ISSN-L 2664-9969; E-ISSN 2706-5448

UDC 004.8:004.41:004.94
DOI: 10.15587/2706-5448.2026.364354

Andrii Dolhyi,
Kirill Smelyakov,
Yuri Romanenkov,
Anastasiya Chupryna

DEVELOPMENT OF A MODEL OF ADAPTIVE BEHAVIOR OF NON-PLAYER CHARACTERS BASED ON INTELLIGENT AGENTS OF NEURAL NETWORK

The object of research is the process of designing and modeling intelligent agents of a neural network for the behavior of non-player characters (NPCs).

Today, the field of video game development continues to progress, but the methodology for creating the "brain" of NPCs still uses primitive approaches with technological limitations. This creates the problem of ludonarrative dissonance, which actualizes this work, that offers a new solution.

The research results demonstrate the potential of a new approach to creating intelligent and realistic NPCs in video games. The developed models demonstrate convergence of training, which confirms the ability to understand complex concepts, such as the psychological personality model International Personality Item Pool 50.

The conclusions of the work are based on the results of theoretical analysis, technical modeling and practical experimentation. Analysis of the collected data forms the final confirmation of the feasibility of the proposed theory.

An increase in the total step reward from –0.00275 to 0.03176 by the median for the best option and from –0.00733 to 0.02734 for the worst was recorded, taking into account the mathematical limit in [–1.0219; 0.9861] and aggressive normalization of the hyperbolic tangent function. The adaptation of models to the environmental conditions was confirmed by reducing the penalty by 30.47% by the median or by 23.59% by the extreme 5% of the data, for the worst configuration.

The obtained results can be used in game development, to create the "brain" of NPCs. This will allow to get rid of the exponential complexity of development, inherent in current methodologies due to the proposed approach. Using reinforcement learning technologies, it is possible to make a more variable and detailed virtual context, with lower computational resources consumption, avoiding ludonarrative dissonance.

Keywords: *deep reinforcement learning, IPIP-50 model, ludonarrative dissonance, non-player characters, Unity ML-Agents package.*

Received: 20.03.2026
Received in revised form: 28.05.2026
Accepted: 08.06.2026
Published: 31.08.2026

How to cite
Dolhyi, A., Smelyakov, K., Romanenkov, Y., Chupryna, A. (2026). Development of a model of adaptive behavior of non-player characters based on intelligent agents of neural network. Technology Audit and Production Reserves, 4 (2 (90)), 78–89. https://doi.org/10.15587/2706-5448.2026.364354

1. Introduction

Today, the video game industry occupies a fairly large share of the global market and continues to attract the attention of new users, ensuring the sustainable development of its target audience, which is observed for all regions considered [1]. Financial report [2], which declares a forecast of significant growth in the financial achievements of the specified industry. This process is a natural result of the fact that technologies are developing, in particular those used in game projects.

The video game industry is relatively young, compared to other media areas, but it has come a long way, from primitives and abstractions in the 1950s to near photorealistic graphics today [3]. At the same time, with the development of certain aspects, others remain stable for a certain period and begin to stand out in the overall immersive experience of the game, causing ludonarrative dissonance in players [4]. One of such factors is the way of creating the "brain" of the behavior of non-player characters in modern products.

Currently, the general industry standard is the approach using behavior trees, which is a simple tool in its base, but has serious drawbacks. It includes the determinism of character actions and the inability to make a sufficiently variable and deep system, due to the problem of scalability and efficiency of such solutions. This topic is discussed in detail in the work [5], where the authors note the same shortcomings of the current approach. They propose improvements through the use of emotional trees, to introduce the factor of emotions into the system, Monte Carlo Tree Search (MCTS), to predict the consequences of actions, etc. However, this is only an optimization of the current technology, which contains fundamental shortcomings in its concept. It is obvious that when using finite state machines, each state must be clearly predicted and developers must implement transitions between each element of behavior, the complexity of which is described by an exponential function, any improvements to the initial mechanism do not change this. Due to the stagnation and technological limit that has been reached in the current concept of the "brain" of non-player characters, there is a need to introduce a new approach in this aspect of creating video game products.

The object of research is the process of designing and modeling intelligent agents of a neural network for the behavior of non-player characters.

The aim of this research is to develop a neural network model that integrates the formalized International Personality Item Pool 50 (IPIP-50) personality profile, the detailed representation of five basic traits, with the reward function using deep reinforcement learning algorithms. This will make it possible to create realistic game products with non-player characters that have variable, non-deterministic behavior that adapts to the virtual environment, player actions, etc., avoiding ludonarrative dissonance between the visual part and the mechanical content.

To achieve the aim, the following objectives were set:

1) to perform theoretical analysis and evaluation of the existing tools provided by the Unity ML-Agents package, using an evaluation based on the analysis of scientific sources, to determine the ranking of neural network training configurations;

2) to perform mathematical formalization of the IPIP-50 model in the context of its integration into the reward function and the neural network's perception system of the virtual environment, including technical implementation;

3) to conduct a practical experiment to test the success of the proposed theory regarding the use of deep reinforcement learning as the "brain" of the behavior of non-player characters.

2. Materials and Methods

To conduct this research and achieve the scientific and practical goal, the Unity game engine was chosen. The choice is due to the modernity of the technical base and the optimal balance between development flexibility and high performance, according to [6, 7]. To implement neural networks in the selected software, the ML-Agents package, namely the technology of intelligent agents, was chosen as the main tool. It is an open source and powerful tool for scientific and technical activities, according to [8]. In particular, the selected toolkit provides a clear division of roles and isolates the engine from the neural network training environment. This is done due to the additional mlagents library, for the Python side, which manages the training of artificial intelligence through the powerful PyTorch mechanisms.

The chosen technology provides the opportunity to train artificial intelligence models using reinforcement learning algorithms: Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), Multi-Agent Posthumous Credit Assignment (MA-POCA); and imitation learning methods Generative Adversarial Imitation Learning (GAIL) and Behavior Cloning (BC). Also, ML-Agents include internal additional reward modules Intrinsic Curiosity Module (ICM) and Random Network Distillation (RND).

This research considers the issue of creating a toolkit that can be adapted to any video game project, therefore, the GAIL and BC methods are not relevant, due to the lack of the ability to generalize the training data set, for all tasks.

The scientific goal of the work is to prove the convergence of the artificial intelligence model training, which integrates personality data according to IPIP-50 to itself and the reward function. This forms the conditions for the provability of the proposed theory, through testing the best and worst training configurations. The mentioned training variations include all possible alternatives that combine the PPO, SAC, MA-POCA algorithms and the additional reward modules RND and ICM.

To identify the necessary training configurations, a multi-criteria analysis was applied according to the decision-making methodology. The evaluation criteria include a list of seven items. They are given in Table 1. The criteria are optimized by maximizing their indicators. The formation of estimates is based on expert analysis of scientific articles on the subject of research. The criterion of training speed is determined by an interval scale, with values from 1 to 10, while the rest are represented by ordinal ones, in the range from 1 to 5. The scales correspond to values from worst to best.

Table 1

List of multi-criteria analysis criteria

Identification	Name	Evaluation method
K_1	Realism of behavior	How consistently and non-stochastically the model behaves
K_2	Training speed	How quickly the model trains
K_3	Training stability	How stable the configuration is to stochastic deviations
K_4	Implementation flexibility	How much the use of an alternative is complicated under specific conditions
K_5	Computational resource requirements	How much more the model requires resources for training
K_6	Model adaptability	How much the model is plastic in learning and able to change behavior for new needs
K_7	Agent sociability	How well the model can cooperate with its agents

This set of criteria for evaluating alternative training configurations is comprehensive and covers all the main needs of the research. It considers both theoretically important aspects, such as the training speed or its stability, and technical aspects that characterize alternatives in terms of hardware requirements or implementation complexity. The central determinants of the quality of training configurations for this work are also considered, namely the realistic behavior of agents, the adaptability of the model to the dynamic environment, and the ability of agents to logically coexist. The specified list is deliberately abstracted in order to maintain a stable structure of the analysis in general and reduce the effect of vulnerability to the inflationary impact of the weights of the criteria. This list is the authors' original development and expert vision of the issues and challenges considered in this work and is based in particular on the materials of the articles [4, 5].

It should be noted that in this multi-criteria analysis, the initial assessment by criteria was conducted by experts, who are the authors of this research. The actual evaluation was performed on the basis of a collegial consensus, relying on the synthesis of information from selected theoretical materials and heuristic judgments.

Considering the literature basis of the theoretical analysis of the subject area, it is worth noting that the initial list consisted of 108 separate sources. These materials were selected from open scientometric databases, such as IEEE Xplore, Scopus, ResearchGate, and others, based on the subject matter of the relevant subject area and related keywords. All obtained manuscripts and texts were primarily filtered by the time factor, namely their publication date within a few years, to ensure currency, except for a selection of the foundational works which remain fundamental to this field.

At the next stage, the sources were analyzed and sorted by value and relevance of use. Thus, the inclusion and exclusion criteria for works were formed, which were formulated on the basis of relevance and compliance with the research topic. Accordingly, the final list included works that had sufficient technical and practical value, namely, contained a meaningful theoretical part, or demonstrated qualitative and clear empirical data that could be interpreted in the context of this research. At the same time, sources were excluded if their scope was distant enough from the context under consideration, or the data were too abstract and could not be clearly interpreted within the framework of the formed factorization of the assessment, due to the formalization of individual aspects.

It is worth noting that the initial set was immediately formed only from materials that corresponded to the research topic by keywords, however, due to a very specific subject area and the limitations imposed by the selected tools and inclusion and exclusion factors, this list was

ISSN-L 2664-9969; E-ISSN 2706-5448

reduced to 47 works, which included technical literature in particular. Based on this list, an expert heuristic collegial evaluation was conducted according to the formed criteria.

At the same time, it is possible to note that for a concise narrative within the framework of the article, the discussion of this part of the analysis was condensed in order to maintain the general flow of presentation of the materials. That was expressed in the explicit discussion of only the main theses, with references to relevant sources, on the basis of which the general theoretical expert assessment can be confirmed.

To solve the issue of selecting theoretical candidates for conducting a practical experiment, a multi-criteria analysis was applied, based on theoretical information. To rank the normalized data, a linear additive model was used, which uses the normalization (1) and utility (2). The mathematical expression uses the iterative sum of the multiplication, for each criterion, of its weight coefficient and the normalized score value:

$$f_{ijnorm} = \frac{f_{ij} - \min(f_j)}{\max(f_j) - \min(f_j)}; \quad (1)$$

$$W(A_i) = \sum_{j=1}^{n} (w_j \cdot f_{ijnorm}). \quad (2)$$

Given the importance of each criterion for achieving the set aim of research, Table 2 was formed.

Table 2

List of multi-criteria analysis criteria

Identification	Name	Rank	Weight
K_1	Realism of behavior	1	0.2500
K_2	Training speed	4	0.1429
K_3	Training stability	3	0.1786
K_4	Implementation flexibility	6	0.0714
K_5	Computational resource requirements	5	0.1071
K_6	Model adaptability	2	0.2143
K_7	Agent sociability	7	0.0357

For the software implementation of the subject of research, an approach was used to simulate human-like behavior, using a combination of an extensive system of sensors for observing the virtual environment: visual, auditory, tactile, spatial layers. Also, this system is integrated with the psychological model IPIP-50, which is represented by a set of 50 separate indicators of the interpretation of personality traits into a mathematical representation with values from 1 to 5.

Considering the issue of using IPIP-50, for consistency in creating realistic input conditions for training neural networks, it is advisable to abandon procedurally generated sets, and instead use a ready-made, public source with a large amount of data created by people [9]. The specified dataset contains more than 800 thousand individual IPIP-50 personality profiles, after validation, in the format of values from 1 to 5, which corresponds to the range of agreement from completely disagree to the opposite. The specified mathematical representation of the data is normalized according to (3), obtaining a range of values from –1 to 1 inclusive

$$norm = \frac{value - 3}{2}. \quad (3)$$

In particular, it is appropriate to determine the conditions of software and hardware support when conducting the practical part of the research. The required data includes the versions of libraries and other tools, which are the main ones, since this directly affects the conditions for conducting the research – Table 3.

Table 3

Main software versions used

Name	Version
Unity Engine	6.0 (6000.0.68f1) lts (Unity Technologies, USA)
ML-Agents	Release 22 (Unity Technologies, USA)
mlagents	1.1.0 (Unity Technologies, USA)
PyTorch	2.2.2 + cu121 (Meta, USA)
TensorFlow	2.13.1 (Google, USA)
TensorBoard	2.19.0 (Google, USA)

It is also worth indicating the characteristics of the computer on which the calculations were performed. Since there is a need for an equivalent comparison of the best training configuration and the worst one, it is appropriate to exclude the option with cloud technologies. When using distributed systems, the learning process can be significantly affected by problems with data transmission over the network. Data delays, packet loss during continuous transmission of sensitive information, inconsistency in receiving consecutive observations, and thus, communication between components of the experiment via network protocols can potentially disrupt the learning process [10]. The described problems can range from a minor impact on local results to a complete destruction of the learning process with a complete disruption of internal processes. At the same time, the lack of control and opacity of the cluster architecture, available capacities and loads on the distributed system, also can affect the learning process, causing fluctuations in results, etc. and literally hide the experiment transparency. That is why the experiments were conducted locally, on the device from Table 4.

Table 4

Computing system specifications

Components	Specifications
GPU (CUDA)	MSI GAMING X SLIM Nvidia RTX 4070 Super, 12 GB DDR6X (MSI, Taiwan and NVIDIA, USA)
CPU	Ryzen 5 7500f, 6c/12t, 3.7– 5.0 GHz (AMD, USA)
RAM	2 × 16 GB Kingston FURY Beast DDR5 6000 MHz, CL30 (Kingston Technology, USA)
Disk	Samsung NVMe 990 PRO, 2 TB, PCIe 4.0 (Samsung, South Korea)
Swap file	30 GB

Note: GPU – Graphics Processing Unit, CUDA – Compute Unified Device Architecture, CPU – Central Processing Unit, RAM – Random Access Memory, "c/t" is cores/threads, lts – Long Term Support

It is also worth noting that during the experimental part, Curriculum Learning methodology was used, which was responsible for the gradual increasing the complexity of the training environment. Initial conditions – few objects, everything is static, low variation of possible factors of object description. As training progresses, more variations of object descriptions begin to appear in the environment, new obstacles are added, dynamic behavior of the environment appears, etc. This thesis is confirmed by the materials [11], using the SAC algorithm as an example.

The training was conducted with a length of 10 million steps, which is the upper value of the recommended range, according to the ML-Agents documentation. 6 separate lessons were implemented, with progressive complexity and fixed length.

The hyperparameters for training neural network models were set according to the recommendations of the official documentation and taking into account the capabilities of the computing system and the requirements of the experiment. In particular, different configurations received comparatively similar settings.

Also important is the aspect of the environment, as noted above, the aim of research is to consider the convergence of the proposed concept in close to real game conditions, with continuous learning. To support this, a stochastic and dynamic virtual environment was implemented, which is procedurally generated each episode from scratch, in accordance with the requirements of the current lesson.

To assess the effectiveness of the obtained experimental observations, visualizations of internal indicators of the learning process were used, such as the growth of the total and step-by-step reward, mathematical predictions, penalty dynamics, policy losses, etc.

Thus, the considered technological basis formed the subject of this work – experimental results. Mathematical forecasting methods were used, such as linear trend, logarithmic, quadratic and sigmoid regression models. When processing data, the exponential smoothing method – Exponential Moving Average (EMA) was used for their visualization. The combination of methods and methodologies described above ensures the consistency and thoroughness of observations, ensuring the scientific weight and value of this work.

3. Results and Discussion

3.1. Theoretical analysis of deep reinforcement learning training configurations

Research [12] records the theoretical advantage of the Off-Policy algorithm – SAC, in terms of data efficiency, which ensures rapid stabilization of decision-making strategies and reward functions. At the same time, the On-Policy algorithm – PPO, demonstrates a significant advantage in terms of training time, according to the materials of the article [13]. It is also worth noting the high stability of training of neural networks by the PPO algorithm, due to the "clipping" mechanism, which is considered in particular in [14]. On the other hand, work [15] declares that the SAC algorithm demonstrates significant sensitivity to the network temperature parameter, this requires clear and careful tuning, otherwise there is a rapid degradation of the training policy, which leads to irrelevance of the results.

Considering the criterion for validating training alternatives by the factor of realistic behavior, the entropy maximization objective of the Off-Policy algorithm provides a variety of model solutions. However, the stochastic nature of maximum entropy algorithms such as SAC can lead to high variance and negatively affect stability, harming precise continuous control, such as micromotion, as observed in [16]. At the same time, the On-Policy algorithm, especially with the use of recurrent Long Short-Term Memory (LSTM) layers, demonstrates more consistent and strategic actions, which is more realistic, but less variable than SAC, according to the materials [17].

The criterion of adaptability of artificial intelligence models to changes is essential, since in the conditions of video game worlds, environments are stochastic and dynamic; therefore it is important to consider additional internal reward modules. ICM, like the other, by its principle of operation, is aimed at stimulating the curiosity of neural networks, which is reflected in the search for unknown states of observations and associated with interaction with the environment. At the same time, the module is vulnerable to the Noisy TV problem, which is why it is vulnerable in stochastic conditions, such as video game environments, which is declared in [18, 19]. At the same time, the RND module has a different architecture and a different principle of operation, using two additional neural networks, which provides higher stability and adaptability in dynamic conditions in combination with PPO, however, increasing the requirements for computing resources, according to [20].

Considering the issue of the efficiency of training configurations, it is worth emphasizing that the Off-Policy algorithm – SAC, requires significantly more memory and time for their processing, under similar conditions. This is due to the architectural features of the experience replay mechanism. The additional computational load becomes most critical when virtual environments, namely intelligent agents, have large and complex input observation vectors. This aspect of the architecture of SAC-like algorithms significantly affects the overall resource consumption, as shown in a research considering the expansion of input states [21].

Considering the seventh criterion – the potential for group interaction of network agents, it is worth noting that the MA-POCA algorithm demonstrates the highest result, which is due to its architecture and the primary purpose of creation [22]. At the same time, the PPO and SAC algorithms form individual, self-interested behavior, for the same reasons, corresponding to the Independent Proximal Policy Optimization (IPPO) configuration, which levels their potential for cooperation, as discussed in [23]. The Multi-Agent Proximal Policy Optimization (MAPPO) agent training configuration provides the MA-POCA algorithm with centralized training, simplifying scalability, while generalizing and combining individual agents into a single whole, which affects the remaining indicators, according to the materials [24]. It is worth noting that the MA-POCA algorithm is On-Policy and based on PPO in its architecture, due to which it receives similar scores according to the criteria, but taking into account the differences and features of each alternative.

It is worth noting that among the theoretical materials considered, most of the results of stable environments were declared, in which algorithms were trained to achieve a specific goal, maximizing a single behavioral strategy. At the same time, the current task requires neural network models to adapt to stochastic and dynamic environments, maximizing not the path to a quick reward, but rather continuous experience at each step of the simulation.

According to the theoretical analysis of alternative training configurations, a vector description was formed, with the formed scores according to the criteria, which are given in Table 5.

Table 5

Vector description of the formed alternatives

Alternative	Criteria						
	K_1	K_2	K_3	K_4	K_5	K_6	K_7
PPO	3	8	4	5	5	2	1
SAC	4	4	2	4	3	3	1
MA-POCA	3	6	4	3	3	2	5
PPO ICM	3	6	2	4	4	3	1
SAC ICM	4	3	1	2	3	4	1
MA-POCA ICM	3	5	2	2	2	3	5
PPO RND	4	7	4	3	2	5	1
SAC RND	5	4	2	2	2	4	1
MA-POCA RND	4	6	4	2	1	5	5

Multi-criteria analysis according to the methodology of decision theory involves analysis using the Pareto method, however, according to the vector description, no dominated variations were identified, and therefore, further analysis is carried out in full.

According to the results of normalization of Table 5 and further application of (1) and (2), Table 6 was formed – the final ranking of alternative training configurations, based on the composite utility score.

As a result of multi-criteria analysis using the decision theory methodology, a ranking was obtained, with the best alternative PPO RND and the worst MA-POCA ICM. However, since MA-POCA is similar to the PPO algorithm, it is appropriate to investigate the comparison of different approaches – On-Policy and Off-Policy. Therefore, this research aims to investigate the convergence of the proposed methodology – the application of the IPIP-50 personality model in training neural networks in a video game environment. Consequently, it is

ISSN-L 2664-9969; E-ISSN 2706-5448

appropriate to consider the alternatives PPO RND and SAC ICM, as representatives of completely different architectures and different levels of compliance with theoretical needs.

Table 6

Final ranking of training configurations

Rank	Alternative	Composite utility score
1	PPO RND	0.6827
2	MA-POCA RND	0.6393
3	SAC RND	0.5077
4	PPO	0.5000
5	SAC	0.3857
6	MA-POCA	0.3774
7	PPO ICM	0.3446
8	SAC ICM	0.3214
9	MA-POCA ICM	0.2506

The main limitation of this theoretical analysis and ranking is that the assessment, although clearly formalized and based on scientific sources, however, as noted above, this work is a new approach in this field. This is expressed in the fact that the basic use of neural networks in games differs from what is being studied – continuous integration of a neural network into the environment. In addition, the assignment of ratings for ranking is a heuristic analysis of the materials considered, which can only partially reveal certain aspects, which imposes its limitations on the accuracy of theoretical ranking. Also, the assignment of weights to the criteria of multi-criteria analysis is the author's vision of the requirements for the new approach, which introduces a certain degree of subjectivity.

For further research, it is advisable to develop the aspect of unification and formalization of approaches to evaluating different configurations of model training using reinforcement learning algorithms. Also, the formation of stable requirements for assessment – criteria and their weights – is a potential goal for developing the theoretical side of this field of knowledge.

3.2. Mathematical model development and implementation

It is worth noting the normal distribution of personality profiles in this dataset, which is proven by the Principal Component Analysis (PCA) projection clustering diagram shown in Fig. 1. According to the visualization, the continuity of the transition from one personality profile to another is clearly traced, covering all training needs.

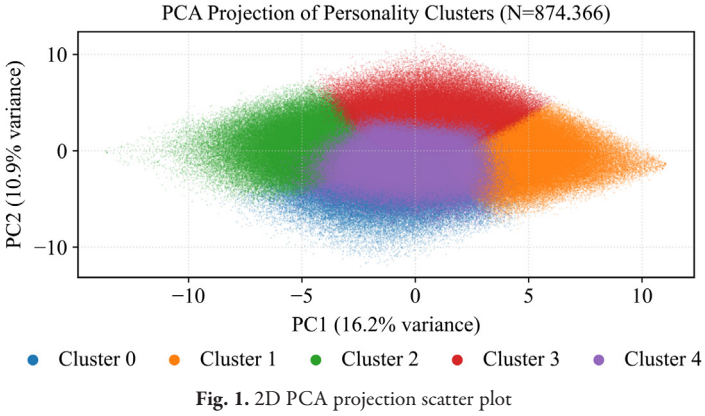


Fig. 1. 2D PCA projection scatter plot

Considering the architectural aspect of the software implementation, it can be noted that the data from the set of profiles directly affects two components of the model. The reward function – it correlates behavior with personality. At the same time, data is transmitted directly to the observations of the neural network agent. These both aspects are necessary to create correlations between the settings of the "brain" and the reward received according to the actions taken. The described architectural solution is shown in Fig. 2, in an abstract, general form.

Note that the neural network has its own architectural limitations that impose requirements on the input vector of agent observations, namely that it is of fixed size with a numerical representation. This means that it is necessary to formalize the description of information about the virtual environment and its content, with which non-player characters interact under the guidance of agents. For this purpose, a system of tags – factors that form a general representation of objects is introduced. Accordingly, the input vector receives a sequence of 0 and 1 (multi-hot encoding method) – a mathematical representation of a fixed size that conveys information about the presence or absence of each factor. It is on the basis of this system that the implementation of the modular reward function of the neural network is formed, in particular the part with the IPIP-50 personality imprint.

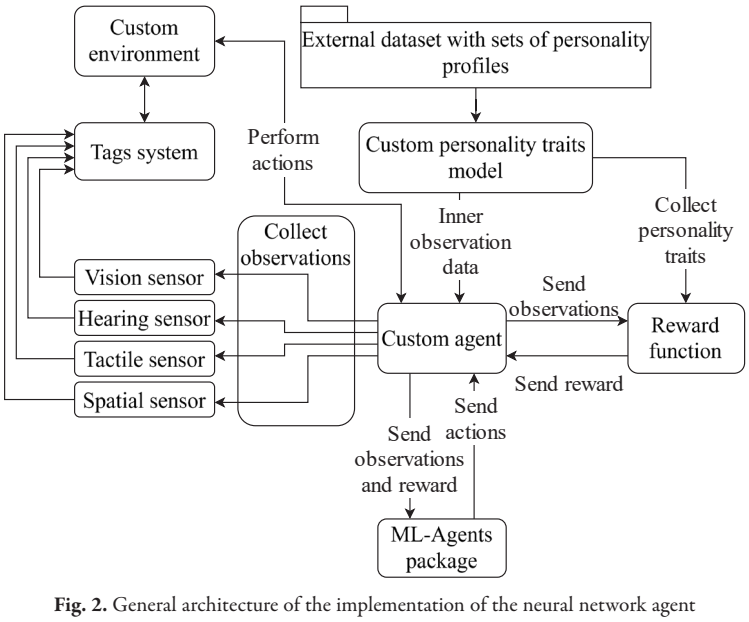


Fig. 2. General architecture of the implementation of the neural network agent

The personality reward system is implemented using a three-level approach. The third, external – proxy entity, used to normalize the final score according to (4), corresponds to the agent level. The second level – represents an intermediate layer between the final formulas and the general controller, here the intermediate normalization is performed within the framework of one IPIP-50 trait, according to (5) and (6). The last layer, the inner one – the first, corresponds to the final formulas – the connections between the trait and the factor of the formalization of the environment. The normalization formulas are aimed at compressing the score into the range between –1 and 1. Such an architectural solution is due to the mathematical needs of the neural network training process [25, 26] and abstraction at all levels. It is necessary for the uniform value of each personality trait, with different numbers of connections with tags, and compressing the total indicator for 50 aspects of the IPIP-50 model:

$$tanh\left(x\cdot 0.05\right)\in\left[-0.987;\ 0.987\right],\ \text{at}\ x\in\left[-50;\ 50\right]; \quad (4)$$

$$norm\_coef=\frac{1}{\sum_{i=1}^{n}\left|max\_reward\right|}; \quad (5)$$

$$x = norm\_coef \cdot \sum_{i=1}^{n} (reward). \quad (6)$$

It is worth noting that agents have a corresponding space of actions that they can perform in a virtual environment. This list includes movement expressed by a direction vector, and non-player characters can also turn around, move their heads in two directions, along two axes: pitch and yawn, simulating looking around, and interact with objects with appropriate tags. For high-quality training, the reward system should include a system of penalties that stimulate the activity of agents and penalize for unrealistic actions, such as 360-degree rotations or step-wise changes in direction, etc. This system has safety features – small ranges with penalty-free threshold to give the neural network a safe space for action. The graphs of the penalty functions with mathematical expectation are shown in Fig. 3, and the general reward function in Fig. 4. Furthermore, Table 7 details mathematical grounds within Expected Values for each penalty component.

To reveal and fix the architectural solution for implementing the integration of the IPIP-50 personality model into the agent reward function, a Unified Modeling Language (UML) diagram was formed that visualizes the abstract representation of the research subject in Fig. 5.

It is worth noting the innovation of this approach, using methods and algorithms of deep reinforcement learning, which differs from the established trends with training models to achieve a static goal in the shortest time – the number of steps. The considered software and technological solution changes the approach, using a continuous method of accumulating evaluation, which stimulates the neural network to optimize not the general strategy of achieving a large, one-time reward, but to adapt at each step of the calculations. That is why it is necessary to normalize the initial reward in order to maintain the stability of learning, through the stabilization of gradient descent.

Considering the development of this work, it is worth noting the conceptual difference from current analogues. The modern application of neural networks in games can be divided into 2 ways: the use of heavy multimodal Large Language Models (LLMs), to form a more human-like interaction, and simplified models that are used in primitive games and aimed at optimizing their passage. However, this work explores the creation of a different type of model – which takes the best aspects of both directions. This development combines resource optimization through limiting input and output data and narrowing specialization to a deterministic space, from which an almost infinite set of behaviors is formed, which is similar to the use of large networks. In addition, the current configuration of the neural network environment perception system is based on the idea of creating human-like behavior, while analogues are built on the principle of completing the task under any conditions.

At the same time, due to the specifics of the architecture and the decisions made, this development has a number of limitations. First of all, due to the increase in the state space, the model architecture requires much more time for full training than simple analogues. In addition, although unification through tags was applied in combination with a three-level reward function and multifactor normalization, the weights of in-game aspects still require clear balancing and adjustment to the needs of developers. In addition, model optimization through fixing the input and output vectors imposes limitations, which are expressed in the need to change the model architecture and complete retraining in the event of a conceptual modification of environmental conditions. Considering the prospects for further development of this development, it is worth noting several main directions. The first is to further unify the architecture and the environment perception system to eliminate the limitations of the determinism of a set of environmental factors, which will allow for additional training of ready-made models. The second is the modification and complication of the architecture of the neural network model itself and the learning algorithms, which will bring flexibility closer to multimodal samples.

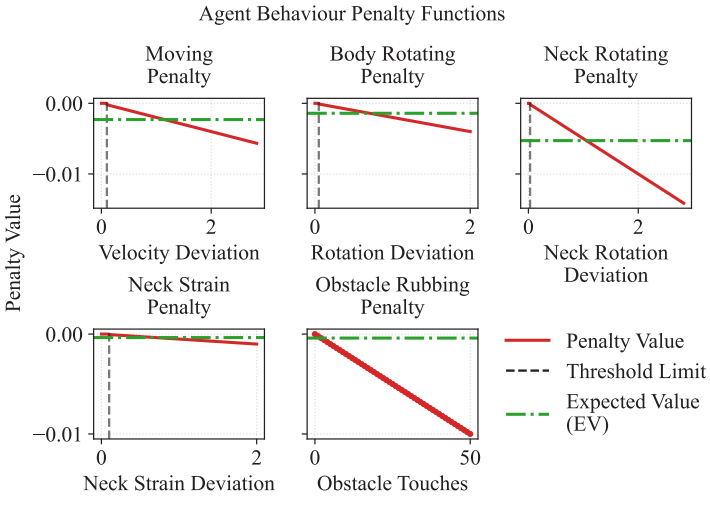


Fig. 3. Graphs of the components of the penalty system

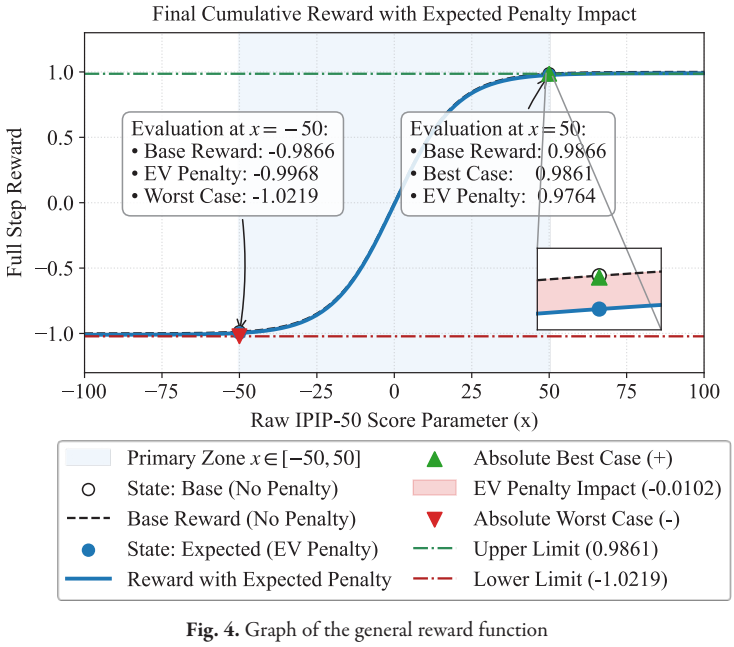


Fig. 4. Graph of the general reward function

Table 7

Mathematical parameters of the penalty system

Component	Threshold	Multiplication coefficient	Expected Value	Max possible value
Moving	0.1	−0.002	−0.0023	−0.00566
Body rotating	0.05	−0.002	−0.00141	−0.004
Neck rotating	0.03	−0.005	−0.00528	−0.01414
Neck strain	0.1	−0.0005	−0.00035	−0.001
Obstacle rubbing	Per touch	−0.0002	−0.0004	Infinite (practically about −0.01)
Existence	Per step	−0.0005	−0.0005	−0.0005
Total	–	–	−0.01024	−0.0353

ISSN-L 2664-9969; E-ISSN 2706-5448

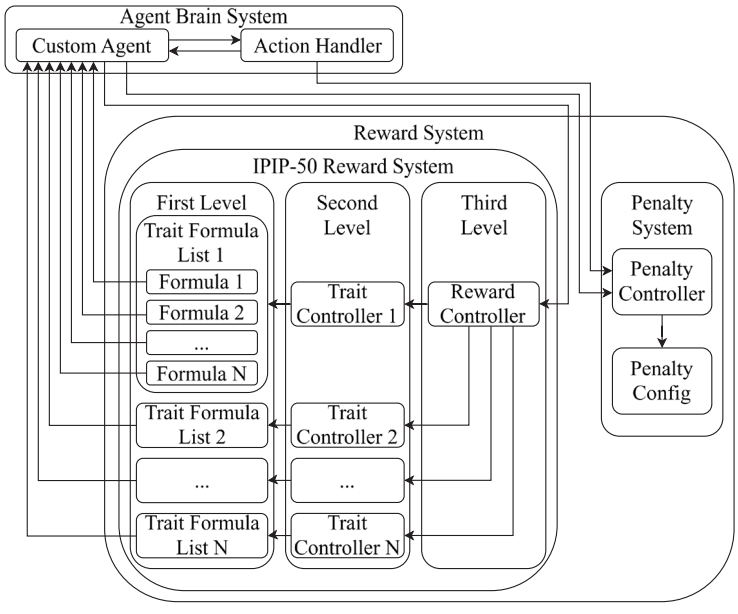


Fig. 5. General architecture of the reward function

3.3. Practical experiment of testing proposed training configurations

To conduct a further qualitative assessment of the results of the practical part of the research, it is necessary to provide summary tables of the data of rewards and penalties – Table 8 for PPO configuration and Table 9 for SAC alternative.

The analysis of the results should begin with an analysis of the total reward graph, which is shown in Fig. 6. The Cumulative Reward graph indicates the dynamics of the total reward per episode for the entire learning process. The next graph – Fig. 7, the dynamics of the step-by-step extrinsic reward, complements the first.

These graphs are of primary importance in determining the convergence of neural network models. As can be seen from the first graph, both training configurations increase their own cumulative reward for each subsequent episode. At the same time, SAC ICM, as predicted in the theoretical analysis, has worse indicators than PPO RND, while demonstrating constant adaptation to the stochastic environment and the dynamics of learning conditions.

Table 8

Summary table of the data obtained on rewards and penalties for PPO model

Metric	Lesson	Mean	Median	Min	Max
Reward	0	0.00500	–0.00275	–0.42484	0.59290
	1	0.01858	–0.00008	–0.37082	0.72453
	2	0.02472	0.00601	–0.44053	0.69577
	3	0.03800	0.01485	–0.41894	0.75655
	4	0.05232	0.01896	–0.47524	0.77154
	5	0.08310	0.03176	–0.54248	0.86864
	Overall	0.04976	0.01345	–0.54248	0.86864
Penalties	0	0.00589	0.00566	0.00074	0.01740
	1	0.00594	0.00571	0.00102	0.02635
	2	0.00598	0.00575	0.00073	0.03085
	3	0.00598	0.00576	0.00050	0.02573
	4	0.00591	0.00570	0.00061	0.02507
	5	0.00586	0.00564	0.00066	0.01885
	Overall	0.00592	0.00570	0.00050	0.03085

Table 9

Summary table of the data obtained on rewards and penalties for SAC model

Metric	Lesson	Mean	Median	Min	Max
Reward	0	–0.00006	–0.00733	–0.45869	0.63428
	1	0.01259	–0.00152	–0.58282	0.61990
	2	0.02118	0.01341	–0.44536	0.66919
	3	0.02300	0.01630	–0.54161	0.71695
	4	0.04767	0.02722	–0.54748	0.78679
	5	0.05878	0.02734	–0.43657	0.88221
	Overall	0.03729	0.01627	–0.58282	0.88221
Penalties	0	0.01110	0.01090	0.00112	0.02351
	1	0.01081	0.01026	0.00167	0.04907
	2	0.00981	0.00921	0.00182	0.05478
	3	0.00959	0.00903	0.00143	0.04662
	4	0.00870	0.00809	0.00093	0.03558
	5	0.00845	0.00778	0.00111	0.03221
	Overall	0.00922	0.00854	0.00093	0.05478

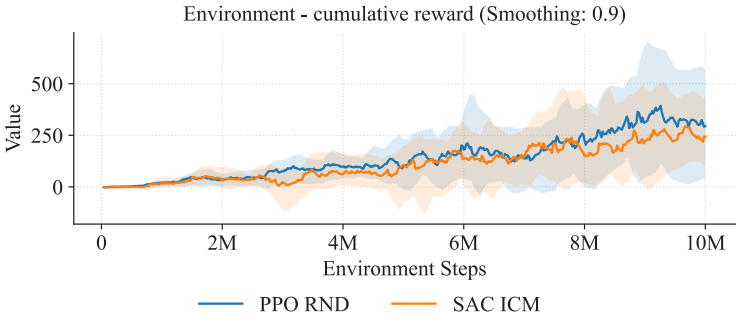


Fig. 6. Cumulative reward value graph, smoothing factor 0.9

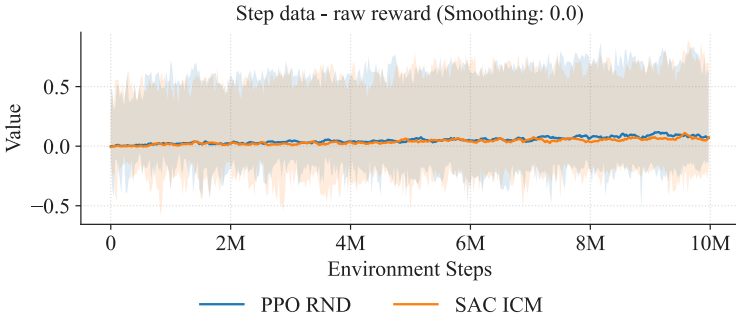


Fig. 7. Raw reward value graph

Since the length of the episodes increases gradually when using Curriculum Learning, it is necessary to analyze the pure step-by-step reward, without taking into account the penalty portion. The results show a gradual adaptation with increasing values, indicating that the model understands the principles and patterns of evaluation, along with the rules of behavior. The graph shows the generalized reward curve, which is again larger in the PPO RND than in the SAC ICM configuration. The semi-transparent areas correspond to the fluctuations of the actual values, without taking into account the EMA smoothing. This visualization clearly demonstrates the dynamics from the initial range of values from –0.4 to +0.4 and after training: from –0.2 to +0.7.

The virtual environment in which the models were trained has a very stochastic nature, starting from procedural generation, ending with dynamic objects that appear over time and cover a larger portion at the end. Therefore, improving the range of evaluation is a clear marker of learning success, given the evaluation limit from –1.0219 to +0.9861.

When considering the aspect of the reward, it is worth considering the aspect of mathematical prediction, the further trend. Fig. 8 visualizes the mathematical predictions of the results of the Cumulative Reward graph, limited to a range of 40% steps, since further calculations lose correlation with the actual indicators.

As can be seen from the lines of potential values, further growth is clearly traced, both for the dominant alternative – PPO RND and for the worst one – SAC ICM. Furthermore, as predicted by the theoretical analysis, SAC ICM loses in all forecasting models. Only the linear trend that is built on the last 20% of steps has a slight slope, which predicts an overtaking of PPO RND in the distant future. At the same time, the sigmoid model demonstrates that the On-Policy algorithm will accelerate the adaptation when the Off-Policy variation reaches a plateau, converging to a suboptimal local minimum.

Another indicative factor of the convergence of the created model are the penalty rate graphs. The general representation on a single canvas, similar to the reward in Fig. 7, 9 demonstrates the dynamics of the raw penalty values. It highlights the smoothed value through the EMA and the actual raw data fluctuations, using translucent zones.

The above graph clearly demonstrates the dominance of the PPO RND algorithm, which from the very beginning has lower penalty values and lower fluctuations in outliers. SAC ICM, on the other hand, is more vulnerable to penalties, which is due to the specifics of learning, where the algorithm does not rigidly consider the latest experience, but random facts from the entire memory buffer (replay buffer), which imposes restrictions. At the same time, the graph clearly demonstrates the growth of penalties after 1 million steps. This is due to the first appearance of dynamic objects moving along a certain trajectory, which neural networks have not observed before. The fact that both alternative configurations coped and adapted to the changes indicates the convergence and relevance of the developed solution. It is worth noting that the On-Policy algorithm, as theoretically predicted, coped with stabilization faster, but the fact that the outsider Off-Policy variation was also able to adapt indicates the generalized success of the model.

It is advisable to analyze the penalty graph smoothed by the EMA algorithm, with visualization of local and global trends, with visualization of the lesson boundary – Fig. 10.

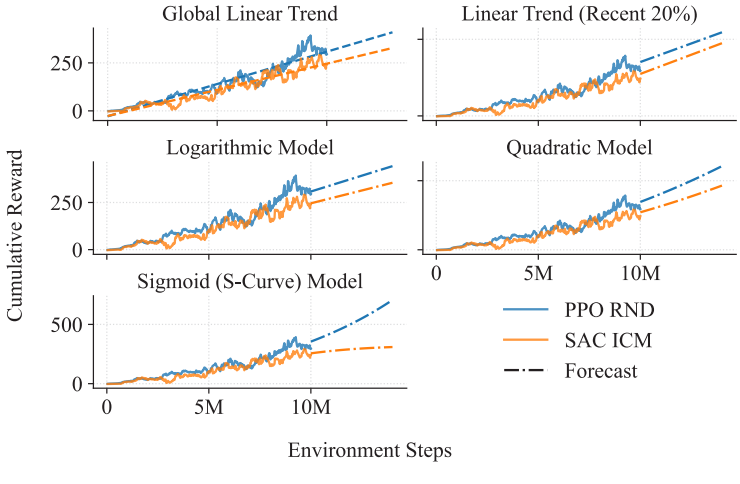


Fig. 8. Mathematical prediction graphs of the cumulative reward graph

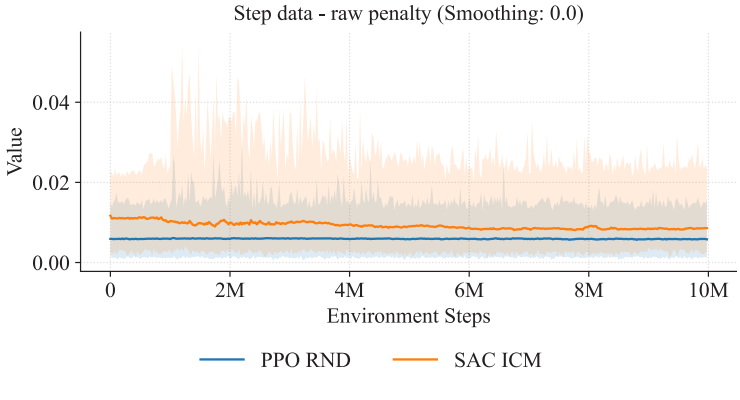


Fig. 9. General graph of the dynamics of the penalty indicator

ISSN-L 2664-9969; E-ISSN 2706-5448

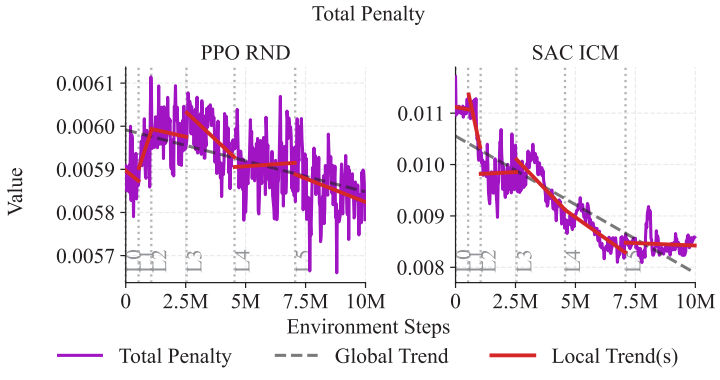


Fig. 10. Comparative presentation of the penalty graph

Analyzing the above graphs, the difference between the obtained configurations is visible, if the PPO RND network from the very beginning forms patterns and decision-making strategies that receive moderately low penalties, while the SAC ICM variation starts with twice as high penalty results. However, as the training progresses, it is clear that the On-Policy algorithm exhibits a tolerance for penalties, demonstrating minimal policy optimization in the specific regard. This aspect is explained by the fact that the total reward exceeds the penalties, which is why a minor violation of the rules is permissible for artificial intelligence. At the same time, the Off-Policy configuration has a different learning algorithm that starts with the accumulation of experience, due to which the network starts with higher indicators. The SAC ICM variation demonstrates a high tendency to a steady decrease in penalties, improving its policy. This is reflected in a 30.47% by median and 23.59% by extreme 5% reduction in the total penalty during training, however, due to the mechanism of accumulation of experience, the Off-Policy algorithm needs much more time for stronger adaptation. However, these graphs demonstrate the success of training, expressed in the initial stability of the PPO RND configuration and the steady adaptation of the SAC ICM alternative, which started with worse indicators.

In addition, it is worth analyzing the components of the penalty values, namely the distribution of the penalty by the categories of behavioral infractions, which is shown in Fig. 11.

Analyzing the above diagrams of the decomposition of the total penalty into individual components, it is possible to see that, as noted above, for PPO RND, the current value of penalties is acceptable, which is why it does not try to significantly optimize the resulting error distribution. At the same time, this fact is confirmed by the fact that the On-Policy model, with an increase in penalties for touching obstacles, reduced other indicators, leaving the overall indicator constant. At the same time, the Off-Policy analogue tries to adapt the behavior to acceptable values, by internalization the objective function, due to which it almost completely removes erroneous body turns and reduces head tilts.

The described results of the analysis of the penalty function are indicative and demonstrate a deep understanding of the neural network models of the connections between various aspects of the environment. Although SAC ICM has vulnerabilities due to the learning algorithm and is generally worst rated alternative in the theoretical part, this configuration also captures causal relationships of its actions and consequences, which proves the convergence of the model.

To finally prove the theory – neural network agents is relevant and successful in terms of convergence, it is necessary to consider the main, internal indicators of artificial intelligence training – Policy Loss and Value Loss. The graphs of these related indicators are shown in Fig. 12. The given diagrams demonstrate the initial difference between the two configurations: The On-Policy algorithm, as noted above, is more stable, in particular using the "clipping" mechanism, which is expressed in almost straight, zero curves of the policy loss and reward prediction indicators. At the same time, the Off-Policy variation initially has adaptation problems due to architectural factors. At a distance of the first million training steps, the SAC ICM configuration only exhibits severe policy degradation, which corresponds to a state of complete random exploration. However, with the accumulation of sufficient experience, this variation begins to adapt quickly, receiving an improvement in the reward and, accordingly, restoring other indicators. At the end of training, both alternative training variations coincide in zero values, which indicates the success of both configurations. At the same time, as predicted by theoretical analysis, PPO RND has greater potential and demonstrates more stable work than SAC ICM.

To prove the factor of realism and variability of the created neural network models using the IPIP-50 model, it is necessary to analyze the entropy curves of both training configurations, which are shown in Fig. 13.

As can be seen from the visualization above, both models demonstrate non-zero final entropy of the models' behavior. The diagram shows that both algorithms reached a certain plateau, starting from 5–6 million training steps. At the same time, the stochasticity level is higher in PPO RND than in SAC ICM, which confirms the initial predictions. However, this result is evidence of the continuity and variability of the obtained implementation of the behavior of non-player characters, since both networks have a sufficient stochastic space of performed actions, unlike finite state machines or trees. The worse result of Off-Policy variation may be due to the fact that due to the initial policy degradation and gradients instability, the algorithm was forced to aggressively adapt, which affected the final result.

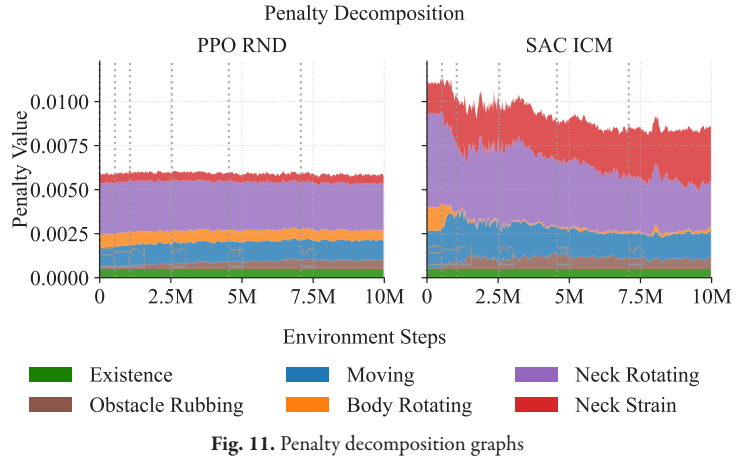
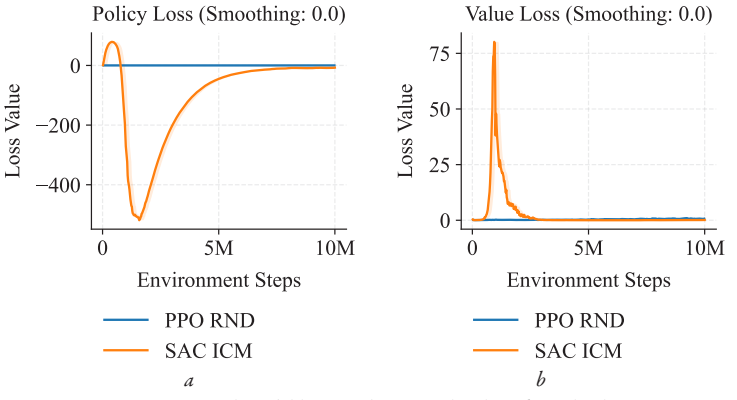


Fig. 11. Penalty decomposition graphs



a *b*

Fig. 12. Internal model loss graphs: *a* – policy loss; *b* – value loss

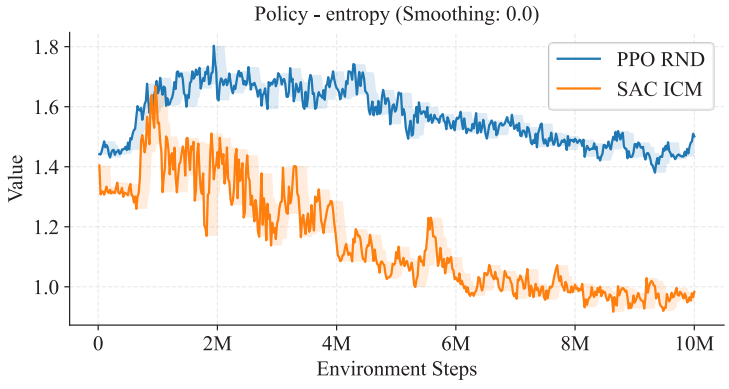


Fig. 13. Graph of comparison of entropy of neural networks

3.4. Discussion

The presented theoretical and practical research results demonstrate the success of the primary theory regarding the feasibility of using the proposed approach for creating non-player character behavior using deep reinforcement learning algorithms, integrating the formalized psychological model IPIP-50 into it.

The tested solution has a conceptual difference from analogues and considered scientific sources, which is expressed in the approach with continuous adaptation of behavior, instead of achieving a deterministic terminal goal, which does not take into account the value of intermediate steps. The results of the reward and penalty demonstrate the understanding of the task set by artificial intelligence, taking into account its non-typicality.

The feature of the proposed methodology is the research of a potential technology that should solve the considered problems. The updated approach to implementing the "brain" of non-player characters involves the use of neural networks, namely deep reinforcement learning algorithms, which should provide high variance and sufficient depth of behavior, including solving the problem of scalability and state explosion of finite state machines.

The key characteristic of the proposed solution is the idea of using psychological models of personalities, which should formalize the internal settings of artificial intelligence agents, ensuring human-like, realistic behavior. The BigFive model was chosen as the factors for tuning the "brain", namely its interpretation IPIP-50, which represents a deeper interpretation of the general basis of the five factors. Today, the power of personal computing systems has increased to a level that allows running heavy LLMs locally. Also, with the popularization of artificial intelligence, its integration into video game products is becoming an increasingly widespread phenomenon. For example: JARVIS-1 [27] and Mantella [28], these are custom game modifications for the existing products Minecraft and The Elder Scrolls V Skyrim, respectively. Both projects use a common approach with LLM that understand the context of the virtual environment and are aimed at reacting realistically, interacting with the player. It is worth noting that JARVIS-1 is the subject of the work "JARVIS-1: Open-World Multi-Task Agents With Memory-Augmented Multimodal Language Models", which proves the relevance of the issue at a higher, scientific, level.

At the same time, the proposed approach is a separate technical solution, since it is aimed at creating a conceptually individual model of behavior simulation, avoiding the use of LLM that require significant computational resources. Instead, the idea of this work is to create a lightweight neural network that imposes minimal computational overhead, freeing up resources for the game, using deep reinforcement learning algorithms.

In particular, the described process considers technologies and methods based on deep reinforcement learning algorithms, specifically using the Unity ML-Agents package. Results can be used to implement the new approach – creating a "brain" of behavior of non-player characters using personalized artificial intelligence models, using the example of integrating the IPIP-50 model.

The research results are of great importance primarily for the gaming industry, in particular the segment of Triple-A (AAA: high-budget and quality, major-publisher) projects. Such projects frequently encounter the problem of ludonarrative dissonance, which arises due to the discrepancy in the realism of the visual component and the primitiveness of the implementation of the virtual world content. Also, the observations formed can be useful in a number of other industries, such as the creation of realistic, human-like assistants with artificial intelligence, etc.

As can be seen from the results of the practical experiment, the developed solution proved the feasibility of the proposed approach, not only by simple convergence of learning, but also by overcoming the main problem of deterministic systems and finite state machines. The environment had a set of states, which consisted of more than 5^{50} states, which only describes the number of combinations of the personality profile. With such a continuous space of configurations, the model not only formed a strategy that leads to the gradual optimization of the reward function, but also maintains a sufficient level of entropy – the variability of the decisions made. It is this aspect that demonstrates the essential and conceptual difference between deterministic approaches and the use of continuous behavior of a neural network in the context of the problem of ludonarrative dissonance, which is considered in this work.

First of all, this work had objective limitations in computational resources. The process of training artificial intelligence models, in particular deep reinforcement learning, is resource-intensive and time-consuming, which is a primary limitation. In addition, standardized out-of-the-box algorithms, such as PPO, SAC, etc., have their own needs and architectural features, which determine boundary conditions, requiring specialized developments, with optimization of architectural and algorithmic solutions for the current task, avoiding generalized methodologies.

The ongoing state of martial law directly affected the experimental part. As stated above, for the stability and controllability of the process of training artificial intelligence models, which lasted a long time of continuous calculations, power outages actually invalidated the current progress of training. Although ML-Agents provides a system of checkpoints and resumption of training from them, the effectiveness of the last episodes will be lost, which leads to errors and deviations in the results, requiring starting from the beginning.

The next directions of development of the obtained results are two main aspects.

The first is the consideration of the finished architecture of the proposed solution from the point of view of modifying the neural network model itself for each aspect of the architecture, learning algorithms, additional components, etc. Developments in this direction should be focused on optimizing approaches specifically to the context of game projects and the need for continuous behavior rather than determinism of the environment and goals for intelligent agents.

The second aspect that should be considered is the issue of scalability and determination of the limit sizes of the model and training configurations, regarding the issue of inflation of the weights of virtual environment factors. This direction is conditioned by an existential question that is formed on the basis of the learning intersection between the state of relevant perception of the environment by the neural network and the moment when it begins to perceive the world not according to objective indicators but according to the totality of the general reward function, according to which the totality of low-cost

ISSN-L 2664-9969; E-ISSN 2706-5448

actions exceeds the reward for the more expected one, due to the inflation of its weight.

4. Conclusions

1. Theoretical analysis and evaluation of the existing tools provided by the Unity ML-Agents package, using an evaluation based on the analysis of scientific sources were performed, to determine the ranking of neural network training configurations.

Theoretical multi-criteria analysis revealed a ranking of training configurations, according to the tools of the ML-Agents package, the Unity video game engine, from the most successful PPO RND variation to the least successful combination of the MA-POCA algorithm with the ICM module. The analysis was based on the methodology of analyzing existing scientific and technical sources that used the selected technology to achieve a deterministic terminal goal, which is conceptually different from the proposed new approach – training continuous adaptation of agents to stochastic, dynamic virtual environments.

According to the results of this task, a clear ranking of alternative configurations corresponding to the selected toolkit was formed. It was determined that PPO + RND is the best alternative, with an integral utility index of 0.6827, while the weakest configuration was MA-POCA + ICM with an index of 0.2506. However, since the research focuses on the individual behavior of non-player characters, it is appropriate to avoid using the MA-POCA algorithm due to its collective nature of adaptation. Therefore, a more relevant alternative was selected for further research – SAC + ICM, with an integral utility index of 0.3214.

2. Mathematical formalization of the IPIP-50 model in the context of its integration into the reward function and the neural network's perception system of the virtual environment, including technical implementation was performed.

As a result of this task, a mathematical model and an architectural solution for implementing the reward function were revealed. A three-level system was developed that plays the role of a modular reward function, which has aspects of normalization and balancing of the observation evaluation system. The feasibility of using an external dataset that has a normal distribution of personality clusters and is responsible for a more humane profiling of the training process was indicated.

The mathematical prediction of the results of the penalty models was declared as –0.01024 per step, or –0.0353 in the worst case. In general, the neural network will receive an assessment in the range [–1.0219; 0.9861], taking into account the worst case. At the same time, the real values will not reach these values and will lie in the space [–1; 1], which fully satisfies the requirements of neural network optimization algorithms.

3. The analysis of the results confirmed the success of adapting neural networks to a different concept of the learning environment and the necessary behavior strategy. Reward indicators increase, even with the complexity of the learning environment, penalties decrease.

It is worth noting that during model training, an increase in the total step reward was recorded from [–0.42484; 0.59290] – with a median of –0.00275, to [–0.54248; 0.86864] – with a median of 0.03176 for the best configuration. At the same time, the worst alternative rose from [–0.45869; 0.63428] – with a median of –0.00733, to [–0.43657; 0.88221] – with a median of 0.02734. Taking into account the mathematical limit in [–1.0219; 0.9861] and the aggressive nature of the normalization of the hyperbolic tangent function, this result demonstrates the success of training. The experiment demonstrated the convergence of the theoretically best and theoretically worst training configurations. In particular, the adaptation of algorithms to environmental conditions was recorded, which is expressed in a reduction in penalties by 30.47% for the median or by 23.59% for the extreme 5% of the data, for the worst configuration and a reduction by 0.85%, respectively for the best one. In addition, it is worth noting that according to the theoretical ranking, the best model shows a penalty level half as low as the predicted value and the worst configuration shows a penalty level close to the predicted one, with a further decrease. The percentages relative to the predicted ones are approximately 55.66% for the overall median and a decrease from 55.27% to 55.08%, in the best configuration, against 83.4% overall and 106.45% with a decrease to 75.98%, respectively, for the worst alternative.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The research was performed without financial support.

Data availability

Data will be made available on reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies in creating the submitted paper.

Authors' contributions

Andrii Dolhyi: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Visualization; ***Kirill Smelyakov***: Project administration, Supervision, Conceptualization, Writing – review and editing; ***Yuri Romanenkov***: Validation, Data curation, Writing – review and editing; ***Anastasiya Chupryna***: Resources, Validation, Writing – review and editing.

All authors have read and agreed to the published version of the article and are jointly responsible for the reliability, accuracy, and integrity of the data and research results presented in the article.

References

1. Palma-Ruiz, J. M., Torres-Toukoumidis, A., González-Moreno, S. E., Valles-Baca, H. G. (2022). An overview of the gaming industry across nations: using analytics with power BI to forecast and identify key influencers. *Heliyon, 8 (2),* e08959. https://doi.org/10.1016/j.heliyon.2022.e08959
2. *AAA Games Market Size, Share, Growth, and Industry Analysis, By Type (PC Games and Console Games), By Application (0–13 Years Old, 13–18 Years Old, and More Than 18 Years Old), and Regional Insight and Forecast to 2035* (2025). Business Research Insights. Market Research Report & Consulting. Available at: https://www.businessresearchinsights.com/market-reports/aaa-games-market-117773
3. Daniela Gomez Rios, M., Araujo, K., Castro, F., Angel Quiroz Martinez, M. (2024). Evolution in videogame graphics: an approach between reality approach and user perspective. *Intelligent Human Systems Integration (IHSI 2024): Integrating People and Intelligent Systems.* AHFE International. https://doi.org/10.54941/ahfe1004535
4. Kleszczyński, K. (2023). The Ambiguity of Ludonarrative Dissonance - the Transhumanist Aspect of Agency in the BioShock. *International Journal of Slavic Studies Transgressive, Pragmatic and Speculative Horizons of Popular Literature and Culture, 5.* https://doi.org/10.61827/m1yk-vafg
5. Hu, C., Wu, R., Deng, X. (2024). Coordination of NPCs in multi-agent systems based on behavior trees. *Applied and Computational Engineering, 95 (1),* 126–140. https://doi.org/10.54254/2755-2721/95/2024melb0068
6. Bao, M., Tao, Z., Wang, X., Liu, J., Sun, Q. (2024). Comparative Performance Analysis of Rendering Optimization Methods in Unity Tuanjie Engine, Unity Global and Unreal Engine. *2024 IEEE Smart World Congress (SWC).* IEEE, 1627–1632. https://doi.org/10.1109/swc62898.2024.00250
7. Mohd, T. K., Bravo-Garcia, F., Love, L., Gujadhur, M., Nyadu, J. (2023). Analyzing Strengths and Weaknesses of Modern Game Engines. *International Journal of Computer Theory and Engineering, 15 (1),* 54–60. https://doi.org/10.7763/ijcte.2023.v15.1330

8. Juliani, A., Berges, V.-P., Teng, E., Cohen, A., Harper, J., Elion, C. et al. (2020). Unity: A General Platform for Intelligent Agents. *arXiv:1809.02627v2*. https://doi.org/10.48550/arXiv.1809.02627
9. Tunguz, B. (2018). *Big Five Personality Test (Version 1). Kaggle*. Available at: https://www.kaggle.com/datasets/tunguz/big-five-personality-test
10. Bielievtsov, S., Ruban, I., Smelyakov, K., Sumtsov, D. (2018). Network Technology for Transmission of Visual Information. *XVIII International Scientific and Practical Conference "Information Technologies and Security" (ITS 2018),* 160–175. Available at: https://ceur-ws.org/Vol-2318/paper14.pdf
11. Cho, D.-S., Cho, J.-M., Kim, W.-T. (2025). Guided deep reinforcement learning framework using automated curriculum scheme for accurate motion planning. *Engineering Applications of Artificial Intelligence, 139*, 109541. https://doi.org/10.1016/j.engappai.2024.109541
12. Zheng, J., Kurt, M. N., Wang, X. (2024). Stochastic Integrated Actor–Critic for Deep Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems, 35 (5),* 6654–6666. https://doi.org/10.1109/tnnls.2022.3212273
13. Alves, J. C., Silva, D. M. d., Mateus, G. R. (2021). Applying and Comparing Policy Gradient Methods to Multi-echelon Supply Chains with Uncertain Demands and Lead Times. *Artificial Intelligence and Soft Computing.* Springer International Publishing, 229–239. https://doi.org/10.1007/978-3-030-87897-9_21
14. Seshagiri, S., Prema, K V. (2023). An Empirical Study of On-Policy and Off-Policy Actor-Critic Algorithms in the Context of Exploration-Exploitation Dilemma. *2023 International Conference on Emerging Techniques in Computational Intelligence (ICETCI).* IEEE, 238–243. https://doi.org/10.1109/icetci58599.2023.10331400
15. Chen, Y., Ying, F., Li, X., Liu, H. (2023). Deep Reinforcement Learning in Maximum Entropy Framework with Automatic Adjustment of Mixed Temperature Parameters for Path Planning. *2023 7th International Conference on Robotics, Control and Automation (ICRCA).* IEEE, 78–82. https://doi.org/10.1109/icrca57894.2023.10087467
16. Zhao, M., Shimosaka, M. (2025). Stable Inverse Reinforcement Learning via Leveraged Guided Motion Planner for Driving Behavior Prediction. *IEEE Access, 13,* 87313–87326. https://doi.org/10.1109/access.2025.3570957
17. Duoxiu, H., Wenhan, D., Wujie, X., Lei, H. (2022). Proximal Policy Optimization for Multi-rotor UAV Autonomous Guidance, Tracking and Obstacle Avoidance. *International Journal of Aeronautical and Space Sciences, 23 (2),* 339–353. https://doi.org/10.1007/s42405-021-00427-2
18. Oh, S., Kim, W., Kim, H. (2024). A Temporally Correlated Latent Exploration for Reinforcement Learning. *arXiv.* https://doi.org/10.48550/arxiv.2412.04775
19. Jarrett, D., Tallec, C., Altché, F., Mesnard, T., Munos, R., Valko, M. (2023). Curiosity in Hindsight: Intrinsic Exploration in Stochastic Environments. *Proceedings of the 40th International Conference on Machine Learning. PMLR, 202,* 14780–14816. https://proceedings.mlr.press/v202/jarrett23a.html
20. Yin, H., Su, S., Lin, Y., Zhen, P., Festl, K., Watzenig, D. (2024). Random Network Distillation Based Deep Reinforcement Learning for AGV Path Planning. *2024 IEEE Intelligent Vehicles Symposium (IV),* 2667–2673. https://doi.org/10.1109/iv55156.2024.10588651
21. Solomon, E. (2025). Autonomous agents: Augmenting visual information with raw audio data. *PLOS One, 20 (5),* e0318372. https://doi.org/10.1371/journal.pone.0318372
22. Cohen, A., Teng, E., Berges, V.-P., Dong, R.-P., Henry, H., Mattar, M. et al. (2022). On the Use and Misuse of Absorbing States in Multi-agent Reinforcement Learning. *RL in Games Workshop AAAI 2022.* Available at: http://aaai-rlg.mlanctot.info/papers/AAAI22-RLG_paper_32.pdf
23. Lowe, R., Wu, Y., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I. (2017). Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *31st Conference on Neural Information Processing Systems (NIPS 2017).* Long Beach. Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/68a9750337a418a86fe06c1991a1d64c-Paper.pdf
24. Lin, Y., Xiao, L., Tao, Y., Zhang, Y., Shu, F., Li, J. (2025). Multi-Agent Computing-Energy-Efficiency Optimization in Vehicular Edge Computing: Non-Cooperative Versus Cooperative Solutions. *IEEE Transactions on Wireless Communications, 24 (7),* 5461–5476. https://doi.org/10.1109/twc.2025.3547377
25. Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., Madry, A. (2020). Implementation Matters in Deep RL: A Case Study on PPO and TRPO. *International Conference on Learning Representations.* Available at: https://iclr.cc/virtual_2020/poster_r1etN1rtPB.html
26. LeCun, Y., Bottou, L., Orr, G. B., Müller, K.-R. (1998). Efficient BackProp. *Neural Networks: Tricks of the Trade.* Berlin, Heidelberg: Springer, 9–50. https://doi.org/10.1007/3-540-49430-8_2
27. Wang, Z., Cai, S., Liu, A., Jin, Y., Hou, J., Zhang, B. et al. (2025). JARVIS-1: Open-World Multi-Task Agents With Memory-Augmented Multimodal Language Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 47 (3),* 1894–1907. https://doi.org/10.1109/tpami.2024.3511593
28. Mantella – Bring NPCs to Life with AI (2023). *ArtFromTheMachine.* Available at: https://www.nexusmods.com/skyrimspecialedition/mods/98631

✉***Andrii Dolhyi,*** *Department of Software Engineering, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, e-mail: andrii.dolhyi.social@gmail.com, ORCID: https://orcid.org/0009-0005-5412-891X*

Kirill Smelyakov, *Doctor of Technical Sciences, Professor, Member of STC, Deputy Head of the Section 2 of STC, Head of Department of Software Engineering, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, ORCID: https://orcid.org/0000-0001-9938-5489*

Yuri Romanenkov, *Doctor of Technical Sciences, Professor, Vice-Rector for Scientific Work, Deputy Head of STC, Professor of the Department of Electronic Computers, Professor of Department of Economic Cybernetics and Management of Economic Security, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, ORCID: https://orcid.org/0000-0002-6544-5348*

Anastasiya Chupryna, *Candidate of Technical Sciences, Associate Professor, Department of Software Engineering, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, ORCID: https://orcid.org/0000-0003-0394-9900*

✉*Corresponding author*