Článok

Detekcia novinky signálu ako skutočná odmena za robotiku

Martin Kubovčík 1,*, Iveta Dirgová Luptáková 2 a Jiří Pospíchal 3,*

[1] Katedra aplikovanej informatiky, Fakulta prírodných vied, Univerzita sv. Cyrila a Metoda, J. Herdu 2, 917 01 Trnava, Slovensko; kubovcik1@ucm.sk

[2] Katedra aplikovanej informatiky, Fakulta prírodných vied, Univerzita sv. Cyrila a Metoda, J. Herdu 2, 917 01 Trnava, Slovensko; iveta.dirgova.luptakova@ucm.sk

[3] Katedra aplikovanej informatiky, Fakulta prírodných vied, Univerzita sv. Cyrila a Metoda, J. Herdu 2, 917 01 Trnava, Slovensko; jiri.pospichal@ucm.sk

* Korešpondencia: jiri.pospichal@ucm.sk ; kubovcik1@ucm.sk

Abstract: In advanced robot control, reinforcement learning is a common technique used to transform sensor data into signals for actuators, based on feedback from the robot's environment. However, the feedback or reward is typically sparse, as it is provided mainly after the task's completion or failure, leading to slow convergence. Additional intrinsic rewards based on the state visitation frequency can provide more feedback. In this study, an Autoencoder deep learning neural network was utilized as novelty detection for intrinsic rewards to guide the search process through a state space. The neural network processed signals from various types of sensors simultaneously. It was tested on simulated robotic agents in a benchmark set of classic control OpenAI Gym test environments (including Mountain Car, Acrobot, CartPole, and LunarLander), achieving more efficient and accurate robot control in three of the four tasks (with only slight degradation in the Lunar Lander task) when purely intrinsic rewards were used compared to standard extrinsic rewards. By incorporating autoencoder-based intrinsic rewards, robots could potentially become more dependable in autonomous operations like space or underwater exploration or during natural disaster response. This is because the system could better adapt to changing environments or unexpected situations.

Keywords: anomaly detection; autoencoder; signal processing; intrinsic reward; robotics; exploration

Abstrakt:V pokročilom riadení robotov je učenie zosilnenia bežnou technikou používanou na transformáciu údajov zo senzorov na signály pre akčné členy na základe spätnej väzby z prostredia robota. Spätná väzba alebo odmena je však zvyčajne riedka, pretože sa poskytuje hlavne po dokončení alebo zlyhaní úlohy, čo vedie k pomalej konvergencii. Ďalšie vnútorné odmeny založené na frekvencii štátnych návštev môžu poskytnúť viac spätnej väzby. V tejto štúdii bola neurónová sieť s hlbokým učením Autoencoder použitá ako detekcia novosti pre vnútorné odmeny na vedenie procesu vyhľadávania cez stavový priestor. Neurónová sieť spracovávala signály z rôznych typov senzorov súčasne. Bol testovaný na simulovaných robotických agentoch v benchmarkovej sade klasických testovacích prostredí OpenAI Gym (vrátane Mountain Car, Acrobot, CartPole a LunarLander), čím sa dosiahlo efektívnejšie a presnejšie ovládanie robota v troch zo štyroch úloh (len s miernym zhoršením v úloha Lunar Lander), keď sa použili čisto vnútorné odmeny v porovnaní so štandardnými vonkajšími odmenami. Začlenením vnútorných odmien založených na autokódovači by sa roboti mohli stať spoľahlivejšími v

autonómnych operáciách, ako je prieskum vesmíru alebo pod vodou alebo počas reakcie na prírodné katastrofy. Systém by sa totiž mohol lepšie prispôsobiť meniacemu sa prostrediu alebo neočakávaným situáciám.

Kľúčové slová:detekcia anomálií; autokóder; spracovanie signálu; vnútorná odmena; robotické; prieskum

---

1. Úvod

Posilňovacie učenie je dobre známy prístup k výučbe robotov v simulovaných prostrediach. V tejto metóde je robot považovaný za agenta, ktorý dostáva informácie vo forme pozorovaní o sebe a svojom prostredí. Agent – simulovaný robot vykonáva akcie v prostredí a dostáva signál odmeny ako spätnú väzbu. Algoritmus učenia sa potom na základe prijatého signálu odmeny snaží predpovedať ideálne akcie, ktoré povedú agenta k úspechu. Existujú však simulácie, kde signál odmeny nestačí na to, aby agent dokončil celkovú úlohu. Stáva sa to vtedy, keď sú odmeny riedke, čo znamená, že agent dostane vysokú odmenu až po úspešnom splnení úlohy, alebo negatívnu odmenu po výraznom neúspechu. V takýchto prípadoch musí vnútorná odmena dopĺňať signál odmeny z prostredia.

Vnútorná kontrola založená na odmene sa týka spôsobu ovládania robotov, ktorý zahŕňa poskytovanie interných odmien za vykonávanie určitých činností alebo dosahovanie určitých cieľov. V ideálnom prípade tieto odmeny nie sú explicitne definované programátorom, ale namiesto toho sa ich naučí robot prostredníctvom pokusov a omylov. Ide o to, že robot si na základe svojich skúseností vytvorí vlastné ciele a motiváciu, čo mu umožní vykonávať úlohy efektívne a ľahko sa prispôsobovať novým situáciám.

Existujúce techniky riadenia sú na druhej strane založené na osvedčených princípoch teórie riadenia, ktoré sa vyvíjali mnoho rokov. Tieto techniky zvyčajne zahŕňajú vytvorenie spätnej väzby medzi robotom a jeho prostredím, v ktorej robot meria svoj výkon a podľa toho prispôsobuje svoje činnosti.

Vnútorná kontrola založená na odmene je relatívne nový prístup, ktorý si v posledných rokoch získal popularitu, pričom existujúce kontrolné techniky boli vyvinuté počas mnohých desaťročí a sú široko používané v robotike.

Vnútorná odmena, ako je chápaná v tomto článku, je doplnková forma spätnej väzby, ktorá sprostredkúva ďalšie informácie o stave agenta v prostredí, ako je napríklad frekvencia navštívených stavov v stavovom priestore robota. Agent je nútený vyhľadávať a pozitívne reagovať na novoobjavené stavy, ktoré predstavujú anomálie v jeho pozorovaní. Čím častejšie sú štáty navštevované, stávajú sa menej anomálnymi, a preto odmena za ich návštevu časom klesá. Podobný prístup je použitý v [1], kde sa odmena zvyšuje, ak je predpokladaný ďalší stav odlišný od skutočného nasledujúceho stavu. Avšak v prípadoch, keď je prostredie dostatočne stochastické, agent nedokáže presne predpovedať ďalší stav, čo vedie k problému známemu ako Noisy TV [2]. Aby sa to prekonalo, hlboká neurónová sieť si môže zapamätať stavy z prostredia namiesto toho, aby sa učila dynamike prostredia.

V minulosti bolo navrhnutých mnoho metód na doplnenie odmien generovaných prostredím. Jeden z prvých prístupov, Counting by Density Model [3], používal pravdepodobnostný model na vyjadrenie metriky pseudo-počtu založenej na počtoch návštev štátu v stavovom priestore agenta. Iná metóda, Counting after Hashing, transformovala vysokorozmerné stavy na hash kódy a uložila ich frekvenciu návštev [4]. Prístupy založené na zvedavosti, ako napríklad Inteligentná adaptívna zvedavosť [5], Modul vnútornej zvedavosti [1] a Variačné

informácie maximalizujúce skúmanie [6], mali za cieľ naučiť sa dynamiku prostredia predpovedaním následných stavov. Tieto metódy však mali problém zvládnuť stochastické prostredia. Akcia zameraná na posilňovanie zamerania [7] a náhodná sieťová destilácia [2] použili na vyriešenie tohto problému hodnoty E (podobné hodnote Q) a zapamätanie. Náhodná sieťová destilácia využíva pár modelov, z ktorých jeden je len náhodne inicializovaný a druhý model sa učí predpovedať rovnaké vlastnosti, aké generuje náhodný model. Modul vnútornej odmeny založený na Generative Adversarial Network sa naučil distribúciu pozorovaných stavov a odmenil agenta za preskúmanie nepreskúmaných stavov [8]. Model GAN je trénovaný tak, aby produkoval vzorky, ktoré sa veľmi podobajú stavom reálneho sveta, pričom súčasne trénuje kodér na mapovanie pozorovaných stavov do priestoru latentného šumu. Následne generátor využíva priestor regenerovaného latentného šumu na generovanie nových pozorovaných stavov. Vnútorná odmena je určená výpočtom strednej štvorcovej chyby medzi pôvodnými stavmi a regenerovanými stavmi. Ak sú stavy neznáme, blok generátora ich nedokáže presne vygenerovať, čo vedie k zvýšeniu odchýlky. Výhodou tohto prístupu je, že diskriminátorovo hodnotenie predtým neviditeľných štátov sa stáva irelevantným. Pre úspech tejto metódy bolo rozhodujúce ladenie hyperparametrov [8]. Odlišný prístup, nazývaný Occupancy-Reward-Driven Exploration [9], bol aplikovaný v robotike na preskúmanie neprebádaných území v rámci štátneho priestoru. V tejto technike sa mapa obsadenosti využíva na získanie informácií o prostredí prostredníctvom senzorov, ako je laserový senzor. Mapa obsadenosti obsahuje hodnoty pravdepodobnosti obsadenosti, kde pravdepodobnosť prekážky predstavuje hodnota pravdepodobnosti. Vyššia hodnota je priradená oblastiam s vyššou úrovňou spoľahlivosti pri detekcii prekážok, zatiaľ čo neznáma oblasť má pravdepodobnosť 0,5 a pravdepodobnosť 0 znamená neprítomnosť prekážok. Odmena robota je potom určená počtom nových segmentov objavených na mape obsadenosti v každom časovom kroku. Tento prístup môže tiež zlepšiť energetickú účinnosť robota [9].

Vnútorné odmeny už boli použité v širokej škále aplikácií riadenia robotov pomocou hlbokej neurónovej siete. Jedným z príkladov je použitie vnútorných odmien na udržanie bezpečnej vzdialenosti medzi ramenom robota a ľudským operátorom [10]. V inej aplikácii boli vnútorné odmeny použité na uľahčenie spolupráce medzi viacerými autonómnymi robotmi. Dosiahlo sa to kombináciou učenia sa na základe učebných osnov, algoritmu PPO a hlbokého učenia sa konvolučnej neurónovej siete na spracovanie viackanálových vizuálnych vstupov [11]. V [12] bol obrázok použitý ako stavový priestor pre navigačnú stratégiu mobilných robotov riadenú zvedavosťou. Okrem toho bol v [13] implementovaný model doprednej dynamiky kontrastnej zvedavosti využívajúci efektívne vzorkovanie pre vizuálny vstup. Okrem toho boli vnútorné odmeny použité spolu s vonkajšími odmenami na simuláciu robotickej manipulácie s rukou v [14]. Ďalej boli v [15] použité vnútorné odmeny na pomoc robotom pri pripájaní nabíjacích staníc poskytnutím vizuálnej identifikácie. V [16] bola cieľom prekvapivej vnútornej odmeny založenej na penalizácii jemná manipulácia s objektom. Hlboká metóda založená na CNN našla uplatnenie aj pri výrobe drôtov [17].

Hlavnou inšpiráciou pre túto prácu bol prístup, ktorý riešil hry Atari prostredníctvom neurónových sietí [18]. Zatiaľ čo cieľ tejto práce riešiť problém riedkej odmeny [2], ako aj prístup v [18] majú spoločné podobnosti, stojí za zmienku, že v hrách Atari bol jediným dostupným signálom obraz obrazovky hry a hráč to neurobil. Neriadia priamo robota vybaveného rôznymi senzormi.

Na rozdiel od vyššie uvedených metód sa tento článok zameriava na využitie architektúry AutoEncoder ako detektora anomálií v signáli. Tento prístup dokáže

vypočítať vnútorné odmeny z anomálií, poskytujúc informácie o novom, predtým nepozorovanom stave v prostredí, kde sa agent pohybuje. Princípy použité v tejto štúdii sú priamo spojené s predchádzajúcim výskumom detekcie anomálií pri dolovaní údajov. Predchádzajúce použitie vnútornej odmeny za detekciu anomálií zahŕňalo iba označovanie súborov údajov alebo jednoduchšie úlohy, ktoré nesúviseli s riadením robota zo signálu[19-22]. Avšak v jednom prípade bola detekcia anomálií použitá na identifikáciu čiastkových cieľov pri riešení zložitého problému[23]. V tejto štúdii sa detekcia novosti používa na simulovaný pohyb robota. V neurónovej sieti s hlbokým učením je prístup AutoEncoder vhodný na zapamätanie si už preskúmaných stavov. Pre nové, predtým neviditeľné stavy, blok dekodéra nepresne zrekonštruuje redukovaný signál, ktorý možno merať ako detekciu novosti pre vnútornú odmenu. Preto vysoká vnútorná odmena pochádza z anomálie, ktorú blok dekodéra nedokáže správne transformovať do pôvodného stavu. Výsledkom je, že agent je nútený preskúmať prostredie, aby našiel nepreskúmané oblasti a štáty.

Cieľom agenta nebolo dokončiť zadanú úlohu (aj keď test skončil po dokončení úlohy), ale skôr zvýšiť svoje skóre skúmaním nových stavov v prostredí, s ktorými sa ešte nestretol. Tento prístup zabezpečil, že agent znovu nenavštívil už videné stavy a uprednostnil maximalizáciu skóre návštevou nepreskúmaných štátov.

Súčasné najmodernejšie techniky na optimalizáciu výkonu agentov využívajú kombináciu neepizodických a epizodických [24] prístupov, kde trénovanie neurónovej siete parametrizuje spektrum politík od vysoko prieskumných až po úplne vykorisťovateľské. Ďalší prístup zahŕňa potlačenie zabúdania v neurónovej sieti na zvýšenie výkonu [25].

Naše testy na súbore benchmarkových prostredí preukázali, že samotné anomálie môžu viesť agenta k úspešnému dokončeniu úlohy a že informácie z nich získané sú dostatočné na to, aby sa agent správne zblížil. Podľa našich najlepších vedomostí neexistuje žiadny prípad úspešného robotického navádzania iba prostredníctvom hľadania novosti pomocou vnútornej funkcie založenej na autokódovači.

2. Materiály a metódy

Nasledujúca časť je rozdelená do troch hlavných častí. Najprv je poskytnutý stručný popis použitej benchmarkovej sady testovacieho prostredia pre simulované roboty. Následne článok popisuje riadiacu architektúru robotov, ktorá využíva hlboké neurónové siete zahŕňajúce AutoEncoder, a poskytuje prehľad o tom, ako sa počítajú a aplikujú vnútorné odmeny.

2.1. Benchmark sada testovacích prostredí pre simulované roboty

Vybrané testovacie prostredia patria k štandardným úlohám riešeným metódami posilňovacieho učenia. Tieto prostredia majú spojitý stavový priestor a diskrétny akčný priestor. Problém LunarLander-v2 [26] predstavuje klasickú výzvu na optimalizáciu trajektórie rakety, ktorej cieľom je pristáť s lunárnym landerom na určenej pristávacej ploche. CartPole-v1 [27] je ďalší problém, ktorý zahŕňa vyváženie obráteného kyvadla na motorom poháňanom vozíku, kde cieľom je udržať vertikálnu polohu tyče pohybom vozíka doprava alebo doľava. Problém Acrobot-v1 [28-29] vyžaduje, aby agent dosiahol špecifickú výšku cieľa pomocou jednoduchého 1-kĺbového ramena. Systém pozostáva z dvoch prepojení, z ktorých jeden je pevný a druhý je k nemu pripojený. Počiatočná poloha má obe spojky visiace dole a cieľom je použiť silu na zmenu uhla spoja tak, aby sa voľný koniec spojov vykýval nad cieľovú výšku. V prostredí MountainCar-v0 [30] je cieľom jazdiť autom do kopca, aby ste dosiahli cieľový stav. Spočiatku je auto náhodne umiestnené na dne sínusového údolia a auto

môže byť zrýchlené v oboch smeroch, ale nie silne. Riešenie si vyžaduje využitie potenciálnej energie jazdou do protiľahlého kopca.

2.2. Architektúra automatického kódovania

AutoEncoder je jednou z techník využívaných v hlbokých neurónových sieťach na identifikáciu anomálií v signáloch robotických senzorov [31]. Tento prístup zahŕňa trénovanie na predtým pozorovaných stavoch z vyrovnávacej pamäte (RB) [32] a potom predpovedanie budúcich stavov na základe týchto pozorovaní. Výsledkom je, že AutoEncoder dokáže rýchlo identifikovať akékoľvek nové, predtým nepozorované stavy, čo umožňuje agentovi preskúmať svoje prostredie a získať prístup k nenavštíveným stavom v rámci svojho stavového priestoru.

Detekcia anomálií sa spolieha na kombináciu prieskumu a využívania, pričom využíva učenie metrík založené na AutoEncoder na meranie chyby rekonštrukcie v predikcii agenta. Stratová funkcia AutoEncoder sa vypočíta ako bežne používaná stredná štvorcová chyba:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}[state_i - AE(state)_i]^2, \quad (1)$$

kde n predstavuje počet vlastností v stavovom priestore agenta, stavi predstavuje vlastnosť stavu, ktorú agent získava z prostredia, a AEi predstavuje výstup modelu AutoEncoder.

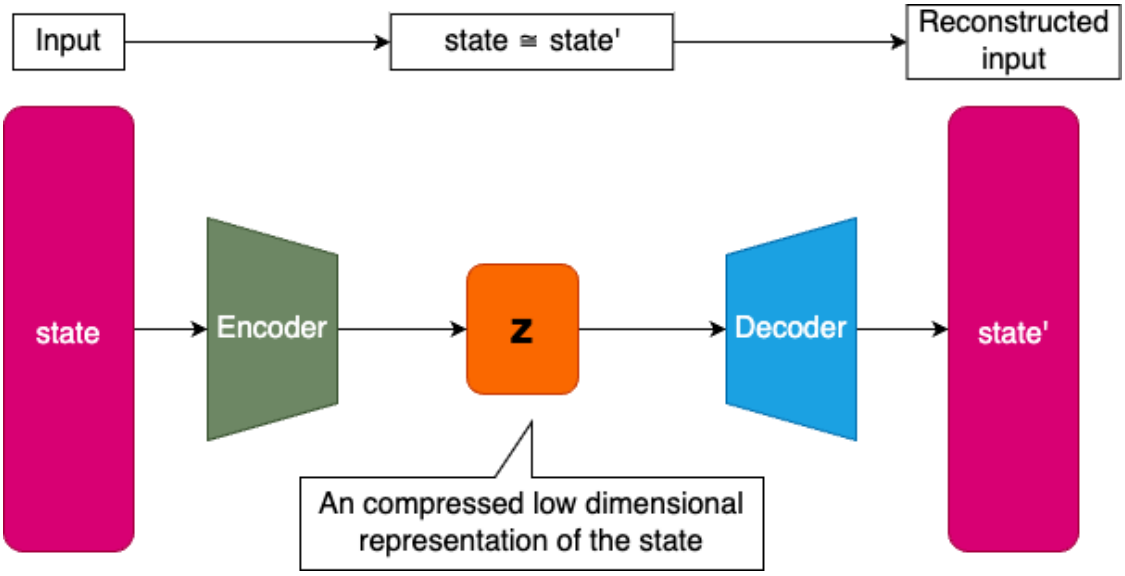
Vnútorná odmena sa vypočíta ako chyba rekonštrukcie:

$$\mathrm{err}(state) = \sum_{i=1}^{n}[state_i - AE(state)_i]^2, \quad (2)$$

$$\text{Intrinsic reward} = ReLU6(\frac{\mathrm{err}(state)-\mu_e}{\sigma_e}), \quad (3)$$

kdeμ_ea$\sigma_e$sú bežiaci priemer a štandardná odchýlka pre err(state). Pri výpočte obvyklej chyby rekonštrukcie [33] sa priemerovanie prvkov neberie do úvahy. Je to preto, že ak existuje veľa senzorov a väčšina z nich meria typické hodnoty, zatiaľ čo iba jeden senzor vykazuje významný rozdiel, použitie priemerovania by mohlo potenciálne maskovať tento signál, čo by malo za následok stratu vnútornej odmeny.

Na zabezpečenie numerickej stability sa priemerná odmena vypočítava výpočtom priebežného priemeru a štandardnej odchýlky vnútornej odmeny. Táto priemerná odmena odráža posun, čo je priemerná chyba automatického kódovača pri rekonštrukcii známeho signálu, a musí sa odpočítať od vnútornej odmeny. Výsledkom odčítania však môžu byť záporné hodnoty, ktoré nie sú žiaduce, pretože môžu penalizovať aj dobre známe stavy. Aby sa predišlo negatívnym odmenám, na normalizovanú vnútornú odmenu sa aplikuje funkcia Rectified Linear Unit 6 (ReLU-6), ktorá zaisťuje, že odmeny sú vždy pozitívne a zabraňuje tomu, aby sa vnútorná odmena stala príliš veľkou [34]. Účelom tejto odmeny je skôr zatraktívniť nenavštívené štáty, než trestať časté návštevy štátu. Napríklad v labyrinte by opakovaný štart po tej istej ceste viedol k negatívnej odmene, čo by agenta mohlo viesť k tomu, že bude túto cestu považovať za nepriaznivú. Priradením odmeny 0 agent považuje cestu za neutrálnu. Stupnica vnútornej odmeny sa riadi štandardnou odchýlkou 1. Architektúra AutoEncoder na detekciu anomálií je znázornená na obr. 1, ktorý ako vstup preberá stav z prostredia a na výstupe predpovedá jeho rekonštrukciu.



Postava 1. AutoEncoder model s veľkosťou vrstiev udávajúcou počet neurónov vo vrstvách, kde najmenšia vrstva z v strede zodpovedá reprezentácii nízkorozmerného latentného priestoru.

Blok kódovača modelu sa skladá zo sekvencie plne prepojených vrstiev, pričom počet neurónov v každej vrstve postupne klesá smerom k latentnému priestoru. Najmenšia vrstva v celom modeli zodpovedá reprezentácii latentného priestoru kódovaného stavu z. Blok dekodéra funguje opačným spôsobom a zvyšuje počet neurónov v každej plne prepojenej vrstve smerom k výstupnej vrstve. Výstupná vrstva obsahuje toľko neurónov, koľko je prvkov v stavovom priestore agenta. Nelinearita aktivačnej funkcie exponenciálnej lineárnej jednotky (ELU) sa používa v skrytých vrstvách na uľahčenie rýchleho a presného učenia hlbokej neurónovej siete [35]. V porovnaní s typicky používanou funkciou aktivácie ReLU má ELU výhodu pri riešení problému umierajúceho neurónu [36]. Váhy modelov sú inicializované pomocou ortogonálneho inicializátora so ziskom nastaveným ako druhá odmocnina z 2 [37]. Výstupná vrstva sa aktivuje pomocou lineárnej funkcie, ktorá umožňuje neobmedzený rozsah výstupných hodnôt a umožňuje aplikáciu AutoEncoder na rôzne typy senzorov v rámci jedného stavového priestoru. Podobne aktivačná funkcia vrstvy latentného priestoru z je tiež lineárna, aby sa zachovala vlastnosť neobmedzeného intervalu aj po kompresii stavového priestoru, pretože táto vrstva je určená skôr na zníženie počtu rozmerov než na účelové obmedzenie intervalu komprimovaných hodnôt.

2.3. Aplikácia vnútornej odmeny architektúrou AutoEncoder

Navádzanie robota smerom k anomáliám signálu sa dá použiť aj na efektívne prehľadávanie bludiska, čím sa minimalizujú návštevy robota v už preskúmaných oblastiach a navádza ho do nepreskúmaných častí. Tento princíp možno prirovnať k tomu, že hráč si pomocou priadze vyznačí cestu bludiskom. V tomto prípade model zapamätania AutoEncoder predstavuje priadzu s nízkou chybou rekonštrukcie, ktorá indikuje, že priadza sa tu rozvinula, a vysokou chybou, ktorá naznačuje, že nie. Tento princíp možno použiť aj na zložitejšie problémy, ako je opísané v časti 2.1, ako je pristátie na lunárnom pristávacom module, vyváženie obráteného kyvadla, výkyv jednoduchého 1-kĺbového ramena alebo jazda autom do kopca s použitím zrýchlenia idúceho dolu oproti. kopec.

Pri kombinovaní vnútorných a vonkajších odmien získaných po vykonaní akcie logika v tabuľke 1 predpokladá, že oba signály spadajú do rozsahu [0, 1].

Stôl 1. Logika kombinovania vnútornej a vonkajšej odmeny bola objasnená na príkladoch kombinácií extrémnych hodnôt a novosti súčasného stavu vo vzťahu k dokončeniu úlohy, ako je ďalej znázornené na obrázkoch 2, 4, 6 a 8.

Vnútorná odmena	Vonkajšia odmena	Výsledok
0	0	Často sa opakujúci stav vedúci k strate
1	0	Nový neviditeľný stav vedúci k strate
0	1	Často sa opakujúci stav vedúci k výhre
1	1	Nový neviditeľný stav vedúci k víťazstvu

Existujú dva možné prístupy na zlúčenie oboch signálov odmeny. Prvým a jednoduchším prístupom je sčítanie vnútornej odmeny a vonkajšej odmeny, kde môžete regulovať váhu vnútornej odmeny úpravou parametra škálovania.β, kdeβ> 0 [24].

$$r_t = r_t^{extrinsic} + \beta\, r_t^{intrinsic}, \tag{4}$$

Pokročilejšia metóda zahŕňa zlúčenie odmien na základe ich príslušných hodnôt Q. To znamená generovanie odlišných hodnôt Q pre vnútorné a vonkajšie odmeny, čo umožňuje použitie špecifických diskontných faktorov (γ) pre každý prediktor [24]. Tento prístup sa ukázal ako výhodný v štúdii Random Network Destillation, ktorá preukázala výhodu použitia vyššejγpre vonkajšie hodnoty Q ako pre vnútorné hodnoty Q.

$$Q(s,a) = Q\left(s,a,\theta^{extrinsic}\right) + \beta\, Q(s,a,\theta^{intrinsic}), \tag{5}$$

Parametreθ^{vonkajších}aθ^{vnútorný}reprezentujú Q parametre modelu učenia sa vonkajšej odmeny a parametre Q modelu učenia sa vnútornej odmeny. Miera vplyvu vnútornej odmeny môže byť opäť kontrolovaná pomocouβparameter.

Avšak kombinácia vnútorných a vonkajších odmien, ako je znázornená v rovniciach (4) a (5), nebola použitá v nasledujúcich výpočtoch. Je to preto, že účelom bolo demonštrovať, že samotná vnútorná odmena môže viesť proces učenia agenta na splnenie úlohy, a to aj pri absencii vonkajšej odmeny.

Táto nová metóda vnútorných odmien nenahrádza tradičné prieskumné stratégie, ako je Epsilon-chtivý [38] alebo Boltzmannov prieskum [39]. Namiesto toho slúži ako doplnok k týmto metódam, ktoré sa primárne zameriavajú na skúmanie akčného priestoru agenta. Ich úlohou je nájsť rovnováhu medzi čisto náhodným výberom akcií a výberom akcií na základe predpovedí, ktoré robí neurónová sieť zo súčasného stavu. Na druhej strane metóda vnútornej odmeny vedie agenta pri skúmaní stavového priestoru prostredia so zameraním na frekvenciu návštev jednotlivých štátov. Skombinovaním oboch prístupov v budúcnosti sa očakáva, že agent získa úplnú kontrolu nad prehľadávaním prostredia.

Rovnako ako iné metódy, ako napríklad Directed Outreach Reinforcement Action-Selection alebo Random Network Distillation, AutoEncoder neresetuje svoje znalosti o návšteve stavu medzi epizódami, čo z neho robí neepizodickú [40] vnútornú metódu odmeňovania. Metóda si zapamätá frekvenciu štátnej návštevy vo viacerých epizódach.

3. Výsledky

Agent použitý v tejto štúdii bol reprezentovaný algoritmom Dueling Deep Q Network (DQN) [41], ktorý bol vybraný kvôli jeho vhodnosti pre diskrétne akčné priestory, ktoré zahŕňajú všetky testovacie prostredia použité v tomto článku. DQN preukázal úspech v hrách Atari [41], a preto sa očakáva, že bude schopný riešiť úlohy riadenia robotov. Agent využíva Boltzmannov prieskum na

vyhľadávanie akčného priestoru (na rozdiel od nenásytnej politiky), pričom teplotný parameter sa časom lineárne znižuje s použitím rovnakej hodnoty poklesu, až kým nedosiahne prednastavenú minimálnu hodnotu teploty. Experimenty odhalili, že rozsiahle vyhľadávanie je výhodné v porovnaní s chamtivou politikou založenou na naučených hodnotách Q, pretože umožňuje vnútornej odmene dosiahnuť vysoké hodnoty a stavy odmeňovania, ktoré predtým neboli pozorované. Potom môže štandardizovaný stavový priestor [42] uľahčiť konvergenciu neurónovej siete rýchlo a presne. Vnútornú odmenu možno prečítať z RB alebo vypočítať priamo počas aktualizácie DQN a dosadiť do Bellmanovej rovnice [43].

Tabuľka 2 Optimalizované hyperparametre pre agenta Dueling DQN.

názov	**Popis**	**Hodnota**
Epochy	Počet tréningových epizód	2 000
Veľkosť vyrovnávacej pamäte	Kapacita vyrovnávacej pamäte pre prehrávanie skúseností	1 000 000
Min. teplota	Minimálna hodnota teploty pre Boltzmannov prieskum	0,01
Init. teplota	Maximálna hodnota teploty pre Boltzmannov prieskum	1,0
Teplota rozpadu	Hodnota zníženia teploty	1 * 10-5
Veľkosť dávky	Počet vzoriek aplikovaných počas tréningu naraz	256
Miera učenia	Miera učenia pre tréningový proces	3 * 10-4
Globálna klipnorma	Orezanie aplikované globálne na prechody	1,0
τ	Hodnota aktualizácie mäkkého cieľa pre kopírovanie originálu do cieľového DQN	0,01
γ	Diskontný faktor pre Bellmanovu rovnicu	0,99
Počet neurónov v DQN	Počet neurónov pre každú skrytú vrstvu	512, 256
Počet neurónov v AutoEncoder	Počet neurónov pre každú skrytú vrstvu	128, 64, 32, 16, LS, 16, 32, 64, 128

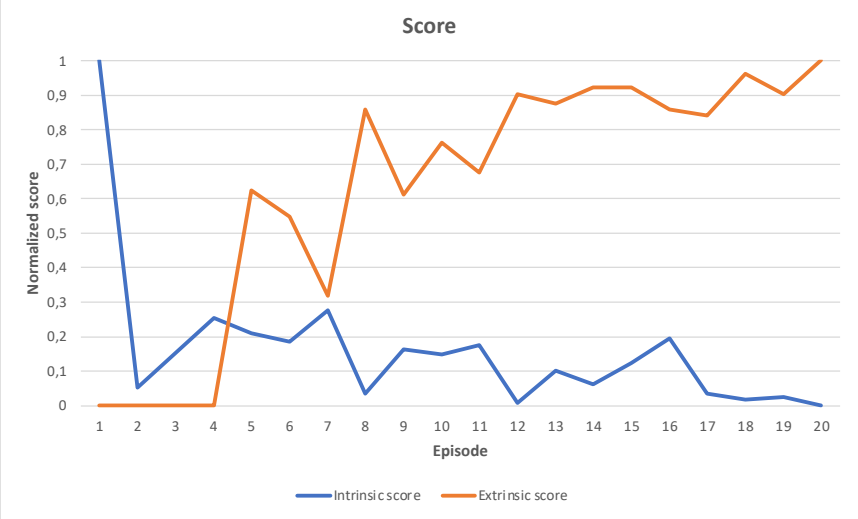
Tabuľka 2 poskytuje úplný zoznam hyperparametrov používaných v prostrediach Acrobot-v1, CartPole-v1, LunarLander-v2 a MountainCar-v0. Tieto hyperparametre boli doladené pomocou nástroja W&B Sweeps [44], kde bolo vykonané náhodné vyhľadávanie na 45 kombináciách hodnôt okolo optimálnych hodnôt. Optimálne hodnoty boli identifikované ako tie, ktoré umožnili úspešné dokončenie úloh v uvedených prostrediach. V tabuľke 2 skratka LS označuje počet neurónov pridelených pre latentný priestor, ktorý bol experimentálne určený ako polovičný počet znakov v stavovom priestore. V budúcnosti sa očakáva zrýchlenie tréningového procesu prostredníctvom distribuovanej paralelizácie [45-47].

Kumulatívne vnútorné a vonkajšie odmeny získané agentom počas jednej epizódy sú reprezentované vnútornými a vonkajšími skóre znázornenými na nasledujúcich obrázkoch. Epizódu možno definovať ako postupnosť časových krokov, po ktorých agent buď splní úlohu, alebo napríklad zničí robota. Ak sa prekročí maximálny počet krokov na dokončenie konkrétnej úlohy, epizóda sa aj tak dostane k externému koncu, ale reťaz Markovovho rozhodovacieho procesu [48] bude pokračovať vďaka rozlišovaniu medzi „signál ukončený“ a „signál skrátený“. „podmienky. Podmienka „signál ukončený“ sa vzťahuje na epizódu, ktorá sa skončí po dosiahnutí terminálneho stavu, ako je definovaný prostredím, zatiaľ čo podmienka „skrátený signál“ sa vzťahuje na epizódu, ktorá sa skončí po externe stanovenom časovom limite na vyriešenie úlohy [49].

V prípadoch, keď latentný priestor z pozostával z viac ako dvoch prvkov, bola na vizualizáciu údajov v 2D priestore použitá technika redukcie t-

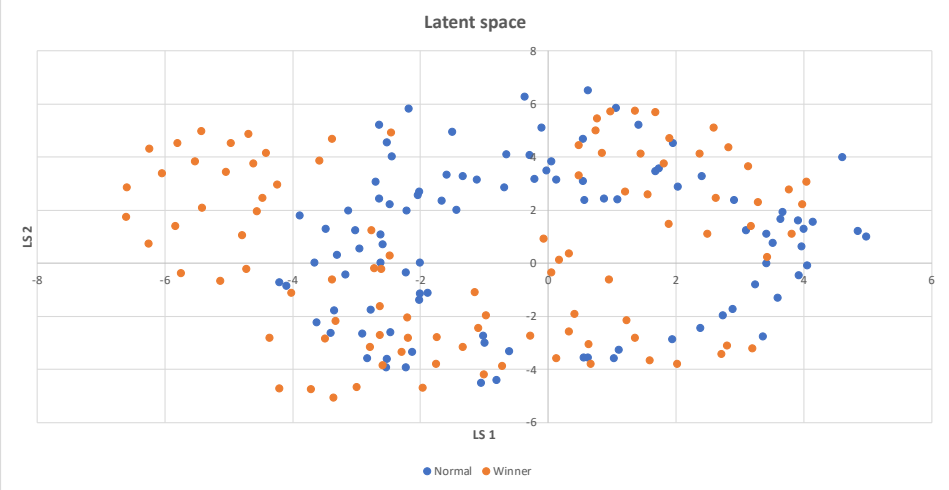
SNE [50]. Keď sa však v latentnom priestore nachádzal iba jeden prvok, všetky body boli umiestnené na osi y s rovnakou hodnotou 1. V prípadoch, keď existovali iba dva prvky, bol latentný priestor priamo zobrazený na grafe. Vo výsledných grafoch latentného priestoru boli body kategorizované na základe toho, či komprimované stavy viedli priamo k víťazstvu, čím sa stali súčasťou víťaznej epizódy, alebo či boli potrebné na preskúmanie prostredia, ale neviedli k rýchlemu víťazstvu.Zo zistení uvedených nižšie možno odvodiť dva potenciálne scenáre. Prvý scenár zahŕňa vznik dvoch odlišných zhlukov, keď agent vyhrá aj v neviditeľných stavoch. Druhý scenár nastáva, keď špecifická kombinácia navštívených stavov vedie k víťazstvu agenta a víťazné aj porazené stavy sú sústredené v jednom zhluku. Keď je ponúknutá vysoká vnútorná odmena, agent skúma úplne nové stavy, zatiaľ čo aj nízka vnútorná odmena vyvoláva prieskum akčného priestoru, čo v konečnom dôsledku vedie k dokončeniu úlohy vyvolanej organizáciou predtým navštívených štátov. V dôsledku toho je kľúčová kombinácia prehľadávania na úrovni stavu a prehľadávania akčného priestoru.

Novinka metóda vnútornej odmeny umožnila naučiť sa všetky testované úlohy bez potreby externých odmien z okolia. Vonkajšie skóre prezentované na obrázkoch 2, 4, 6 a 8 bolo použité výlučne na hodnotenie výkonu úlohy a nemalo žiadny vplyv na proces učenia agenta.



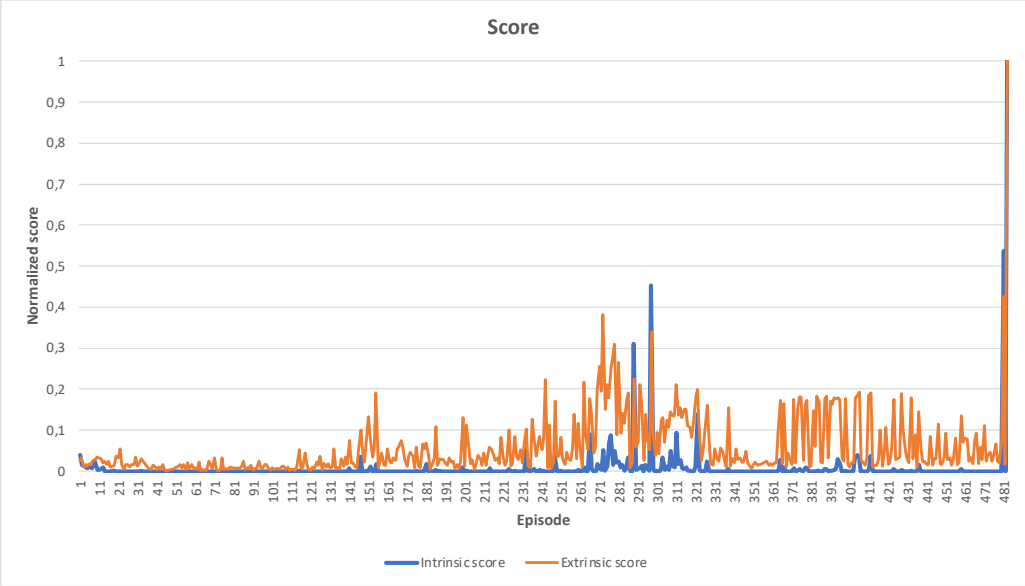
Obrázok 2 Skóre učenia Acrobot-v1 naznačuje relatívne stabilné zlepšovanie vonkajšieho skóre po dokončení úlohy, čo je sprevádzané znížením skúmania noviniek. To znamená postupné dolaďovanie parametrov.

Na obrázku 2 je uvedené porovnanie medzi normalizovaným vnútorným a vonkajším skóre v prostredí Acrobot-v1. Toto porovnanie odhaľuje, že ako agent počas učenia skúma prostredie, vnútorné skóre sa v priebehu času postupne znižuje, zatiaľ čo vonkajšie skóre sa zvyšuje. Toto zvýšenie vonkajšieho skóre je spôsobené tým, že agent skúma prostredie, až kým nedosiahne prah skóre potrebný na dokončenie úlohy, ktorá zahŕňa dosiahnutie určitého výškového cieľa svojou pažou. Agent skenuje viacero výškových úrovní, kým konečne nedosiahne cieľ. Vzťah medzi vonkajšími a vnútornými skóre demonštruje súhru medzi dynamikou vyhľadávania a komplexnosťou prostredia. Vysoké vonkajšie skóre naznačuje, že úloha je blízko dokončenia, zatiaľ čo nízke vnútorné skóre naznačuje, že na dokončenie úlohy sú potrebné len menšie úpravy a vzdelávacie prostredie nepredstavuje žiadne náhle alebo neočakávané problémy.

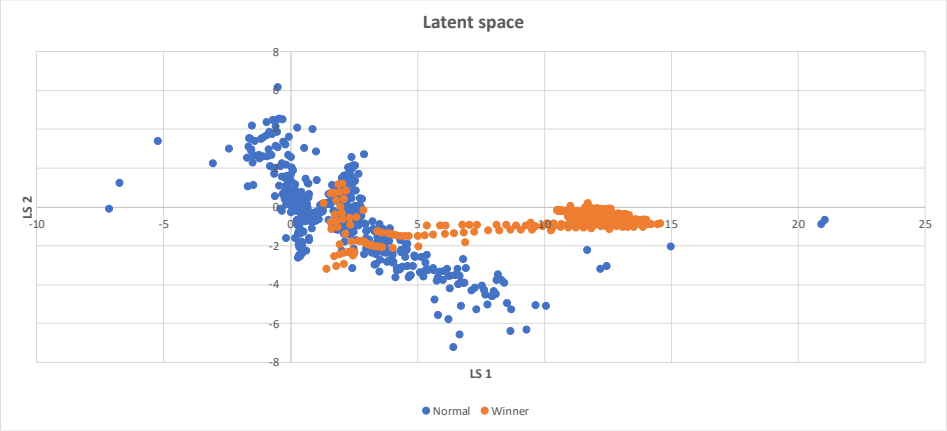


Obrázok 3. Latentný priestor prostredia Acrobot-v1, kde sa vľavo nachádza zreteľný zhluk pozostávajúci z niektorých víťazných stavov.

Na obrázku 3 je analyzované rozloženie víťazných bodov v latentnom priestore. Analýza odhaľuje, že väčšina výherných bodov je rovnomerne rozložená v latentnom priestore aj v bodoch bez výhry, čo sú často opakujúce sa stavy, ktoré okamžite nevedú k výhre. Existuje však jeden malý zhluk víťazných bodov, ktorý sa líši od nevýherných bodov.



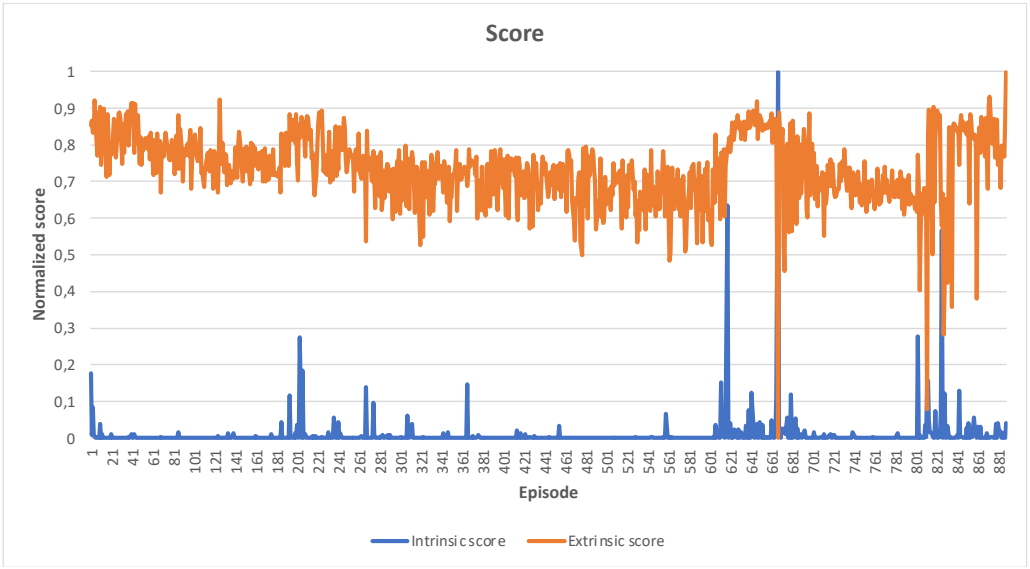
Obrázok 4. Skóre učenia CartPole-v1, kde vnútorné aj vonkajšie značne kolíšu, čo naznačuje zložitosť vyhľadávacieho priestoru.



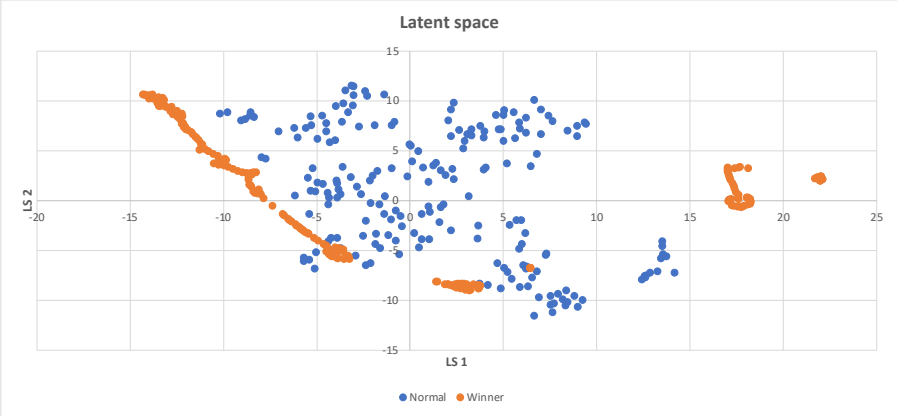
Obrázok 5.V latentnom priestore prostredia CartPole-v1 sa víťazné stavy zvyčajne líšia od normálnych, čo znamená, že objavenie nových stavov s väčšou pravdepodobnosťou povedie k úspešnému dokončeniu úlohy.

V prostredí CartPole-v1 ukazuje obrázok 4 porovnanie medzi normalizovaným vnútorným a vonkajším skóre. Porovnanie ukazuje, že agent spočiatku skúmal blízke okolie, čo minimalizovalo vnútorné skóre. Ako agent postupoval, narazil na nové stavy vedúce k zvýšeniu vnútorného skóre. Nakoniec agent objavil stavy, ktoré viedli k výhre, kde vonkajšie aj vnútorné skóre boli vysoké. Absencia zreteľného vzoru v skóre naznačuje, že prostredie je veľmi zložité a vykonávanie postupných zmien nie je efektívne. Namiesto toho sa pokrok smerom k (takmer) úplným riešeniam dosahuje skúmaním predtým neprebádaných oblastí štátneho priestoru.

Obrázok 5 ukazuje, že výherné body sú husto rozmiestnené aj mimo nevýherných bodov, čo znamená, že tieto body predstavujú nepreskúmané stavy.



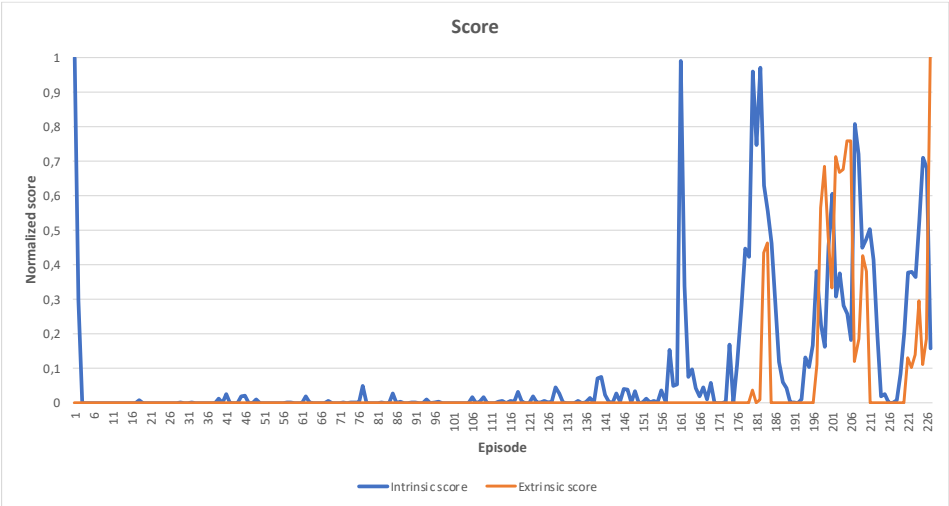
Obrázok 6. Skóre učenia LunarLander-v2, vysoké vnútorné hodnoty spojené s nízkymi vonkajšími hodnotami okolo epizód 661 a 800 naznačujú zlyhanie kontroly počas skúmania neznámych stavov.



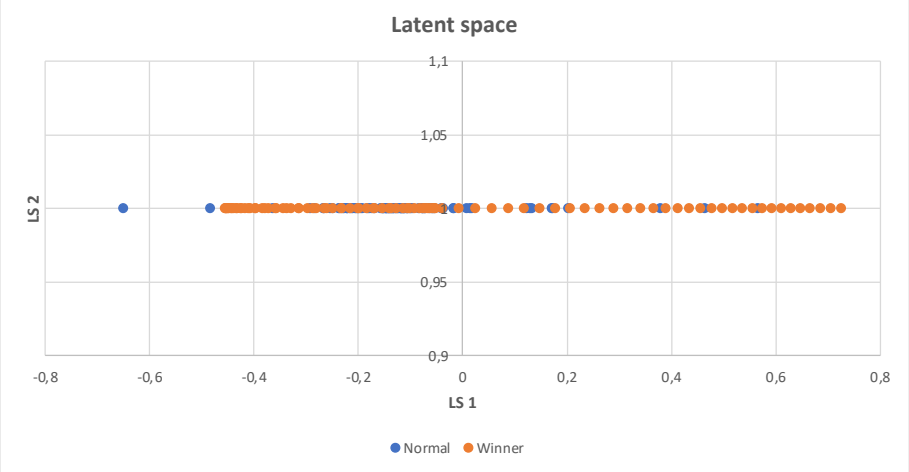
Obrázok 7. Latentný priestor prostredia LunarLander-v2, víťazné stavy a normálne stavy sú len zriedka blízko, čo znamená, že nepreskúmané stavy sú rozhodujúce pre splnenie úlohy.

V prostredí LunarLander-v2 obrázok 6 ilustruje porovnanie medzi vnútorným a vonkajším skóre, ktoré bolo normalizované. Vnútorné skóre sa časom znižuje, keď agent na začiatku učenia skúma svoje bezprostredné okolie, zatiaľ čo vonkajšie skóre zostáva nezmenené, čo nenaznačuje žiadnu výhodu zo skenovaných stavov. Keď agent skúma viac stavov, dochádza k miernemu zvýšeniu vonkajšieho skóre, ale vysoké vnútorné skóre sa udeľuje aj za stavy vedúce k jasnej strate, čo nie je žiaduce. Keďže však tieto štáty boli málo navštevované, stále sa to dalo. Na konci procesu učenia agent vyriešil úlohu prekročením prahu skóre, ale nedosiahol vysoké vnútorné skóre, čo naznačuje, že boli pozorované často sa opakujúce stavy vedúce k výhre. Porovnanie medzi vonkajšou a vnútornou odmenou naznačuje, že aj keď agent mohol mať pri skúmaní neprebádaných území nejaké významné zlyhania v kontrole, dokázal dosiahnuť uspokojivé výsledky, keď bola vnútorná odmena vysoká a vonkajšia odmena nízka.

Na obrázku 7 sú víťazné body zoskupené oddelene okolo nevýherných bodov. Víťazné body väčšinou pozostávajú z predtým nenavštívených štátov, čo naznačuje, že prieskum agenta bol účinný pri objavovaní nových štátov.



Obrázok 8. Skóre učenia MountainCar-v0, ako proces učenia postupuje, novo preskúmané stavy blízko dokončenia úlohy prinášajú vysoké skóre pre oba typy meraní.



Obrázok 9. Latentný priestor prostredia MountainCar-v0, víťazný a normálny stav sa výrazne prekrývajú, čo naznačuje, že dokončenie úlohy si pravdepodobne nebude vyžadovať prudké zmeny parametrov.

V prostredí MountainCar-v0 ukazuje obrázok 8 porovnanie medzi normalizovaným vnútorným a vonkajším skóre. Agent zrejme od začiatku procesu učenia skenoval svoje bezprostredné okolie. Ako čas pokročil, agent začal odhaľovať nové stavy, čo viedlo k vysokému vnútornému skóre a vonkajšie skóre sa postupne zvyšovalo. Toto správanie demonštruje, ako sa auto postupne približovalo k cieľovému stavu na vrchole pravého kopca.

Obrázok 9 ilustruje, že víťazné aj nepriame body sú rovnomerne rozložené v latentnom priestore. Často opakované stavy vedú k víťazstvu.

Tabuľka 3. Porovnanie výkonu DQN s použitím výlučne vonkajšej odmeny a vnútornej odmeny. Výsledky sú hodnotené na základe skóre, kde pozitívnejšie skóre znamená lepší výkon. Dokončenie úlohy vyžaduje prekročenie prahu skóre v poslednom stĺpci a oba prístupy ho dosiahli pre všetky prostredia. Najlepšie výsledky sú zvýraznené tučným písmom.

Životné prostredie	skóre (iba s vonkajšou odmenou)	skóre (Tento papier)	Hranica skóre
MountainCar-v0	-194,95 ± 8,48 [51]	**-96,684 ± 7,028**	-110
Acrobot-v1	-91,54 ± 7,20 [51]	**-86,12 ± 4,604**	-100
CartPole-v1	488,69 ± 16,11 [51]	**499 483 ± 3,05**	475
LunarLander-v2	**280,22 ± 13,03**[52]	234 881 ± 33,86	200

Tabuľka 3 uvádza porovnanie skóre medzi prístupom použitým v tomto dokumente, ktorý sa zameriaval výlučne na vnútornú odmenu, a konvenčnými výsledkami DQN získanými z prostredia CleanRL [51] a prostredia Stable-Baselines3 [53]. Výsledky v tejto štúdii boli získané z priemeru 100 nezávislých po sebe nasledujúcich experimentov a do štatistickej analýzy boli zaradené iba úspešné behy, ktoré viedli k vyriešeniu danej úlohy. Experiment sa vždy skončil, keď agent prekročil vopred stanovený prah skóre [54], uvedený v treťom stĺpci tabuľky 3. Ak by agent pokračoval v hľadaní a učení sa aj po tomto bode, výsledky by sa zhoršili, pretože víťazné stavy by boli označené ako často navštevovaný a agent by preskúmal iné podradné smery. Výsledky ukazujú, že samotná

vnútorná odmena môže byť použitá ako metóda vyhľadávania; agent však nemusí úplne pochopiť svoj skutočný cieľ v prostredí.

Jednou nevýhodou spoliehania sa výlučne na vnútornú odmenu bolo, že v niektorých prípadoch sa agent zasekol, ak vyčerpal všetky možnosti vo svojom bezprostrednom okolí, čo viedlo k priemernej vnútornej odmene 0, z ktorej sa potom stala riedka odmena.

Napriek týmto potenciálnym obmedzeniam výsledky v tabuľke 3 naznačujú, že v troch zo štyroch prostredí sa pri použití samotnej vnútornej odmeny dosiahli lepšie výsledky ako pri spoliehaní sa výlučne na vonkajšiu odmenu, ako ukazujú tučné skóre v tabuľke. Vo štvrtom prostredí, kde boli výsledky vnútornej odmeny o niečo horšie, bol problém stále úspešne vyriešený.

Agenti používajúci výlučne vnútornú odmenu vo všetkých prostrediach dokončili úlohy. Ak by sa parametre z predchádzajúcej víťaznej epizódy použili pre agenta v novej inštancii, bol by schopný úspešne dokončiť úlohu, ako keby sa naučil z externej odmeny. Výhradou však je, že bez toho, aby používateľ nastavil prah skóre alebo nejaký iný indikátor, spoliehanie sa výlučne na vnútornú odmenu by neposkytlo žiadnu indikáciu dokončenia úlohy a agent by pokračoval v hľadaní.

3.1. Kedy vypočítať vnútornú odmenu

Existujú dva možné prístupy k výpočtu vnútorných odmien. Jeden prístup zahŕňa vyjadrenie vnútornej odmeny po tom, čo agent získa stav z prostredia, a jeho uloženie do vyrovnávacej pamäte pre prehrávanie skúseností (RB). V tomto prípade je AutoEncoder trénovaný na pozorovanom stave z RB, zatiaľ čo predikcia vnútornej odmeny sa robí na základe aktuálneho stavu, ktorý používa agent v prostredí na predpovedanie akcie. To však môže viesť k nevhodnosti, keď vnútorné odmeny naskúšané z minulosti už nemusia zodpovedať aktuálnej frekvencii návštev daného štátu a nepresne ovplyvnia politiku agenta počas aktualizácií modelu DQN. Prístup v "Never Give Up" [55] tiež ukladá vnútornú odmenu v RB.

Alternatívne je možné vnútorné odmeny vypočítať počas aktualizácie modelu DQN bez okamžitého uloženia odmeny. Keďže stavy sú zvyčajne súčasťou uložených skúseností, vnútorné odmeny možno vypočítať na základe aktuálnych návštev stavu v agentovom stavovom priestore. V tomto prípade sa AutoEncoder učí nastatetz RB a predikcia vnútornej odmeny sa vykonáva zstatet+1, ktorá je tiež súčasťou RB. Na porovnanie oboch možností boli vygenerované štatistiky v rôznych prostrediach so 100 spusteniami v každom prípade.

Tabuľka 4.Úspešné riešenie problémov vo všetkých testovaných prostrediach pre rôzne metódy aktualizácie vnútorného skóre. Žiadnu metódu nemožno považovať za všeobecne lepšiu, pretože účinnosť každej metódy závisí od konkrétneho prostredia.

Životné prostredie	**Metóda**	**Percentuálna úspešnosť**
Acrobot-v1	Uložené v RB	100
	Vypočítané počas aktualizácie modelu DQN	100
CartPole-v1	Uložené v RB	**75**
	Vypočítané počas aktualizácie modelu DQN	60
LunarLander-v2	Uložené v RB	11
	Vypočítané počas aktualizácie modelu DQN	**14**

MountainCar-v0	Uložené v RB	61
	Vypočítané počas aktualizácie modelu DQN	**98**

Podľa tabuľky 4 je optimálne načasovanie výpočtu vnútornej odmeny závislé od typu prostredia a konkrétnej vykonávanej úlohy. Pre vybrané prostredia MountainCar-v0 a LunarLander-v2 vykazovali konzistentnejšiu cieľovú konvergenciu, keď bola vnútorná odmena vypočítaná počas aktualizácií modelu DQN pomocou uložených stavov v RB. Na druhej strane, pre prostredie CartPole-v1 bolo výhodnejšie vnútorné odmeny ukladať do RB a následne ich vzorkovať počas aktualizácií modelu DQN. Napriek ťažkostiam s učením kvôli riedkym vonkajším odmenám, MountainCar-v0 dosiahol vysokú úspešnosť úloh až 98 percent, keď boli využité vnútorné odmeny.

4. Diskusia

V tejto štúdii nebola použitá vonkajšia odmena. Namiesto toho dokument demonštruje účinnosť vnútorných odmien zameraných na novinky za posilnenie učenia, čím sa vonkajšie odmeny stávajú nepotrebnými. Napriek absencii vonkajšej odmeny bola úloha úspešne dokončená, väčšina z nich dokonca lepšie, ako keď boli vonkajšie odmeny testované v iných štúdiách [51,52]. Napriek tomu sa odporúča, aby budúce učenie robotov zahŕňalo spojenie vnútorných a vonkajších odmien pre optimálne výsledky. Spojenie vnútorného riadenia robota založeného na odmene s existujúcimi riadiacimi technikami opísanými vyššie v rovniciach (4) a (5) kombinuje dva rôzne prístupy k riadeniu robotov spôsobom, ktorý využíva silné stránky každého prístupu. Týmto spôsobom je robot schopný využiť stabilitu a spoľahlivosť existujúcich kontrolných techník a zároveň profitovať z flexibility a adaptability vnútornej kontroly založenej na odmene. V budúcnosti možno preskúmať pokročilejšie metódy kombinovania rôznych typov odmien.

Tu navrhovaná metóda umožňuje robotovi preskúmať svoje bezprostredné okolie počas autonómneho vyhľadávania detekciou anomálií, ktoré ho vedú smerom k neprebádanému terénu. Napríklad rover na Marse, ktorému chýba komunikácia so Zemou v reálnom čase, môže použiť navrhovanú metódu namiesto v súčasnosti používaného plánovacieho prístupu založeného na sieti. V súčasnosti používaný plánovač generuje mapu terénu pomocou stereo kamier a vytvára cestu na mriežke pomocou plánovacieho algoritmu Field D* [56]. Tento algoritmus priraďuje náklady každej bunke mriežky a generuje cestu, ktorá tieto náklady minimalizuje. Tento dokument však navrhuje iný prístup, ktorý využíva neurónovú sieť na generovanie ciest založených na predtým neviditeľných stavoch. Cieľom je vybrať si cestu, ktorá poskytuje najnovšie informácie pri skúmaní povrchu Marsu, namiesto výberu najoptimálnejšej cesty.

Ďalším príkladom, kde úspech robota závisí od objavovania neprebádaného terénu, je archeologický prieskum morského dna alebo pevniny vo väčších hĺbkach. Na rýchlu navigáciu v prázdnych oblastiach a na nájdenie ďalšej priechodnej trasy v zložitých prostrediach sa zvyčajne používajú vyhľadávacie techniky, ako je náhodný stromový algoritmus (RRT) alebo RRT* [57]. Na rozdiel od toho sa tento článok zameriava na použitie čistej neurónovej siete ako pamäte namiesto generovania stromu. Tento prístup môže byť užitočný aj pri reakcii na prírodné katastrofy tým, že automatizuje vyhľadávanie v zložitých prostrediach a navádza robota k cieľu alebo k východu. V každom z týchto scenárov je potrebné vnútornú odmenu doplniť o informácie o tom, čo má robot hľadať.

Vnútorná odmena môže pomôcť riadiť správanie robota aj inými spôsobmi. Robot by mohol byť napríklad odmenený za dosiahnutie určitej úrovne presnosti vo svojich pohyboch alebo za povzbudenie robota, aby vykonával úlohu

rýchlejšie alebo spotreboval menej energie. V úvode sa uvádzalo, že takéto aplikácie vnútornej odmeny sa predtým úspešne používali v iných kontextoch bez použitia autokódera, ktorý sa používa v tejto štúdii. Navrhuje sa, že použitie automatického kódovača pre vnútornú odmenu môže byť vhodnejšie pre zložitejšie úlohy, ako je vyhľadávanie noviniek.

5. Závery

Výsledky úspešne demonštrovali, ako môžu vnútorné odmeny pomocou autokódera na spracovanie viacerých signálov pomôcť pri navigácii agenta cez priestor hľadania stavu predstavujúci pohyb robota v jeho prostredí. Takáto schopnosť by sa mohla ukázať ako užitočná v scenároch, kde má robot obmedzené prostriedky komunikácie so svojím ľudským operátorom. To by malo byť prospešné, keď robot potrebuje pracovať v prostredí, ktoré je nebezpečné alebo je pre ľudí ťažko prístupné. V následnom výskume bude nasadený operačný robot na vyhodnotenie účinnosti vnútorných odmien pri riešení rovnakých úloh, ktoré vykonávajú simulované roboty. Mal by poskytnúť pohľad na skutočnú aplikáciu vnútorných odmien v robotike a mohol by viesť k vývoju pokročilejších a efektívnejších autonómnych systémov.

Doplnkové materiály: Interaktívne grafy: https://wandb.ai/markub/Dueling-DQN-with-AutoEncoder; Zdrojové kódy: https://github.com/markub3327/Dueling-DQN-with-AutoEncoder

Autorské príspevky: Konceptualizácia, IDL a JP; metodika, MK; softvér, MK; validácia, MK; formálna analýza, JP; vyšetrovanie, MK; zdroje IDL; správa údajov, MK; písanie—pôvodná príprava návrhu, MK; písanie – recenzia a úprava, JP; vizualizácia, MK; supervízia, IDL a JP; administrácia projektov, IDL; získanie financovania, IDL Všetci autori si prečítali a súhlasili s publikovanou verziou rukopisu.

Financovanie: Tento výskum bol financovaný Kultúrnou a vzdelávacou grantovou agentúrou MŠVVaŠ SR, číslo grantu KEGA 020UCM-4/2022, a projektom Erasmus+ FAAI: Budúcnosť je v aplikovanej umelej inteligencii - 2022-1-PL01-KA220-HED-000088359, práca balík WP3.

Vyhlásenie inštitucionálnej revíznej rady: Nepoužiteľné.

Vyhlásenie informovaného súhlasu: Nepoužiteľné.

Vyhlásenie o dostupnosti údajov. Nepoužiteľné.

Konflikt záujmov: Autori nedeklarujú žiadny konflikt záujmov.

Literatúra

1. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-driven exploration by self-supervised prediction. In International conference on machine learning (pp. 2778-2787), PMLR, Sydney, Australia (6-11 August 2017). Available online: https://arxiv.org/pdf/1705.05363.pdf (accessed on 7.3.2023).
2. Burda, Y.; Edwards, H.; Storkey, A.; Klimov, O.. Exploration by random network distillation. 2018; Available online: https://arxiv.org/abs/1810.12894 (accessed on 7.3.2023).
3. Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; Munos, R. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems* 2016, *29*. Available online: https://arxiv.org/abs/1606.01868 (accessed on 7.3.2023).
4. Tang, H.; Houthooft, R.; Foote, D.; Stooke, A.; Xi Chen, O.; Duan, Y.; Schulman, J.; DeTurck, F.; Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems* 2017, 30. Available online: https://arxiv.org/pdf/1611.04717.pdf (accessed on 7.3.2023).
5. Oudeyer, P.Y.; Kaplan, F.; Hafner, V.V. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation* **2007**, *11*(2), 265-286. Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.177.7661&rep=rep1&type=pdf (accessed on 7.3.2023).
6. Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; Abbeel, P. Vime: *Variational information maximizing exploration. Advances in neural information processing systems* **2016**, *29*. Available online: https://arxiv.org/abs/1605.09674 (accessed on 7.3.2023).
7. Choshen, L.; Fox, L.; Loewenstein, Y. Dora the explorer: Directed outreaching reinforcement action-selection. 2018. Available online: https://arxiv.org/pdf/1804.04012.pdf (accessed on 7.3.2023).
8. Kamar, D.; Üre, N.K.; and Ünal, G. GAN-based Intrinsic Exploration For Sample Efficient Reinforcement Learning. 2022. Available online: https://arxiv.org/pdf/2206.14256.pdf (accessed on 7.3.2023).
9. Kamalova, A.; Lee, S.G.; Kwon, S.H. Occupancy Reward-Driven Exploration with Deep Reinforcement Learning for Mobile Robot System. *Applied Sciences* **2022**, *12*(18), p.9249. Available online: https://www.mdpi.com/2076-3417/12/18/9249 (accessed on 7.3.2023).
10. Liu, Q.; Liu, Z.; Xiong, B.; Xu, W.; Liu, Y. Deep reinforcement learning-based safe interaction for industrial human-robot collaboration using intrinsic reward function. *Advanced Engineering Informatics* **2021**, 49, 101360. Available online: https://www.sciencedirect.com/science/article/pii/S1474034621001130 (accessed on 2.4.2023).
11. Chen, Z.; Subagdja, B.; Tan, A.H. End-to-end deep reinforcement learning for multi-agent collaborative exploration. In 2019 IEEE International Conference on Agents (ICA), IEEE: 2019, pp. 99-102. Available online: https://ieeexplore.ieee.org/abstract/document/8929192 (accessed on 2.4.2023).
12. Shi, H.; Shi, L.; Xu, M.; Hwang, K.S. End-to-end navigation strategy with deep reinforcement learning for mobile robots. *IEEE Transactions on Industrial Informatics* **2019**, 16(4), 2393-2402. Available online: https://ieeexplore.ieee.org/abstract/document/8807287 (accessed on 2.4.2023).
13. Nguyen, T.; Luu, T.M.; Vu, T.; Yoo, C.D. Sample-efficient reinforcement learning representation learning with curiosity contrastive forward dynamics model. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 3471-3477). IEEE: 2021. Available online: https://ieeexplore.ieee.org/abstract/document/9636536 (accessed on 2.4.2023).
14. Zhang, C.; Ma, L.; Schmitz, A. A sample efficient model-based deep reinforcement learning algorithm with experience replay for robot manipulation. International Journal of Intelligent Robotics and Applications **2020**, 4, 217-228. Available online: https://link.springer.com/article/10.1007/s41315-020-00135-2 (accessed on 2.4.2023).
15. Burgueño-Romero, A.M.; Ruiz-Sarmiento, J.R.; Gonzalez-Jimenez, J. Autonomous docking of mobile robots by reinforcement learning tackling the sparse reward problem. In Advances in Computational Intelligence: 16th International Work-Conference on Artificial Neural Networks, IWANN 2021, Virtual Event. Proceedings, Part II (pp. 392-403). Cham: Springer International Publishing, 2021. Available online: https://link.springer.com/chapter/10.1007/978-3-030-85099-9_32 (accessed on 2.4.2023).
16. Huang, S.H.; Zambelli, M.; Kay, J.; Martins, M.F.; Tassa, Y.; Pilarski, P.M.; Hadsell, R. Learning gentle object manipulation with curiosity-driven deep reinforcement learning. 2019. Available online: arXiv preprint arXiv:1903.08542 (accessed on 2.4.2023).
17. Szajna, A.; Kostrzewski, M.; Ciebiera, K.; Stryjski, R.; Woźniak, W. Application of the Deep CNN-Based Method in Industrial System for Wire Marking Identification. Energies 2021, 14, 3659. Available online: https://doi.org/10.3390/en14123659 (accessed on 7.3.2023).
18. Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence* **2018**, *32*(1). Available online: https://arxiv.org/pdf/1710.02298.pdf (accessed on 7.3.2023).
19. Pang, G.; van den Hengel, A.; Shen, C.; Cao, L. Toward deep supervised anomaly detection: Reinforcement learning from partially labeled anomaly data. In Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining (pp. 1298-1308), August 2021. Available online: https://arxiv.org/pdf/2009.06847.pdf (accessed on 7.3.2023).

20. Michalski, P. Anomaly detection in the context of Reinforcement Learning. (2021). Available online: https://www.researchgate.net/profile/Patrik-Michalski/publication/354694975_Anomaly_detection_in_the_context_of_Reinforcement_Learning/links/6148336fa595d06017db791d/Anomaly-detection-in-the-context-of-Reinforcement-Learning.pdf (accessed on 7.3.2023).
21. Wang, Y.; Xiong, L.; Zhang, M.; Xue, H.; Chen, Q.; Yang, Y.; Tong, Y.; Huang, C.; Xu, B., Heat-RL: Online Model Selection for Streaming Time-Series Anomaly Detection. In Conference on Lifelong Learning Agents (pp. 767-777). PMLR: November 2022. Available online: https://proceedings.mlr.press/v199/wang22a/wang22a.pdf (accessed on 7.3.2023).
22. Ma, X.; Shi, W. Aesmote: Adversarial reinforcement learning with smote for anomaly detection. *IEEE Transactions on Network Science and Engineering* **2020**, *8*(2), 943-956. Available online: https://ieeexplore.ieee.org/abstract/document/9124651 (accessed on 7.3.2023).
23. Rafati, J.; Noelle, D.C. Learning representations in model-free hierarchical reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, July 2019, Vol. 33, No. 01, pp. 10009-10010. Available online: https://ojs.aaai.org/index.php/AAAI/article/view/5141 (accessed on 7.3.2023).
24. Badia, A.P.; Piot, B.; Kapturowski, S.; Sprechmann, P.; Vitvitskyi, A.; Guo, Z.D.; Blundell, C. Agent57: Outperforming the atari human benchmark. In International conference on machine learning (pp. 507-517). PMLR: November 2020. Available online: https://arxiv.org/abs/2003.13350 (accessed on 7.3.2023).
25. Lindegaard, M.; Vinje, H.J.; Severinsen, O.A. Intrinsic Rewards from Self-Organizing Feature Maps for Exploration in Reinforcement Learning. 2023. arXiv preprint arXiv:2302.04125. Available online: https://arxiv.org/pdf/2302.04125.pdf (accessed on 2.4.2023).
26. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. Openai gym. 2016; Available online: arXiv preprint arXiv:1606.01540 (accessed on 7.3.2023).
27. Barto, A.G.; Sutton, R.S.; Anderson, C.W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics* 1983, *5*, 834-846. Available online: https://ieeexplore.ieee.org/document/6313077 (accessed on 7.3.2023).
28. Sutton, R.S., 1995. Generalization in reinforcement learning: Successful examples using sparse coarse coding. Advances in neural information processing systems, 8, 1038-1044. Available online: https://papers.nips.cc/paper/1995/hash/8f1d43620bc6bb580df6e80b0dc05c48-Abstract.html (accessed on 7.3.2023).
29. Sutton, R.S.; Barto, A.G. *Reinforcement learning: An introduction*. MIT press, Cambridge, MA, 2018. Available online: http://www.incompleteideas.net/book/the-book-2nd.html (accessed on 7.3.2023).
30. Moore, A.W. *Efficient memory-based learning for robot control* Technical report No. UCAM-CL-TR-209), University of Cambridge, Computer Laboratory, 1990. Available online: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-209.pdf (accessed on 7.3.2023).
31. Jakovlev, S.; Voznak, M. Auto-Encoder-Enabled Anomaly Detection in Acceleration Data: Use Case Study in Container Handling Operations. *Machines* **2022**, *10*(9), 734. Available online: https://www.mdpi.com/2075-1702/10/9/734 (accessed on 7.3.2023).
32. Fedus, W.; Ramachandran, P.; Agarwal, R.; Bengio, Y.; Larochelle, H.; Rowland, M.; Dabney, W. Revisiting fundamentals of experience replay. In International Conference on Machine Learning (pp. 3061-3071). PMLR: November 2020. Available online: https://arxiv.org/pdf/2007.06700.pdf (accessed on 7.3.2023).
33. Feeney, P.; Hughes, M.C. Evaluating the Use of Reconstruction Error for Novelty Localization. 2021. Available online: https://arxiv.org/pdf/2107.13379.pdf (accessed on 7.3.2023).
34. Krizhevsky, A. Convolutional deep belief networks on cifar-10. Unpublished manuscript, 2010, pp. 1-9. Available online: http://www.cs.utoronto.ca/%7Ekriz/conv-cifar10-aug2010.pdf (accessed on 7.3.2023).
35. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). 2015. Available online: https://arxiv.org/pdf/1511.07289v5.pdf (accessed on 7.3.2023).
36. Lu, L.; Shin, Y.; Su, Y.; Karniadakis, G.E. Dying relu and initialization: Theory and numerical examples. 2019 Available online: https://arxiv.org/pdf/1903.06733.pdf (accessed on 7.3.2023).
37. Saxe, A.M.; McClelland, J.L.; Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. 2013. Available online: https://arxiv.org/pdf/1312.6120.pdf (accessed on 7.3.2023).
38. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. 2013. Available online: https://arxiv.org/pdf/1312.5602v1.pdf (accessed on 7.3.2023).
39. Usama, M.; Chang, D.E. Learning-driven exploration for reinforcement learning. In 2021 21st International Conference on Control, Automation and Systems (ICCAS) (pp. 1146-1151). IEEE: October 2021. Available online: https://arxiv.org/pdf/1906.06890.pdf (accessed on 7.3.2023).
40. Steinparz, C.A. Reinforcement Learning in Non-Stationary Infinite Horizon Environments/submitted by Christian Alexander Steinparz, BSc. Master Thesis, Linz 2021. Available online: https://epub.jku.at/obvulihs/download/pdf/6725095?originalFilename=true (accessed on 7.3.2023).
41. Wang, Z.; Schaul, T.; Hessel, M.; Hasselt, H.; Lanctot, M.; Freitas, N.,. Dueling network architectures for deep reinforcement learning. In International conference on machine learning (pp. 1995-2003). PMLR, June 2016. Available online: https://arxiv.org/pdf/1511.06581.pdf (accessed on 7.3.2023).

42. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*(11), 2278-2324. http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf (accessed on 7.3.2023).
43. Jang, B.; Kim, M.; Harerimana, G.; Kim, J.W.. Q-learning algorithms: A comprehensive classification and applications. *IEEE access* **2019**, *7*, 133653-133667. Available online: https://ieeexplore.ieee.org/document/8836506 (accessed on 7.3.2023).
44. Weights & Biases: Tune Hyperparameters. Available online: https://docs.wandb.ai/guides/sweeps (accessed on 7.3.2023).
45. Zhao, X.; An, A.; Liu, J.; Chen, B.X. Dynamic stale synchronous parallel distributed training for deep learning. In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS) (pp. 1507-1517). IEEE: July 2019. Available online: https://ieeexplore.ieee.org/abstract/document/8885215 (accessed on 7.3.2023).
46. Šimon, M.; Huraj, L.; Siládi, V. Analysis of performance bottleneck of P2P grid applications. *Journal of Applied Mathematics, Statistics and Informatics* **2013**, *9*(2), 5-11. Available online: https://sciendo.com/fr/article/10.2478/jamsi-2013-0008 (accessed on 7.3.2023).
47. Skrinarova, J.; Dudáš, A. Optimization of the Functional Decomposition of Parallel and Distributed Computations in Graph Coloring With the Use of High-Performance Computing. *IEEE Access* **2022**, *10*, 34996-35011. Available online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9741791 (accessed on 7.3.2023).
48. Van Otterlo, M.; Wiering, M. Reinforcement learning and markov decision processes. Reinforcement learning: State-of-the-art, Springer Berlin, Heidelberg: 2012, pp.3-42. Available online: https://link.springer.com/chapter/10.1007/978-3-642-27645-3_1 (accessed on 7.3.2023).
49. Pardo, F.; Tavakoli, A.; Levdik, V.; Kormushev, P. Time limits in reinforcement learning. In International Conference on Machine Learning (pp. 4045-4054). PMLR: 2018, July. Available online: https://arxiv.org/pdf/1712.00378.pdf (accessed on 7.3.2023).
50. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **2008**, 9(11), 2579-2605. Available online: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf (accessed on 7.3.2023).
51. Huang, S.; Dossa, R.F.J.; Ye, C.; Braga, J.; Chakraborty, D.; Mehta, K.; Araújo, J.G. CleanRL: High-quality single-file implementations of deep reinforcement learning algorithms. 2021. Available online: https://www.jmlr.org/papers/volume23/21-1342/21-1342.pdf (accessed on 7.3.2023).
52. Raffin, A. DQN Agent playing LunarLander-v2. Available online: https://huggingface.co/araffin/dqn-LunarLander-v2 (accessed on 7.3.2023).
53. Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *The Journal of Machine Learning Research* **2021**, *22*(1), 12348-12355. Available online: https://jmlr.org/papers/volume22/20-1364/20-1364.pdf (accessed on 7.3.2023).
54. Balis, J. Gymnasium. Available online: https://github.com/Farama-Foundation/Gymnasium/blob/main/gymnasium/envs/__init__.py (accessed on 7.3.2023).
55. Badia, A.P.; Sprechmann, P.; Vitvitskyi, A.; Guo, D.; Piot, B.; Kapturowski, S.; Tieleman, O.; Arjovsky, M.; Pritzel, A.; Bolt, A.; Blundell, C. Never give up: Learning directed exploration strategies. 2020. Available online: https://arxiv.org/abs/2002.06038 (accessed on 7.3.2023).
56. Carsten, J.; Rankin, A.; Ferguson, D.; Stentz, A.. Global path planning on board the mars exploration rovers. In 2007 IEEE Aerospace Conference (pp. 1-11). IEEE: March 2007. Available online: https://www-robotics.jpl.nasa.gov/media/documents/IEEEAC-Carsten-1125.pdf (accessed on 7.3.2023).
57. Liu, J. Research on the development and path exploration of autonomous underwater robots. In ITM Web of Conferences (Vol. 47, p. 01029). EDP Sciences 2022. Available online: https://www.itm-conferences.org/articles/itmconf/pdf/2022/07/itmconf_cccar2022_01029.pdf (accessed on 7.3.2023).