

Prof. dr hab. inż. Bogusław Cyganek
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Informatyki, Elektroniki i Telekomunikacji
Instytut Elektroniki
cyganek@agh.edu.pl

Kraków, 11.09.2026

Recenzja rozprawy doktorskiej
Pana magistra inżyniera Jakuba Bednarka
Composite neurosymbolic architectures for learning and reasoning
from real and abstract data

Wstęp

Recenzja rozprawy doktorskiej Pana magistra inżyniera Jakuba Bednarka pt. „*Composite neurosymbolic architectures for learning and reasoning from real and abstract data*”¹. Rozprawa powstała w roku 2026 na Politechnice Poznańskiej. Promotorem pracy jest Pan prof. dr hab. inż. Krzysztof Krawiec. Przygotowanie recenzji zostało wykonane na zlecenie Dziekana Wydziału Informatyki i Telekomunikacji Politechniki Poznańskiej, Pana dra hab. inż. Mikołaja Morzego, prof. PP. Rozprawa doktorska została napisana po angielsku. Recenzja została przygotowana w formie odpowiedzi na pytania dotyczące rozprawy doktorskiej Pana mgr inż. Jakuba Bednarka.

1. Problem badawczy oraz jego znaczenie

Jaki jest najważniejszy problem rozważany w rozprawie?

Żyjemy w czasach przełomu technologicznego związanego z powstaniem tzw. sztucznej inteligencji – AI (ang. *Artificial Intelligence*). Są to głównie systemy operujące na sztucznych sieciach neuronowych oraz tzw. transformerach (ang. *transformers*), które dzięki możliwościom obliczeniowym systemów komputerowych ostatniej dekady, osiągają wcześniej niespotykane możliwości wnioskowania za pomocą komputerów. Zaproponowane zaledwie kilka lat temu struktury transformerowe oraz – zbudowane na ich bazie – wielkie modele językowe wprowadziły jeszcze większe możliwości generowania treści, które zbliżyły się, a w wielu zakresach nawet przekroczyły możliwości ludzkiego intelektu. Na bazie tych systemów możliwy jest rozwój kolejnych technologii, takich jak między innymi robotyka oraz automatyczne generowanie programów komputerowych. Mimo jednak przełomowych osiągnięć systemów opartych o uczenie z danych, takich jak głębokie sieci neuronowe (ang. *deep neural networks* – DNN) oraz transformery i wielkie modele językowe (ang. *large language models* – LLM) – których działanie może sprawdzić każdy z nas, choćby wpisując pytanie w przeglądarce internetowej czy na platformie ChatGPT – systemy które łączyłyby struktury neuronowe z obliczeniami symbolicznymi ciągle są w fazie badawczej. Jak zwykle w nauce, nie jest jasne czy przyniosą one

¹ Złożone architektury neurosymboliczne do uczenia się i wnioskowania na podstawie danych rzeczywistych i abstrakcyjnych

kolejne przełomowe rozwiązania, ale na pewno jest to bardzo ciekawy naukowo i obiecujący kierunek badań.

Rozprawa doktorska Pana mgra inż. Jakuba Bednarka wpisuje się w wyżej wymienione dyscypliny naukowe, ze szczególnym naciskiem na roboty oraz systemy bazujące na architekturach głębokich struktur neuronowych uzupełnionych o metody wnioskowania symbolicznego.

Czy praca ma charakter naukowy?

Rozprawa doktorska Pana mgra inż. Jakuba Bednarka niewątpliwie posiada walory naukowe. Zaproponowane przez niego metody oraz rozwiązania bazują na wiedzy Autora w zakresie osiągnięć w dziedzinie AI, jak również obejmują jego osobiste dokonania w tym zakresie. Doktorant jest autorem lub współautorem co najmniej dziesięciu publikacji naukowych zgodnych z linią dyscypliny informatyka techniczna i telekomunikacja.

Czy praca ma znaczenie praktyczne?

Praca Pana mgra inż. Jakuba Bednarka posiada również możliwości zastosowań praktycznych. Autor przedstawił metody sterowania robotami, które mogą być zastosowane w realnych systemach tego typu, poruszających się np. w nieznanym terenie.

2. Wkład Autora

Jaki jest najważniejszy wkład Autora rozprawy?

Gdy przeczytałem wstęp do pracy p. mgra inż. Jakuba Bednarka, a szczególnie jego główną tezę czyli połączenie obliczeń symbolicznych (ang. *symbolic computing*) z NN liczyłem na przełomową pracę. Liczyłem tym bardziej, że zarówno ja, jak i wielu naukowców zastanawia się i pracuje nad tego typu rozwiązaniami. Co prawda osobiście jestem gorącym orędownikiem „zaszcicia” mechanizmów *à la* język PROLOG, tym niemniej obietnice Doktoranta w tym zakresie były co najmniej zachęcające. Niestety, wraz z lekturą pracy p. mgra inż. Jakuba Bednarka narastało we mnie raczej rozczarowanie. W swojej znacznie przydługiej pracy Autor wielokrotnie powtarza swoją tezę „*This thesis addressed the challenge of bridging this gap through composite neuro-symbolic architectures*” ale – moim zdaniem – niestety, jest to stwierdzenie na wyrost. Nie oznacza to jednak, że Autor nie wykonał oryginalnej pracy badawczej, czy też że nie ma wiedzy – jest wręcz przeciwnie! Główny jednak problem polega na tym, że rozprawa doktorska p. mgra inż. Jakuba Bednarka, która nosi datę 2026, to w rzeczywistości kompilacja prac Doktoranta wraz z zespołem naukowym z okresu ostatnich prawie 10 lat. Już sam fakt takiej kompilacji nie jest czymś co bym zalecał jako struktura rozprawy doktorskiej w roku 2026, ale zasadniczy problem polega na tym, że właśnie przez tę ostatnią dekadę nastąpiła rewolucja technologicznie-cywilizacyjna, która niewątpliwie znajdzie się na kartach historii. Chodzi mi tutaj właśnie o rozwój AGI (ang. *Artificial Generative/General Intelligence*), która zupełnie „przewraca” istniejący porządek na świecie – od nauki w szkołach, poprzez medycynę, procesy technologiczne, aż do geopolityki i porządku światowego. Tymczasem Autor wydaje się tego faktu nie zauważać. Prezentuje swoje rozwiązania, które rzeczywiście były nowatorskie i pomysłowe ale w latach ich ukazywani się – i ten fakt możemy dopiero docenić analizując każdą z publikacji, w której p. mgr inż. Jakub Bednarek jest współautorem, ale nie z samej jego rozprawy doktorskiej, którą oceniamy w roku 2026! A jakżeż takie porównanie byłoby ciekawe! Zastosowania CNN (ang. *convolutional neural networks*) oraz ViT (ang. *visual transformer*) choćby w robotyce, której jest poświęcona połowa

tekstu rozprawy, są niebotyczne. Praktycznie codziennie pojawiają się publikacje w znaczących periodykach światowych dotyczące choćby detekcji przeróżnych obiektów, nawet takich jakie nie były użyte podczas uczenia sieci – jest to tzw. *zero-shot learning* – co zupełnie zmienia możliwości ich użytkowników. Bez problemu możemy znaleźć systemy detekcji podłoża, znaków drogowych, typów wydobywanych skał w kopalni, terenu, obiektów podwodnych i tysięcy innych klas specyficznych dla różnorodnych dziedzin nauki i techniki. Metody te już od dawna znajdują zastosowanie w wielu dziedzinach, w tym w robotyce i systemach agentowych, często operując w tzw. trybie czasu rzeczywistego.

Ale to dopiero początek, a opisanie stanu AGI choćby w ostatnich trzech latach wymagałoby kilkudziesięciu stron, których właśnie bardzo mi brakuje w rozprawie p. mgra inż. Jakuba Bednarka. Mimo iż choćby pobieżny przegląd najciekawszych rozwiązań AGI wymagałby wielu stron tekstu, jako wskazówki do porównania chciałbym podsunąć Doktorantowi systemy oparte na strategii uczenia ze wzmocnieniem (ang. *reinforcement learning* – RL), które w wielu dziedzinach już dowiodły nadludzkich możliwości. Mam tu na myśli serię programów komputerowych Alpha-xxx. Jest to choćby model AlphaZero, opracowany już w roku 2017 przez firmę DeepMind, który pokazał możliwość uczenia systemów agentowych – początkowo głównie gier – na zasadzie uczenia się na błędach podczas gry ze sobą w pętli wyników-nagród-modyfikacji itd. Oczywiście tak spektakularne wyniki tej grupy systemów były wynikiem algorytmów opracowanych przez ludzi, ale również synergii z nowoczesnymi platformami obliczeniowymi opartymi na jednostkach obliczeniowych GPU (ang. *graphics processing units*). Systemy bazujące na tej strategii – między innymi AlphaDev oraz AlphaGo – szybko wykazały możliwości gry, najpierw w szachy, a następnie w grę Go, przekraczające umiejętności największych mistrzów. Kolejnymi przełomowymi wersjami tej grupy metod są AlphaFold – przełomowe rezultaty w predykcji struktur białkowych (nagroda Nobla w 2024 dla Demis Hasabis i John Jumper)². Przełomowy moim zdaniem jest też system AlphaTensor, który również bazując na metodzie AlphaZero, dokonał nowatorskiego odkrycia szybszego algorytmu mnożenia tensorów, w dziedzinie w której ostatnie „ludzkie” osiągnięcia datowane były na koniec lat sześćdziesiątych (algorytm Strassen)³. Metodyka RL wraz z modyfikacjami znajduje wiele zastosowań, w tym do trenowania systemów agentowych. Systemy bazujące na RL są jednak znacząco różne od metodyki zaproponowanej przez Doktoranta, a zakładającej uzupełnienie sieci CNN regułami wnioskowania symbolicznego. Co prawda Doktorant w paru miejscach wymienia samą nazwę RL ale bez poświęcenia jej większej uwagi, a tym bardziej bez krytycznej analizy swoich rozwiązań w kontekście choćby tej metodyki, a nie jest to jedyne podejście. Na temat RL znajdujemy następujące stwierdzenie „*Although deep reinforcement learning (RL) has shown promise, high sample inefficiency complicates training robust policies directly in the real world*”, powołując się przy tym na pracę Dulac-Arnold *et al.* z roku 2019 (str. 4 rozprawy). Stwierdzenie to sugeruje jakby rozprawa była pisana co najwyżej w roku 2020, a nie w 2026.

Kolejne przełomowe rozwiązanie AI, które znalazło wielkie i błyskawiczne zastosowanie na całym świecie to wielkie modele językowe LLM, bazujące na różnorodnych architekturach transformerowych. Znalazło ono tak szeroki oddźwięk, że w dzisiejszych czasach praktycznie trudno się na nie natknąć – choćby używając popularnych przeglądarek internetowych. Systemy takie jak ChatGPT, Claude, czy Copilot⁴ – a jest ich znacznie więcej – umożliwiają generowanie praktycznie wszelkich rodzajów

² Jumper J. et al.: *Highly accurate protein structure prediction with AlphaFold*. Nature, No. 596, 2021.

³ Fawzi A. et al.: *Discovering faster matrix multiplication algorithms with reinforcement learning*, Nature, Vol. 610, Iss. 7930, 2022.

⁴ Qile Jiang et al.: *DeepSeek vs. ChatGPT vs. Claude: A comparative study for scientific computing and scientific machine learning tasks*, Theoretical and Applied Mechanics Letters, Vol. 15, Issue 3, 2025.

tekstów, począwszy od rozprawek, wierszy i piosenek, pism prawniczych, analiz medycznych, a na kodzie komputerowym kończąc. To właśnie w tym ostatnim kontekście byłoby bardzo ciekawe porównanie rozwiązań zaproponowanych przez Doktoranta. Niestety w rozprawie takiej analizy nie znajdziemy, a Autor – mimo iż parokrotnie wymienia LLM – wydaje się jednak wręcz pomijać i umniejszać rolę LLM („*stochastic parrots*”), ponownie cytując prace z roku 2021, jak również z lat dziewięćdziesiątych ubiegłego stulecia. Czytając to wszystko odnoszę wrażenie cofnięcia się w czasie, podczas gdy choćby miesiąc temu otrzymaliśmy kolejną przełomową wiadomość o rozwiązaniu przez AI, wyposażonego w gromadę kilku tysięcy autonomicznych agentów, milenijnego problemu dotyczącego zbadania istnienia i gładkości rozwiązań równań Naviera-Stokesa (*openai.com*, 8 września 2026)⁵.

Od lat też prowadzone są liczne badania nad połączeniem aproksymujących właściwości głębokich sieci neuronowych z twierdzeniami i prawami fizycznymi – są to tzw. sieci neuronowe uwzględniające prawa fizyki (ang. *physics-informed NN*; *theory-trained NN*)⁶. Umożliwiają one już w procesie uczenia sieci na wprowadzenie *a priori* zewnętrznej wiedzy o prawach fizyki rządzących danym zbiorem danych; np. prawach, które można opisać za pomocą cząstkowych równań różniczkowych itp. Dzięki temu, tak trenowane sieci mogą znajdować rozwiązania, a następnie je generalizować, nawet mimo często dość ubogich zbiorów treningowych. Warto by poszukać analogii między systemami zaproponowanymi przez Doktoranta a tego typu podejściem.

Ciągle trwają też prace nad budową modeli trenowanych w stylu od początku do końca (ang. *end-to-end*) oraz na innych architekturach np. typu nauczyciel-student (ang. *teacher-student*), których wspólnym celem jest zbudowanie wielkich modeli fundamentalnych, np. DINOv2,3 (ang. *Self-Distillation with No Labels*) itp. Cechą tego typu modeli jest możliwość wnioskowania i odkrywania podczas już samej predykcji wzorców, których nie widziały podczas procesu uczenia – jest to tzw. uczenie typu *zero-shot*. Liczne publikacje na temat tego typu systemów, wykorzystujących modele fundamentalne w wielu szerszych zastosowaniach, świadczą o istotnej skuteczności tego podejścia, mimo iż nie bazują one na *a priori* zadanych komponentach symbolicznych. Co równie istotne, wymienione tu pokrótce głębokie struktury neuronowe podczas uczenia tworzą warstwy semantyczne, które reprezentują kolejne poziomy „wiedzy” wyuczonej z danych – są to tzw. *embeddingi*, które można traktować jako rodzaj wewnętrznych „słów” na których przeprowadzana jest klasyfikacja wzorców. W wielu rozwiązaniach to właśnie te wewnętrzne reprezentacje wykorzystywane są m.in. do szybkiego poszukiwania informacji (ang. *content based image retrieval*), do przetwarzania danych wielomodalnych (np. *contrastive language-image pretraining*, CLIP), czy też do systemów tłumaczących wyniki predykcji za pomocą AI (ang. *explainable AI*).

Po tym krótkim przeglądzie bieżącego stanu i w kontekście zagadnień naukowych podjętych w rozprawie p. mgr inż. Jakuba Bednarka trudno się tutaj zgodzić ze stwierdzeniami Autora, szczególnie bez gruntownej ewaluacji naukowej: „*This cognitive perspective forms the core premise of this thesis. It suggests that scalable, robust AI cannot be achieved solely by scaling data and network parameters. Instead, architectures must separate the “intuitive” perception of the world from the “logical” manipulation of its rules. This distinction requires a structural integration of neural learning and symbolic reasoning*” (str. 3).

Tym niemniej badania nad systemami neuro-symbolicznymi są jak najbardziej uzasadnione i w przyszłości mogą przynieść kolejne przełomowe rozwiązania. Opierają się one m.in. na reprezentacji wiedzy oraz automatycznym wnioskowaniu symbolicznym, jak również na logice klasycznej oraz

⁵ <https://www.quantamagazine.org/ai-has-solved-one-of-maths-1-million-millennium-prize-problems-20260908/>

⁶ Qile J. et al. *AgenticSciML: collaborative multi-agent systems for emergent discovery in scientific machine learning*. npj Artificial Intelligence, 2026.

rozmytej, co dostarcza dodatkowe ścieżki wnioskowania obok tych wytrenowanych w warstwowej sieci neuronowej. Systemy tego typu ciągle znajdują się w fazie badań, podczas gdy szeroko rozwijane są systemy o których była mowa we wcześniejszych akapitach tej sekcji. Prace badawcze p. mgra inż. Jakuba Bednarka wpisują się w ten nowatorski nurt badań. Autor prowadził je od około dziesięciu lat będąc członkiem zespołów naukowych. Główne osiągnięcia p. mgra inż. Jakuba Bednarka zaprezentowane w jego rozprawie doktorskiej są następujące:

1. *Opracowanie metody TransCoder do rozwiązywania abstrakcyjnych problemów za pomocą neuronowej syntezy programów.*

System TransCoder operuje z wykorzystaniem języka dziedzinowego, a po odpowiednim wytrenowaniu z wykorzystaniem zbioru danych „benchmarkowych” Abstract Reasoning Corpus (ARC, zaproponowany przez F. Cholleta, twórcy m.in. platformy Keras na TensorFlow, w pracy z 2019 roku) umożliwia rozwiązywanie domenowych abstrakcyjnych zadań za pomocą syntezy kodu. ARC to kolekcja 800 zadań – 400/400 treningowych oraz testowych – polegających na znajdowaniu mapowań pomiędzy specyficznymi obrazami w rodzaju rastrowych mozaik o wymiarach ok. 30×30 pikseli, a każdy piksel może przybierać do 10 różnych kolorów. Zbiór treningowy (demonstracyjny) zawiera trzy do sześciu par wzorców, natomiast partycja testowa to przeważnie pojedyncza para. Dla każdego zadania mapowania istnieje jedna reguła takiego mapowania, jednakże nieznana dalszym modułom wnioskującym; muszą one się tego mapowania nauczyć na podstawie danych treningowych. Zadanie znajdowania tego typu mapowań należy do trudnych i stanowiło jeden z konkursów platformy Kaggle⁷, a sama baza ARC jest używana do testowania stopnia „inteligencji” systemów AI.

System TransCoda łączy w sobie zalety sieci neuronowych oraz wnioskowania symbolicznego. Zaproponowany przez p. mgra inż. Jakuba Bednarka system rozwiązywania problemów tego typu wykorzystuje reaktywnego agenta, którego zadaniem jest właśnie zsyntezowanie kodu programu do rozwiązania abstrakcyjnego zadania. Kod ten stanowi przykład domenowego języka komputerowego (ang. *domain-specific language* – DSL), który różni się od języków uniwersalnych typu Python, czy C++ tym, że jest on ukierunkowany na rozwiązywanie wyłącznie zadań o znacznie ograniczonym zakresie. Sukcesywne wykonanie w ten sposób zsyntezowanego programu dostarcza rozwiązanie wyjściowego zadania. Zaprojektowany przez Doktoranta z zespołem system posiada cechę uczenia TransCoda na podstawie pomyłek (ang. *learning from mistakes*).

Architektura TransCoda składa się z modułu percepcyjnego, modułu rozwiązywania problemu (tzw. solver/solwer) oraz modułu generatora kodu programu. Zarówno moduł percepcji, jak i solwer są zbudowane na bazie sieci neuronowych. Zadaniem modułu percepcyjnego jest odpowiednie przekodowanie przestrzeni wejściowego problemu dopasowania obrazów rastrowych do przestrzeni reprezentacji wewnętrznej o stałym rozmiarze, która jest łatwiejsza do przetwarzania przez kolejne moduły.

Z kolei, zadaniem solwera jest wydobycie reguł dopasowywania wzorców na podstawie dostępnej partycji przykładów treningowych. Reguły te są oczywiście używane później w trybie predykcyjnym, w którym system operuje wyłącznie na partycji testowej.

Z kolei generator kodu używa właśnie na przetwarzaniu symbolicznym, które korzysta z informacji dostarczanych przez sieci neuronowe. Moduł generatora używa pamięci przetwarzania symboli (tzw. workspace), które operują z wykorzystaniem zarówno „wiedzy” specyficznej do zadania, jak i wspólnej dla całej przestrzeni rozwiązań. Generator kodu operuje na przestrzeni wewnętrznej zbudowanej przez moduł percepcyjny i konstruuje abstrakcyjne

⁷ <https://www.kaggle.com/competitions/abstraction-and-reasoning-challenge/overview>

drzewo syntaktyczne (ang. *abstract syntax tree* – AST), używane do reprezentacji programów komputerowych zgodnie z gramatyką ich języków.

Ostatni z zaproponowanych modułów to interpreter. Jego zadaniem jest wykonanie programu zakodowanego w odpowiednim drzewie AST, używając obrazów rastrowych jako danych wejściowych, a następnie produkując odpowiadające im obrazy rastrowe jako dane wynikowe. Porównanie tego wyjścia z „prawdziwymi” danymi, tj. pochodzącymi z danych testowych, pozwala określić stopień dopasowania i w rezultacie dokładność całego systemu.

System TransCoda został przedstawiony w dwóch bardzo podobnych publikacjach⁸, a następnie w trzeciej, tj. rozprawie doktorskiej.

2. Opracowanie neuro-symbolicznego systemu do syntezy programów.

Pan mgr inż. Jakub Bednarek zaproponował modularny system do syntezy relatywnie prostych programów komputerowych bazując na języku naturalnym. Zaproponowany system otrzymuje krótki opis żądanej funkcjonalności z języka naturalnego, a następnie generuje wykonywalny fragment kodu. Autor zaproponował połączenie modułu enkodera dostarczającego tzw. osadzenia (*embeddings*) reprezentujące informacje zawarte w tekście języka naturalnego, a jednocześnie reprezentujące pewien rodzaj przestrzeni wewnętrznej (ang. *latent space*), które następnie są przetwarzane przez podwójnie rekurencyjne dekodery, również trenowane z wykorzystaniem abstrakcyjnych drzew składniowych (struktur dobrze znanych z teorii języków i kompilatorów komputerowych). W celu zmapowania osadzeń do przestrzeni informacji wewnętrznej p. mgr inż. Jakub Bednarek zaproponował użycie wielowarstwowych sieci typu LSTM (ang. *long short-term memory*). Są to sieci, które do nadejścia epoki transformerów, często były używane do przetwarzania struktur sekwencyjnych, w tym szeregów czasowych. Autor dodatkowo zaproponował ciekawy schemat propagacji sygnału oraz mechanizm miękkiej uwagi (ang. *soft attention*). System ten został przetestowany w zadaniach syntezy programów w języku *AlgoLisp*, zaproponowanych w pracy autorstwa Polosukhina i Skidanova z roku 2018. Jest to istotne osiągnięcie p. mgr inż. Jakuba Bednarka zaprezentowane w rozprawie, które jednak wcześniej było opublikowane w publikacji trzech autorów (Doktorant jako pierwszy) na konferencji ICLR 2019.

3. Opracowanie głębokich sieci plotowych sieci neuronowej do rozpoznawania typów terenu dla nawigacji robota cztero-kończynowego w różnych warunkach. Pan mgr inż. Jakub Bednarek wraz z zespołem opracowali tu dwa systemy:

A. Pierwszy to system usprawniający poruszenie się robota wielokończynowego dzięki pomiarowi sygnałów zamontowanych na jego stopach. Przenoszą one informacje o rodzaju podłoża, które są następnie przekodowywane za pomocą opracowanych struktur sieci neuronowych na informacje, które usprawniają proces poruszania się robota. Pan mgr inż. Jakub Bednarek wraz z zespołem opracowali struktury sieci CNN oraz RNN które efektywnie przetwarzają informacje o sile i momencie obrotowym bez konieczności rozwiązywania wprost równań fizycznych, czy też polegania na innych cechach eksperckich (ang. *handcrafted features*). Klasyfikacja typu podłoża obejmowała 8 klas terenu, a do trenowania Autor musiał przygotować kilka tysięcy próbek pomiarów robota kroczącego w różnym terenie. W sumie użyto 7018 próbek do trenowania oraz 1775 do testowania systemu. Jak pokazały eksperymenty, zastosowanie tego modułu znacznie zwiększa dokładność trajektorii poruszania się robota.

⁸ (1) arXiv:2410.04480v1 [cs.AI] 6 Oct 2024

(2) <https://journals.sagepub.com/doi/10.1177/29498732261419309>

B. Drugi w tej kategorii to system wspomagający proces poruszania się robota typu ANYmal na podstawie pomiaru punktu podparcia jego stóp. Główny moduł to opracowana przez p. mgra inż. Jakuba Bednarka wraz z zespołem sieć neuronowa do przetwarzania danych wejściowych w postaci map wysokościowych podłoża (ang. *elevation map*). Wyjściem natomiast są parametry punktu podparcia dla stóp robota zapewniających optymalne kroczenie, dodatkowo przekonwertowane do identyfikatora klasy odpowiadającego danemu rodzajowi terenu. Jest to więc rodzaj kwantyzacji w celu przejścia od problemu regresji do klasyfikacji. Jednocześnie sieć ta ma zadanie zastąpić wcześniej używany do tego zadania model analityczny. Architektura sieci to tzw. efektywna sieć rezydualna ConvNet (ang. *efficient residual factorized ConvNet* – ERF). W eksperymentach użyto wspomnianego robota ANYmal wraz z symulatorem Gazebo w scenariuszach poruszania się po schodach oraz po nierównym terenie. Robot używał sensora RGB-D do zapisu mapy otoczenia. Użyto też gotowego modelu planowania trajektorii, natomiast pomiarowi podlegał jedynie model punktu podparcia stóp.

4. *Opracowanie głębokich splotowych sieci neuronowej do wyboru tzw. punktu podparcia (ang. foothold) dla systemu cztero-kończynowego robota oraz mechanicznie adaptacyjnej miękkiej stopy robota.*

Jest to rozwinięcie prac p. mgra inż. Jakuba Bednarka wraz z zespołem nad systemami obliczenia optymalnych punktów podparcia dla stóp robota kroczącego. W tym przypadku Autorzy używają mechanicznie adaptacyjnej miękkiej stopy robota. Doktorant wraz z zespołem ponownie użyli różnych wariantów sieci typu ERF do emulacji modelu analitycznego, który na podstawie równań wynikających z praw fizyki oblicza mapę kosztu na podstawie mapy elewacji terenu. Zastosowana sieć umożliwiła emulację tego typu obliczeń za pomocą systemu wytrenowanego wyłącznie za pomocą danych. Dokonuje ona analizy lokalnego kształtu podłoża oraz parametrów punktu podparcia stóp w celu predykcji wynikowej mapy kosztu punktów podparcia stóp biorąc pod uwagę potencjalne kolizje pomiędzy nogami robota a różnie ukształtowanym podłożem. System został zamontowany, a następnie przetestowany na platformie cztero-kończynowego robota ANYmal wykazując poprawę jakości trajektorii poruszania się robota przy zastosowaniu zaproponowanych sieci dzięki możliwości adaptacyjnego dostosowywania się każdej ze stóp robota do zmieniającego się podłoża.

Po uwzględnieniu odpowiedniego przedziału czasowego i stanu wiedzy można zakwalifikować te opracowania p. mgra inż. Jakuba Bednarka jako istotne i nowatorskie rozwiązania Doktoranta. Niestety p. mgra inż. Jakuba Bednarka wydaje się nie zauważać upływającego czasu, a przede wszystkim przełomowych osiągnięć w dziedzinie którą się zajmuje i które wprost można odnieść do jego prac. Jest to niewątpliwa i niestety nie jedyna wada rozprawy doktorskiej p. mgra inż. Jakuba Bednarka.

5. Poprawność naukowa

Czy stwierdzenia zawarte w rozprawie są godne zaufania? Czy uzasadnienia są poprawne? Wskaż zauważone słabości i błędy. Wskaż także te aspekty dotyczące poprawności, które są najbardziej wartościowe.

Główna słabość rozprawy doktorskiej p. mgra inż. Jakuba Bednarka to przedstawienie swoich rozwiązań, które opracowywał od roku 2018 bez odwołania i porównania z systemami obecnie istniejącymi. Była o tym już mowa w poprzednim rozdziale tej recenzji. Autor nie wykazał wyższości, czy też dominujących zalet proponowanych przez niego rozwiązań bazujących na systemach neuro-

symbolicznych, w porównaniu do znanych i wymienionych już systemów AI, a wielokrotne powtarzanie w rozprawie „*bridging the gap*” to zaklinanie rzeczywistości, które nie zmienia stanu rzeczy.

Najważniejszy system opracowany przez p. mgra inż. Jakuba Bednarka TransCoder doczekał się kolejnej edycji konkursu Kaggle⁹. Konkurs ten, który zakończył się w roku 2025, zaowocował kilkoma ciekawymi rozwiązaniami. Aż dziwi mnie, że w swojej rozprawie doktorskiej z 2026 Doktorant ani razu nie wspomniał, ani w żaden inny sposób nie odwołał się do tych nowszych rozwiązań. Jak porównuje się system TransCoda na ich tle, łącznie z konkursem z roku 2020?

Negatywnie oceniam też dosłowne przepisywanie zarówno tekstu, jak i kopiowanie rysunków i diagramów z wcześniejszych publikacji autora do rozprawy doktorskiej autora, jednakże bez zaznaczenia tego faktu, ani bez każdorazowego podania odwołań do stosownych publikacji. Niestety tego typu miejsc pomiędzy rozprawą a publikacjami jest wiele. Dla przykładu Figure 11.2, Figure 11.3, Figure 11.4 w rozprawie, wraz z ich podpisami i otaczającym tekstem, to wierna kopia Figure 2, Figure 3 oraz Figure 4 wraz z tekstem z pracy Jakub Bednarek, Karol Piaskowski & Krzysztof Krawiec: *Ain't Nobody Got Time For Coding: Structure-Aware Program Synthesis From Natural Language*, opublikowanej w roku 2018 na konferencji ICPR 2019. Podobnie Figure 12.1 (i kolejne) wraz z podpisem z rozprawy jest identyczne do Figure 1 z tekstem z publikacji J. Bednarek i K. Krawiec: *Learning To Solve Abstract Reasoning Problems With Neurosymbolic Program Synthesis And Task Generation*, Neurosymbolic Artificial Intelligence IOS Press, 2026. Niestety przykładów tego typu zapożyczeń jest w rozprawie doktorskiej p. mgra inż. Jakuba Bednarka znacznie więcej. Zupełnie nie rozumiem i nie popieram takiego podejścia do prezentowania swoich osiągnięć przez doktorantów. Jasne jest, że Doktorant bazuje na pracy naukowej, którą przeprowadzał przez wiele lat i która znalazła odzwierciedlenie w licznych publikacjach – i to jest chwalebne. Wystarczy więc przytoczyć odwołania do tych prac, albo – jeśli to możliwe – zamieścić ich wierne reprints po uzgodnieniu z wydawcą. Cenniejsze byłoby więc pozostawienie tekstów w stosownych publikacjach i syntetyczne zarysowanie kontekstu podjętej tematyki, a przede wszystkim porównanie opracowanych przez autora rozwiązań z obecnymi systemami AI, czyli powiedzmy tymi które zostały opracowane od 2024.

Ponadto, ponieważ publikacje p. mgra inż. Jakuba Bednarka są wieloautorskie to warto aby Doktorant jasno określił swój oryginalny wkład naukowy w każdą z nich. W kontekście każdej, nie mniej cenne byłoby choć krótkie odniesienie się do stanu współczesnego rozwoju AI.

Oprócz powyższych zastrzeżeń, kolejne, bardziej szczegółowe uwagi i pytania do treści naukowej rozprawy p. mgra inż. Jakuba Bednarka są następujące:

1. Pojęcie „*structural isomorphism*”, wielokrotnie używane w rozprawie (np. str. 7 i kolejne), wymaga głębszego wyjaśnienia i uzasadnienia.
2. Schemat zaprezentowany na Figure 1.1 wymaga głębszego uzasadnienia szczególnie w kontekście postępów w tej dziedzinie i w uwzględnieniu stanu na rok 2026.
3. Czym się różni diagram z Figure 1.2 w stosunku do tego z Figure 1.1?
4. Doktorant powinien lepiej uzasadnić/zdefiniować pojęcie autonomicznego agenta.
5. Autor prezentuje szereg wzorów, np. (3.7-3.10) ... (4.5-4.9) ... (8.2-8.4) itd. – jaki jest cel tych prezentacji? Gdzie te wzory są używane w pracy Doktoranta?
6. Gdzie Doktorant używa Algorithm 1 ze strony 41?
7. Str. 44 – „*trajectory diverges from the ground truth*”, ale jaka dokładnie ta odchyłka była? Jak była mierzona?
8. Str. 47 – czy użycie wyłącznie 8 klas podłoża jest wystarczające w praktyce?

⁹ <https://www.kaggle.com/competitions/arc-prize-2025/discussion/652078>

9. Str. 48 – co to jest „*terrain semantics*”?
10. W przypadku głębokiej splotowej sieci neuronowej do rozpoznawania kilku typów terenu dla nawigacji robota cztero-kończynowego, czy można by tu zastosować podejście sieci neuronowej uwzględniającej prawa fizyki, o którym pisałem w poprzednim rozdziale?
11. Str. 58, 62 – prace sprzed 2012 roku nieco straciły na aktualności, jeśli nawet nie na wartości, dlatego że zasadniczo zmieniły się paradygmaty AI. Czy Doktorant mógłby podać nowsze publikacje?
12. W rozdziale 7 Autor zaproponował architekturę sieci typu efektywna sieć rezydualna ConvNet (ERF). Czemu taka architektura, która nie należy do najnowszych? Czy Autor rozważał inne rodzaje sieci? Jaką architekturę moglibyśmy użyć w roku 2026?
13. Autor przetestował swoje systemy robotyczne wyłącznie z użyciem jednego typu robota. Jak i czy można te metody „ekstrapolować” na inne typy podobnych, tj. kroczących robotów? Jak wygląda stan wiedzy w tej dziedzinie w roku 2026?
14. Rozdział 7.2.3 – czy możliwe byłoby jednak użycie jakiś nowszych architektur NN, np. YOLO?
15. Table 7.1 – wyniki podano dla partycji walidacyjnej, czyli biorącej udział w procesie uczenia, a jakie są wyniki przy użyciu „czystej” partycji testowej?
16. Rozdział 7.2.4 – czy nie można użyć żadnej z dobrze znanych metod augmentacji danych? Przecież już w rozdziale 7.2.2 autor wspomina „flip”; a może dałoby się użyć np. sieci GAN?
17. Rozdział 7.3 – sieć neuronowa w tym systemie ma ściśle określone i ograniczone zadania. A czy można by jednak wytrenować model całościowy typu end-to-end do sterowania robotem tego typu?
18. Str. 73 – „*The selected footholds are close to the nominal foothold*” – skąd ta wiedza, że obydwie pozycje są „blisko” siebie?
19. Rozdział 8.2 – jak ten „*cost of potential footholds*” jest zbierany w danych treningowych? Przecież musimy go otrzymać *a priori*, skoro nim uczymy CNN.
20. Str. 78 – „*CNN provides the cost map, which is used to modify the nominal foothold*”, a czy sieć nie mogłaby już od razu podawać wynikowej wartości optymalnego położenia stopy robota?
21. Algorithm 2 – a nie możemy tego całego algorytmu wbudować w sieć neuronową?
22. Table 8.1 – nie jest jasne jak były mierzone parametry zamieszczone w tej tabeli.
23. Str. 82 – „*a compromise between the preferences given by the neural network and the position of the nominal foothold*”, nie jest jasne jak ten „kompromis” jest w praktyce obliczany.
24. Table 8.2 – dziwna konstrukcja prostokątnej *confusion matrix*; wymaga wyjaśnienia.
25. Rozdział 10 – brakuje głębszego porównania użytej metodyki do metodyki uczenia ze wzmocnieniem RL.
26. Rozdział 11.2 – czy użycie transformerów zamiast zaproponowanych LSTM usprawniłoby system?
27. Rozdział 11.3.1 – autor wspomina osiągnięcie wyników SOTA, ale w porównaniu z jakimi pracami ta konkluzja?
28. Str. 117 – czy możliwe byłoby tu użycie transformera zamiast zaproponowanej reprezentacji neuronowej?
29. Str. 118 – „*do not attempt to formalize the semantics*”, może chodziło jednak o *syntax*?
30. Str. 118 – autor wspomina język PROLOG, ale czy zamiast użytego DSL można by użyć podzbioru bardziej popularnego języka, np. Python?
31. Str. 121 – jakiego typu jest gramatyka użytego DSL?
32. Str. 122 – „*The ineffectiveness of RL motivates the SL mode*”, o jakiej nieefektywności RL pisze Doktorant? Jak ma się to stwierdzenie do systemów opartych na architekturze AlphaZero i pochodnych?

33. Str. 144 – „how to design” właśnie siła współczesnych CNN itd. tkwi w tym że są bardzo uniwersalne i że w zasadzie nie trzeba ich specjalnie projektować – w wielu przypadkach wystarczy zebrać odpowiednio duże dane i otagować, np. YOLO może detekować wszystko co wynika z danych, a nie z jego architektury. Czy Doktorant może więc doprecyzować swoje konkluzje?

Powyższe pytania i sugestie poszerzenia o dodatkowych zagadnienia mają charakter dyskusyjny i nie umniejszają istotnego wkładu merytorycznego przedstawionego w rozprawie Pana mgra inż. Jakuba Bednarka.

6. Wiedza Kandydata

Które z rozdziałów rozprawy omawiają istniejący stan wiedzy i dzięki temu potwierdzają ogólny stan wiedzy kandydata w zakresie informatyki? Jakie obszary tych dyscyplin zostały omówione w tych rozdziałach/sekcjach? Jaka jest opinia recenzenta o bibliografii? Prosimy o podanie innych argumentów za lub przeciw, że kandydat posiada ogólną wiedzę w dyscyplinie ITT.

Rozprawa doktorska Pana mgra inż. Jakuba Bednarka jest bardzo obszerna – liczy 184 strony. Zawiera ona aż 15 rozdziałów uzupełnionych bibliografią. Rozprawa składa się z dwóch niezbyt ze sobą związanych części. Pierwsza to prace p. mgra inż. Jakuba Bednarka związane z opracowaniem nowych metod usprawnienia poruszania się robotów. Drugi, zaczynający się od rozdziału 10, jest poświęcony obliczeniom neuro-symbolicznym. Lektura rozprawy p. J. Bednarka nie jest łatwa, tym bardziej że oparta została na materiałach zamieszczonych w artykułach, w których p. mgr inż. Jakub Bednarek jest jednym z autorów, tyle że udział treści z publikacji w stosunku do nowych treści zawartych tylko w rozprawie nie został jasno zaznaczonych, a bardzo szkoda. Praca zawiera zarówno treści zaawansowane, jak np. opis TransCoda, ale również i naiwne proste, np. rozdział 3.3 z próbą wytłumaczenia działania najprostszycy sieci typu CNN. Tego typu styl pisanie rozpraw doktorskich może być uzasadniony kilkadziesiąt lat temu, kiedy nie było szybkiego Internetu, łatwo dostępnych artykułów oraz innych treści, a przede wszystkim AI. Tym niemniej, mimo niekiedy nieco przesadzonej opinii na temat donośności swoich osiągnięć, rozprawa zawiera bardzo bogaty materiał naukowy będący wynikiem kilkunastu lat pracy doktoranta, oczywiście przy współpracy z zespołem badawczym do którego należy.

Na uznanie zasługuje lista publikacji obejmująca około 200 pozycji. Świadczy ona o dużym czytaniu Doktoranta w tematach ściśle związanych z dyscypliną informatyka techniczna i telekomunikacja.

Pan mgr inż. Jakub Bednarek ma ugruntowaną pozycję w świecie naukowym, o czym świadczą parametry związane z aktywnością publikacyjną Doktoranta. Prezentuje je poniższa tabela (na podstawie bazy Web of Science).

Parametr	Wartość w serwisie Web of Science (wrzesień 2026)
H-Index	6
Liczba publikacji	11
Liczba cytowań (bez auto-cytowań)	125

Liczba publikacji wynosząca 11 jest wysoka, mimo iż są to prace wieloautorskie. W tym zestawie najwyżej punktowanym jest artykuł pt. „*Learning-Based Foot-Shape-Aware Foothold Selection for Quadrupedal Robots*”, opublikowany w roku 2025 w IEEE Transactions on Systems, Man, and Cybernetics: Systems, w którym p. mgr inż. Jakub Bednarek jest trzecim w gronie pięciu autorów.

Stosunkowo wysoki jest też tzw. współczynnik Hirscha, którego wartość wynosi $H=6$. Świadczy to o dobrej „widoczności” przez światowe środowisko naukowe prac, w których p. mgr inż. Jakub Bednarek jest współautorem.

Powyższe fakty świadczą o głębokiej wiedzy Doktoranta Pana magistra inżyniera Jakuba Bednarka w dyscyplinie informatyka techniczna i telekomunikacja, a w szczególności w dziedzinie uogólnionej sztucznej inteligencji.

7. Podsumowanie

Biorąc pod uwagę moją ocenę zaprezentowaną w poprzednich punktach, jak również wymagania zdefiniowane przez art. 187 Ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce, z późniejszymi zmianami, stwierdzam że rozprawa doktorska Pana magistra inżyniera Jakuba Bednarka prezentuje ogólną i głęboką wiedzę Kandydata w dyscyplinie informatyka techniczna i telekomunikacja, a w szczególności w zakresie nowoczesnych metod sztucznej inteligencji. Rozprawa doktorska Pana Jakuba Bednarka świadczy o umiejętności samodzielnego prowadzenia badań naukowych na wysokim światowym poziomie, jak również o dobrym opanowaniu warsztatu badawczego. Zaprezentowane w rozprawie doktorskiej Pana magistra inżyniera Jakuba Bednarka metody i algorytmy są oryginalne i dotyczą rozwiązania istotnych problemów naukowych dotyczących uogólnionej sztucznej inteligencji. Opracowane metody stanowią oryginalny i własny wkład Doktoranta Pana magistra inżyniera Jakuba Bednarka w rozwój dyscypliny naukowej informatyka techniczna i telekomunikacja.

Recenzowaną pracę oceniam jako z spełniającą formalne oraz zwyczajowe wymagania stawiane rozprawom doktorskim. Wnioskuje o jej przyjęcie oraz o dopuszczenie Pana magistra inżyniera Jakuba Bednarka do publicznej obrony.