленню графа знань у реальному часі.

Ще однією відмінністю є здатність до reasoning у конфліктних умовах — наприклад, коли GPS-сигнал не відповідає візуальним даним. Завдяки локальним підграфам система може підтримувати кілька альтернативних гіпотез і обирати дії з урахуванням імовірнісного висновку. Такий підхід відсутній у класичних end-to-end системах, де модель часто має єдину траєкторію і не реагує на невизначеність.

Таким чином, представлена архітектура вирізняється не лише якісно новим рівнем інтерпретації середовища, а й високою адаптивністю, стійкістю до нестачі даних і модульністю, що полегшує масштабування системи для складніших середовищ.

Висновки.

Запропонована архітектура інтелектуального дрона з семантичним ядром демонструє принципово новий підхід до автономної інтерпретації середовища. На відміну від традиційних нейромереж, що працюють на рівні поверхневих ознак, система, описана у цій роботі, інтегрує логічне мислення через граф знань, що дозволяє дрону виконувати висновування, аналізувати складні ситуації, враховувати контекст і адаптувати поведінку в режимі реального часу.

Емпіричні та теоретичні дослідження показують, що поєднання трансформерної обробки з логіко-графовим контролем значно підвищує інтерпретованість та надійність автономних систем. Подальші дослідження мають бути спрямовані на інтеграцію дедуктивних логік, мультиграфових структур та систем розподіленого знання, що дозволить реалізувати повноцінну модель семантичної автономії в складних середовищах.

Пропонована система демонструє новий рівень автономності, де дрон — це не просто сенсорна платформа, а суб'єкт із елементами семантичного мислення.

Подальші дослідження будуть спрямовані на інтеграцію 3D-моделі сцени, багатомодальні онтології та спайкові трансформери, що знизить енергоспоживання при збереженні когнітивних можливостей.

Література

[1] *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I.* Attention is All You Need // Advances in Neural Information Processing Systems. – 2017. – Vol. 30.

[2] *Battaglia P., Hamrick J., Bapst V.* et al. Relational inductive biases, deep learning, and graph networks // arXiv preprint. – 2018. – arXiv:1806.01261. – 21 p.

[3] *Pearl J. Causality*: Models, Reasoning and Inference. – 2nd ed. – Cambridge: Cambridge University Press, 2009. – 432 p.

[4] *Marcus G.* The Next Decade in AI: Why Common Sense Matters // arXiv preprint. – 2020. – arXiv:2002.06177. – 12 p.

[5] *Kipf T. N., Welling M.* Semi-Supervised Classification with Graph Convolutional Networks // Proc. Int. Conf. on Learning Representations (ICLR). – 2017.

[6] *Hamilton W. L., Ying R., Leskovec J.* Representation learning on graphs: Methods and applications // IEEE Data Engineering Bulletin. – 2017. – Vol. 40, No. 3. – P. 52–74.

[7] *Chen Y., Dai X., Liu M.* et al. A Survey on Vision Transformer // arXiv preprint. – 2021. – arXiv:2012.12556. – 30 p.

[8] *Brachman R. J., Levesque H. J.* Knowledge Representation and Reasoning. – San Francisco: Morgan Kaufmann, 2004. – 381 p.

[9] *Zellers R., Bisk Y., Farhadi A., Choi Y.* From Recognition to Cognition: Visual Commonsense Reasoning // Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). – 2019. – P. 6720–6731.

Logical-Graph Interpretation of Scene in the Transformer Core of an Autonomous Drone

Nazariy Lopukh, Petro Zhuk, Adrian Torskyy

This paper presents the concept of an intelligent drone equipped with a semantic core designed to construct logical-graph representations of the environment based on sensory data. The main objective is to develop an adaptive system capable not only of object recognition but also of forming causal, spatial, and functional relationships between objects to enable logical reasoning. The proposed approach integrates reinforcement learning techniques with graph-based semantics within a transformer architecture, thus ensuring autonomous operation and behavioral flexibility in partially observable environments.

Keywords: residual autonomous navigation; semantic modeling; transformer neural network; logical-graph interpretation; knowledge graph; adaptive control.

Отримано 30.07.2025