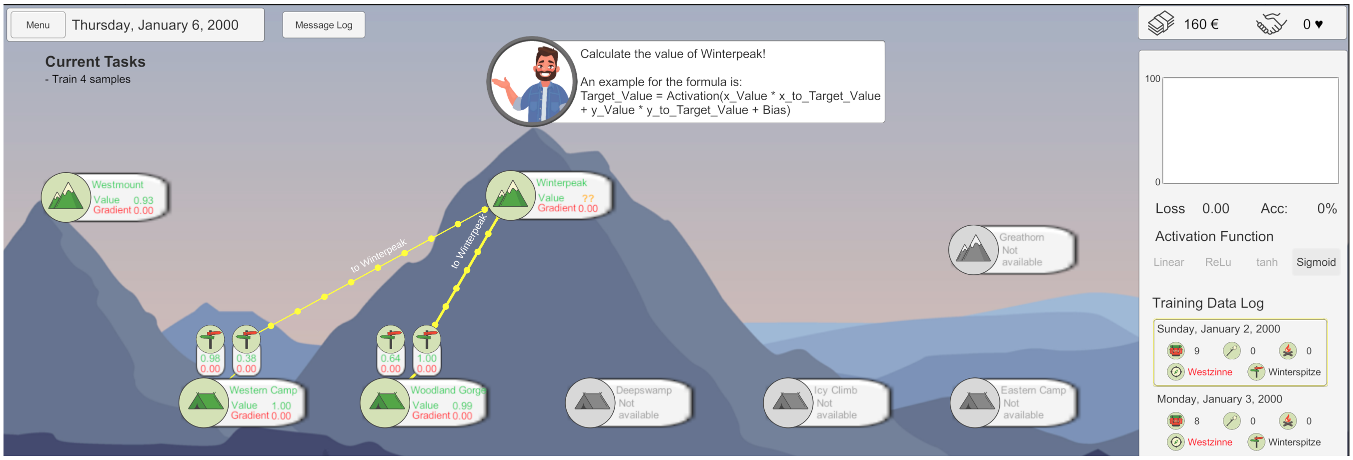
Traversing the Pass: Improving the Knowledge Retention of Serious Games Using a Pedagogical Agent

Philipp Krop
philipp.krop@uni-wuerzburg.de
Human-Computer Interaction Group,
University of Würzburg
Würzburg, Bavaria, Germany

Sebastian Oberdörfer
sebastian.oberdoerfer@uni-wuerzburg.de
Human-Computer Interaction Group,
University of Würzburg
Würzburg, Bavaria, Germany

Marc Erich Latoschik
marc.latoschik@uni-wuerzburg.de
Human-Computer Interaction Group,
University of Würzburg
Würzburg, Bavaria, Germany



Current Tasks
- Train 4 samples

Calculate the value of Winterpeak!

An example for the formula is:
Target_Value = Activation(x_Value * x_to_Target_Value + y_Value * y_to_Target_Value + Bias)

Figure 1: *Traversing the Pass*, a serious game teaching the basics of machine learning together with the pedagogical agent *Tom*.

ABSTRACT

Machine learning is an essential aspect of modern life that many educational institutions incorporate into their curricula. Often, students struggle to grasp how neural networks learn. Teaching these concepts could be assisted with pedagogical agents and serious games, which both have proven helpful for complex topics like engineering. We present "Traversing the Pass," a serious game that utilizes a mentor-like agent to explain the underlying machine learning concepts and provides feedback. We optimized the agent's design in a pre-study before evaluating its effectiveness compared to a text-only user interface with experts and students. Participants performed better in a second assessment two weeks later if they played the game using the agent. Although criticized as repetitive, the game created an understanding of basic machine learning concepts and achieved high flow values. Our results indicate that agents could be used to enhance the beneficial effects of serious games with improved knowledge retention.

CCS CONCEPTS

• **Computing methodologies** → **Intelligent agents**; *Machine learning*; • **Applied computing** → **Interactive learning environments**; • **Theory of computation** → *Convergence and learning in games*.

KEYWORDS

Pedagogical Agents, Serious Games, Machine Learning

ACM Reference Format:
Philipp Krop, Sebastian Oberdörfer, and Marc Erich Latoschik. 2023. Traversing the Pass: Improving the Knowledge Retention of Serious Games Using a Pedagogical Agent. In *ACM International Conference on Intelligent Virtual Agents (IVA '23), September 19–22, 2023, Würzburg, Germany.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3570945.3607360

1 INTRODUCTION

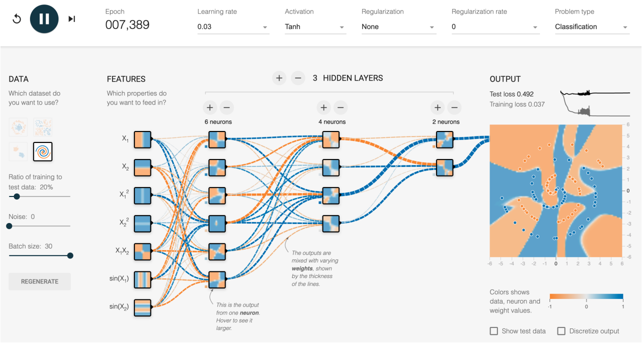
Pedagogical Agents are beneficial in learning applications, where they support learning by offering explanations and guidance, giving feedback, and motivating students. They are usually represented in text form accompanied by a 2D or 3D representation [28]. They positively affect learning performance [17, 20] and overall learning outcome [28], and have been shown to increase student motivation [19, 20] and the meaningfulness of learning applications [2, 30]. These effects are especially useful in more complex topics like engineering [4, 17]. Thus, they could be beneficial to foster an understanding of hard-to-grasp engineering concepts, such as

IVA '23, September 19–22, 2023, Würzburg, Germany
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9994-4/23/09.
https://doi.org/10.1145/3570945.3607360

machine learning. The demand for machine learning experts is ever-growing, and it is now one of the most important topics in science, with over a million articles available on *Google Scholar* since 2017 [13]. Already, many schools and universities teach it in their curricula. Unfortunately, students perceive machine learning as hard to learn and understand. Lavesson [22] found that computer science students find evaluating and comparing different models hard to understand. Often, students and experts use frameworks where they get no information about how the training works. Thus, they treat these frameworks as black boxes and experiment with the parameters until they get the desired result [36]. Thus, pedagogical agents could be used to improve how basic machine learning concepts are taught, provide immediate feedback for exercises, and increase motivation in struggling students.

A suitable learning application to use pedagogical agents with could be *Serious Games*, which are games used for purposes other than mere entertainment, e.g., learning [10, 45]. These games envelop the concepts they want to teach with game mechanics [11, 34] and require the players to repeatedly apply them, supporting learning by repetition and generating flow [8, 12]. Furthermore, repeated successes during gameplay and overall interactivity foster learning [9, 43] and improve the high-level understanding of the enveloped concepts [31]. Serious games have also effectively been used in teaching complex topics, like AI [42], logistics [25], computational thinking [23], debugging [47], geometry [33], and affine transformations [32]. Thus, combining the beneficial effects of pedagogical agents and serious games could be suitable to improve how basic machine learning concepts are taught.

Contribution: This work explores a pedagogical agent's effect on the learning outcome of a serious game that teaches machine learning concepts. We present a serious game called "Traversing the Pass" that teaches the basics of machine learning using a pedagogical agent as displayed in Figure 1. The game was designed to be similar to neural network visualizers, enhanced with gameplay elements and a pedagogical agent. In a pre-study with machine-learning experts, we gathered qualitative feedback to improve the game and let them vote on an agent from which they would prefer to learn machine learning. The agent and its behavior were designed according to guidelines for pedagogical agents. We evaluated the influence of the pedagogical agent in a study with machine learning students currently enrolled in a machine learning course at university level. Players played the game either accompanied by the agent or a depersonalized popup. A knowledge test about the basics of machine learning was administered before, directly after, and two weeks after playing the game. The evaluation revealed that participants retained significantly more knowledge if the agent accompanied them during the experiment but showed no significant differences otherwise. Although participants rated the user experience as below average, the game achieved high flow values and a medium task load, indicating optimal learning conditions. Our results show that a pedagogical agent can enhance the beneficial effects of serious games with improved knowledge retention.


Epoch 007,389

Learning rate 0.03

Activation Tanh

Regularization None

Regularization rate 0

Problem type Classification

Ratio of training to test data: 20%

Noise: 0

Batch size: 30

Test loss 0.492

Training loss 0.037

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

This is the output from one **neuron**. Hover to see it larger.

Colors shows data, neuron and weight values.

☐ Show test data

☐ Discretize output

Figure 2: The user interface of a visualizer called `Tensorflow Playground`. The neural network, its neurons, and their connections all visualize their current value. Users can change the inputs, learning rate, and other values and directly observe the impact of their changes.

2 RELATED WORK

Pedagogical agents have already been successfully used in serious games and educational applications. One early pedagogical agent is *Adele* [41], a 2D agent that helps users to learn about medicine. She was implemented as a conversational bot with a 2D visualization that could have various states. The agent could explain medicine topics, hold simple conversations about medicine, and offer quizzes to deepen knowledge. The authors found that the agent increases motivation, especially through a realistic appearance and facial expressions. Since then, different authors have tried to develop design guidelines and evaluate the benefits of such agents further, showing that the most important aspects are the visualization and the agent's role. Baylor [2] defined three prominent roles for pedagogical agents: The motivator, the mentor, or the expert. A motivator-type agent is primarily used to motivate students, while an expert just explains topics and can answer questions. The mentor type does both and is often implemented as a guide, peer, or co-learner that builds a relationship with the student. It has been shown to increase the positive effect on the learning outcome and motivation and is perceived as more humanlike and supportive [5, 19], making it the overall best choice.

Baylor and Ryu [4] tested different visualizations for pedagogical agents regarding their coolness, age, and attractiveness. They let participants rank them and then decide on one agent from which they want to learn from about specific concepts. For engineering, most participants selected a male, middle-aged agent, which they rated as uncool and non-attractive. Thus, a similar visualization could be suitable for a serious game about machine learning. Recently, a meta-study by Martha and Santoso [28] revealed that pedagogical agents are usually implemented as text-based (72%), with either a 2D or 3D visualization. Often, the agent's appearance is able of gestures, facial expressions, and simple emotions/states. A pedagogical agent led to better learning results in 76% of analyzed papers and improved student behavior in 50% of papers, highlighting the benefit such agents could bring. Especially in big or complex environments, pedagogical agents help build spatial models by showing players where to find relevant UIs or locations [17]. Thus, a text-based mentor-like pedagogical agent could be beneficial to enhance the beneficial properties of serious games.

2.1 Visualizers

Learning how neural networks learn is perceived as hard to understand. A survey among instructors of machine learning courses revealed that higher learning goals, like integrating the aspects of a neural network or analyzing and comparing a network to another, are the areas most students struggle with [35, 44]. Thus, some applications to support teaching machine learning were already created in the scientific and commercial fields. A common way to show the underlying mechanics of a neural network is with so-called visualizers. These visualizers display the underlying neural network as a graph with interconnected paths (see Figure 2). Usually, the weights and values of each node can be seen at a glance. Some applications, like `GAN-Lab` [7] or `Tensorflow Playground` [43], visualize the data transformation by showing how data is distributed at intermediate steps. The line thickness of connections is often used to indicate the underlying weight: The thicker the line, the bigger the weight. These visualizers, however, are limited to this basic functionality. They don’t have any underlying game mechanics or gamified aspects and thus miss the motivation-enhancing effects these mechanics can bring.

2.2 Serious Games

Several serious games to teach basic machine learning mechanics were developed to overcome these shortcomings. `ML-Quest` [37] is a serious game teaching basic machine learning concepts by encoding them in tasks like finding the exit of a maze by repeatedly following instructions (supervised learning) or by finding the steepest slope (gradient descent). An evaluation with high school students showed that the game was perceived well, but only 42% of the participants knew about the underlying machine learning concepts after playing the game. `ArtBot` [48] is a serious game about classification and reinforcement learning where the players must train a robot to recover stolen artwork from a dungeon. They introduced *Mad-lib explainers*, which are tooltips next to each learning parameter that explain how changing the parameter would affect the agent’s behavior. An evaluation with 130 students and 17 teachers revealed that although participants liked the ease of use, the graphics, the customization options, and the explanations of specific systems, they also perceived the game as boring and monotonous and would have liked additional material about the parameters and the underlying concepts. There are two notable games on the commercial side: `While True: Learn()` [26], where players have to solve increasingly complex sorting puzzles with various sorting algorithms, and `Learning Factory` [27], where players have to manage a factory and calculate the prices of their goods using machine learning algorithms. Both games offer short text descriptions about machine learning concepts and longer, more detailed video explanations of underlying concepts.

Unfortunately, all these games do not visualize the neural network as a graph with interconnected paths, which visualizers excel in. They are thus losing the capability to visualize intermediate steps, which is valuable for students learning machine learning. To compensate, they provide the user with further information on demand, like tooltips or instructional videos, which require the player to interact with them to get the information. The inner workings of these networks are not explored during play and thus remain a black box. A learning application integrating the informational clarity of visualizers, the gaming aspects of serious games, and a pedagogical agent could be suitable to overcome these shortcomings.

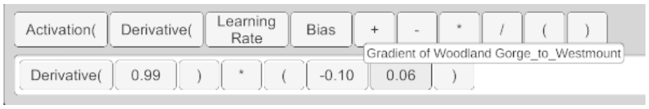


Figure 3: The calculator which students used to build the formulas of the forward and backward pass.

3 SYSTEM DESCRIPTION

We present *Traversing the Pass*, a 2D serious game where players have to train a neural network for a travel agency that predicts the destination of different tourist groups. The game envelops the backpropagation algorithm in various game mechanics and teaches students about other related machine-learning concepts like the learning rate and different activation functions. The game’s map is inspired by visualizers, showing all neurons, connections, and their current value at a glance. A pedagogical agent accompanies players during gameplay to showcase the user interface, explain the theoretical concepts, and give feedback. It introduces the players to the theoretical background in increasingly challenging scenarios like calculating the output of a specific node during forward propagation. Players must solve these challenges by building the correct formula from code building blocks (see Figure 3). These challenges reward the player with two currencies, *money* and *likes*, which they can spend to unlock visual upgrades and additional features for the neural network, like different activation functions. The players win the game by completing all challenges and milestones. The game is available in German and English. An overview of the game’s user interface can be seen in Figure 1.

3.1 Gameplay Loop

The gameplay consists of two phases: *training* and *prediction*. The player must manually train the neural network in the *training* phase. To do so, a pedagogical agent leads them through the four components of the forward pass and backpropagation one after another to incrementally increase the difficulty. Their first challenge is to calculate the value of an output node, then calculate the gradient of a weight, then calculate the value of a node in the hidden layer, and finally, calculate the new value of a weight. When the player encounters a challenge for the first time, the agent explains the theory in detail before providing an example formula. After players enter their solution, it states whether it is correct or explains the correct solution so that players can learn from their errors. The players must complete these challenges for four training samples, each with different in- and outputs, after which the training phase ends. In the *prediction* phase, the neural network predicts newly generated sample data, replacing the old training data. The game rewards the player with money for each prediction and likes for each correct prediction, incentivizing players to improve their neural network. They can spend these currencies on visual upgrades, alternative activation functions, and increased data storage.

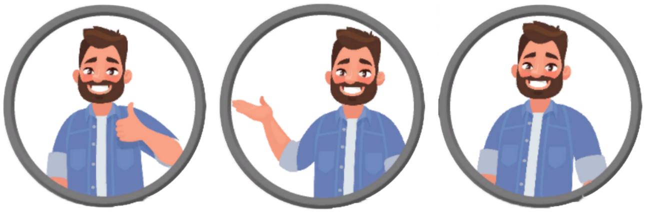


Figure 4: The three states the agent can have, which show them being happy, explaining, or idle.

The game supports learning processes in a multitude of ways. First, further information about the theory behind each element is available on demand: The user can hover over each machine learning term to show a tooltip explaining the concept behind the term, the purpose of the concept, and the formula, if available. Second, the pedagogical agent explains each part of the underlying theory at least once. He shows a relevant example formula each time a player has to solve a challenge. These challenges are introduced iteratively: In the first data sample, players are only introduced to the forward pass challenge. In the second sample, they must complete the forward pass challenge again before the pedagogical agent introduces them to the second challenge, calculating the weight gradient. This sequential introduction and the tooltips allow students to exercise the concepts after they are introduced and build on constant repetition to solidify knowledge. The game gives the players constant feedback and the feeling of success every time they complete a challenge. It is intended to motivate students in conjunction with reward systems, such as earning money that players can spend on improvements.

3.2 Pedagogical Agent

Tom, the pedagogical agent, is introduced as a colleague that helps the players on their first day in the agency and supports them while implementing this neural network. He falls in the *mentor/peer* category of pedagogical agents, which is more successful in supporting learning than an *expert* or a pure *motivator* [2]. The agent highlights essential user interface elements during the tutorial to explain them in detail. He then explains each challenge once so students get a basic understanding of the learning contents. During a challenge, he showcases an example formula to help students solve the challenge. If a student gets something wrong, he explains what the correct solution would have been. The agent is represented as an image with three states: happy, explaining, and idle (see Figure 4). We opted against a sad state to not discourage students if they get something wrong. He communicates with the player solely by text. Each time he explains something, the message slides in from the top, and a sound is played to catch the user's attention. A log of all the agents prior messages is available on demand.

3.3 Technology

The game was created using the *Unity Engine 2021.3.11f1* [46]. It runs on every state-of-the-art Windows PC. In the study, a PC with *32 GB of RAM*, an *Intel i7-9700k* processor, an *Nvidia RTX 2080* graphics card, and *Windows 10* was used. We used *Superlux HD-330* headphones and a *Lenovo ThinkVision P27h-20* monitor.

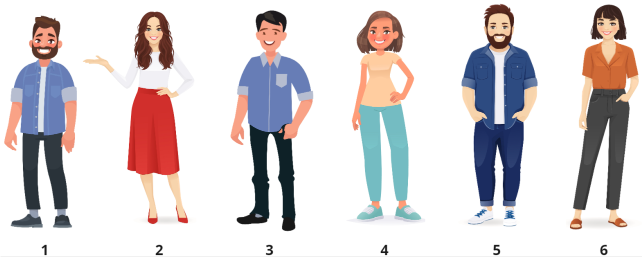


Figure 5: Six possible visualizations for the pedagogical agent. Participants voted on who they wanted to learn from about machine learning and then rated their age, coolness, and attractiveness. In the end, visualization one was selected to represent the game's pedagogical agent.

4 METHODS

We performed two evaluations: First, we conducted a prestudy, where we iteratively improved the game and let participants vote on the agent's appearance. After improving the game based on the feedback collected in the prestudy, we evaluated the agent's effectiveness compared to a depersonalized popup. The survey tool *LimeSurvey* [24] was used for hosting the questionnaires.

4.1 Prestudy

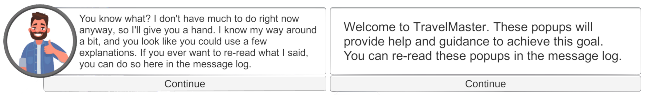
A pre-study was performed that served two purposes: First, we gathered qualitative feedback with the *Thinking Aloud* [6] method during the study, which we used to improve the game iteratively. The second purpose was to find a suitable appearance for the agent. Similar to the study of Baylor and Ryu [4], we let participants vote on different visualizations for pedagogical agents regarding their coolness, age, and attractiveness and if they would like to learn machine learning from them. We pre-selected six different visualizations (three male, three female) from *Adobe Stock* [1] (as seen in Figure 5). Six participants (four female, two male) who had already completed a machine learning course participated in the pre-study. We refrained from gathering demographic data since they could be used to identify the participants. We found no difference between the age and coolness of the visualizations, but visualization six was perceived as more attractive and thus excluded from further consideration. Visualizations one and five received two votes, three and six received one vote, and visualizations two and four received zero votes. Since visualization one and five did not differ significantly from another, we randomly picked visualization one.

4.2 Study Design

Based on the presented related work, we expected two outcomes:

(1) Students show better learning outcomes after playing the game accompanied by a pedagogical agent
(2) Students score higher on intrinsic motivation after playing the game accompanied by a pedagogical agent

To evaluate this, we conducted a between-subjects study, where participants were either accompanied by a pedagogical agent or a depersonalized popup. Contrary to the pedagogical agent, the popup had no 2D representation and consisted of only text without



You know what? I don't have much to do right now anyway, so I'll give you a hand. I know my way around a bit, and you look like you could use a few explanations. If you ever want to re-read what I said, you can do so here in the message log.

Welcome to TravelMaster. These popups will provide help and guidance to achieve this goal. You can re-read these popups in the message log.

a) Pedagogical Agent

b) Depersonalized Popup

Figure 6: The experiment conditions: Participants were either accompanied by a) a pedagogical agent or b) a depersonalized popup while playing the game.

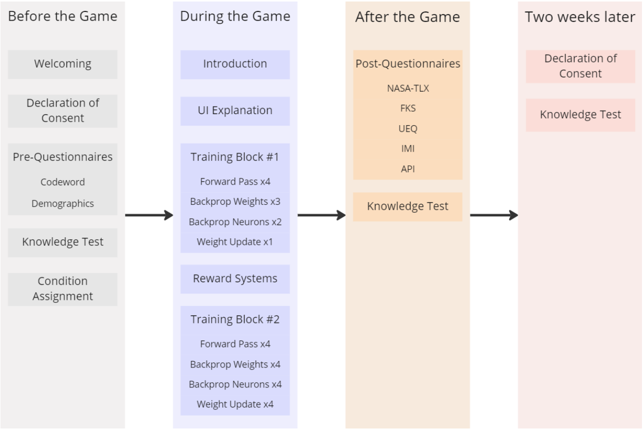


Figure 7: The procedure of the study. Participants gave their consent, filled in pre-questionnaires, and answered a knowledge test. They then played the game and entered 26 machine learning formulas. Afterward, they filled in post-questionnaires and another knowledge test. They received another knowledge test two weeks later.

any personal notes (see Figure 6). Each participant was evaluated twice: On the first date, they played the game at the university and filled out questionnaires and knowledge tests. For the second evaluation, they received an online questionnaire, where they had to fill in another knowledge test. Participants provided written consent to participate in the study voluntarily and were rewarded with 15€. The study was approved by the ethics committee of the university.

4.3 Measures

Learning Effects. We logged how long each participant needed for each training block and how many of the 26 available tasks they got correct. Also, all participants had to complete three knowledge tests, each resembling typical exercises in teaching machine learning, e.g., calculating the value of an output neuron and the gradient of a connection, before answering three open questions. Answers for the first two questions were marked as correct if participants either entered the value or the correct formula to calculate it. We administered two different tests: One before participants played the game to see if they knew disproportionally more about the basics of machine learning than the other participants, and a second one directly after the game and two weeks later to see how much knowledge participants remembered after two weeks. Participants had ten minutes to answer each knowledge test. This second test asked participants to answer the following questions:

(1) In front of you, you see a neural network, including connections and weights. The bias is always 1. The linear activation function is used. The input is (2,1). Please enter the value of node O2.
(2) The error of the network is 1. Both the error of O1 and O2 is 0.5. Calculate the gradient of the connection from O1 to L1.
(3) What is the purpose of the learning curve?
(4) What are the four parts of training?
(5) What is the Accuracy?

Motivation. We used the *Intrinsic Motivation Inventory (IMI)* [29] to measure if there is an effect on the intrinsic motivation of participants. It consists of 45 items encoded with a 7-point Likert scale ranging from one ('Not true at all') to seven ('Very true'), which build the subscales *Interest/Enjoyment, Perceived Competence, Effort/Importance, Pressure/Tension, Perceived Choice, Value/Usefulness*, and *Relatedness*. The subscale *Interest/Enjoyment* is considered the primary measure of intrinsic motivation, whereas *Pressure/Tension* is seen as a negative predictor.

Agent. To measure how the pedagogical agent is perceived, participants had to fill out the *Agent Persona Instrument (API)* [3], which measures the subscales *Facilitating Learning*, *Credible*, *Human-like*, and *Engaging* on, in total, 25 items on a five-point Likert scale from one to five each. The construct *Informational Usefulness* can be built by combining the subscales *Facilitating Learning* and *Credible*, and the other two subscales build the construct *Affective Interaction*. To be able to compare both conditions, they should yield similar values in *Informational Usefulness*, and the agent should have higher values in *Affective Interaction* than a depersonalized popup.

Task Load. We measured the task load with the raw version of the *NASA Task Load Index (NASA-TLX)* [14] to ensure that our game challenges the students but also leaves cognitive resources free for learning. The questionnaire consists of the six subscales *Mental Demand*, *Physical Demand*, *Temporal Demand*, *Performance*, *Effort* and *Frustration*, each ranging from zero (low) to 100 (high). A total score for task load is created by calculating the sum of all subscales.

User Experience. Finally, we measured the user experience of the game with the *User Experience Questionnaire (UEQ)* [21], which consists of 26 opposing word pairs, e.g., 'good' vs. 'bad', each rated on a seven-point Likert scale ranging from one word (-3) to the other (3). The subscales *Attractiveness*, *Perspicuity*, *Efficiency*, *Dependability*, *Stimulation* and *Novelty* can be constructed from these items, measuring the overall user experience of the product, how difficult it is to get familiar with it, how easily the users can work with it, how in control a user feels, how motivating the interactions are, and how creative the design feels.

Flow. We used the *Flow Short Scale (FSS)* to measure flow [39]. It consists of thirteen items measured on a seven-point Likert scale from one to seven, which build the subscales *Fluency of Performance*, *Absorption by Activity*, and *Perceived Importance*, and three additional items, which measure the subscale *Fit of demand and skills*. A total score can be calculated from *Fluency of Performance* and *Absorption by Activity*, with scores above four indicating high flow.

4.4 Procedure

First, participants were welcomed to the study and thanked for their willingness to partake. They then signed a consent form and generated a code word before filling out a demographic questionnaire and a knowledge test. Then, they played the game, where they learned about the basics of machine learning and had to practice each formula at least five times. Afterward, they filled in questionnaires and another knowledge test. Two weeks later, they received an online questionnaire where they had to fill in the second knowledge test again. In total, the experiment took about one and a half hours.

4.5 Participants

We recruited participants by contacting current and former students of various machine learning courses at the University of Würzburg. Sixteen students (eleven male, five female) participated in the main study. One male participant had to be excluded because of a bug that prohibited him from finishing the game. The fifteen remaining participants were between 21 and 31 years old ($M = 25.47, SD = 2.50$) and were separated into the *agent* ($n = 7$) and the *popup* ($n = 8$) condition. Five had a vision impairment, which was corrected with glasses, and no participant was colorblind or hearing impaired. All participants were right-handed and spoke German as their mother tongue. They used the internet and the computer daily and played video games at least once a month. Thirteen participants completed the second test two weeks later.

5 RESULTS

We used *RStudio* [38] with *R 4.1.2* [16] for statistical analysis. We calculated independent t-tests for comparisons between participants with a significance level of $\alpha = .05$. We found no gender differences between the conditions. The means, standard deviations, and statistical analyses are displayed in Table 1.

Learning Effects. We assumed higher learning outcomes for participants in the agent condition. Thus, we calculated one-sided t-tests for the knowledge test and correctly answered in-game tasks. We found no significant difference between the agent and popup conditions in all knowledge tests. However, we found a significant difference between the second and third knowledge tests with a large effect, $t(10.67) = 2.02, p < 0.05, d = 1.09$, indicating that knowledge retention was higher in the agent condition. The assumption of normality was violated for the amount of correctly answered in-game tasks, so we used the robust Yuen's test from the `WRS2` package instead. However, we did not find any significant difference between the conditions regarding the amount of correctly answered in-game tasks.

Motivation. We used a one-sided t-test to calculate *Interest /Enjoyment* since we expected the agent to influence motivation. Although we found a medium effect on *Interest/Enjoyment*, students did not show significantly higher values in the agent condition. We found no significant difference for *Pressure/Tension*.

Agent. We found no significant difference between the conditions for *Informational Usefulness*. However, we found a large effect on *Affective Interaction*, with significantly higher ratings in the agent condition, $t(8.67) = 2.71, p < 0.05, d = 1.46$.

Task Load. There was no significant difference in the overall score of the *NASA-TLX* nor in the *Mental Demand* subscale. The overall task load of the game is in the typical range for educational applications, which is 47 ± 7 [15].

User Experience. We found no significant differences between the conditions for all subscales of the *UEQ*. The overall values indicate that the game's user experience is *below average* [40].

Flow. Overall, the game achieved medium to high flow, with medium to high values in the agent condition and low to medium values in the popup condition. We found a medium, but not significant, effect for both the overall flow score and the *Perceived Importance*. There was no significant difference regarding the *Fit of demand and skills*.

6 DISCUSSION

Our results look promising but need further work. Students found that both the agent and the depersonalized popup were equally useful in teaching the concepts, indicated by the mid to high values in *Informational Usefulness* in both conditions. The *Affective Interaction* subscale of the API showed that the agent was perceived as being more humanlike and engaging than the popup, thus satisfying the manipulation check. The game elicited high flow, which positively influences learning outcomes [18], and a medium task load, which is typical for learning applications [15].

6.1 Learning Effects & Motivation

In theory, participants should perform better in knowledge tests after playing the game with a pedagogical agent [17, 20]. This was only partially so. We found no significant difference between the groups for the second and third knowledge tests, showing that the pedagogical agent did not increase the direct learning outcome. However, the performance of students deteriorated more between the second and third tests in the popup condition, indicating that participants in the agent condition retained significantly more learning content. We suspect that this is due to the agent being perceived as more human-like. The interactions with the agent are more direct. Students learn personal information about the agent, like where it works and what its personality is like, thus possibly accepting it as a social interaction partner. Thus, we think that students built a trusting relationship with the agent and that this interpersonal communication gives the learning content more meaning, aligning with previous work that found that pedagogical agents make the learning experience more meaningful [2, 30].

Playing a serious game with a pedagogical agent should also increase intrinsic motivation [19, 20]. Our results, however, do not support this. We found high intrinsic motivation and negative pressure for both conditions, but no statistical difference. We attribute this to the simplicity of our pedagogical agent since it only consisted of texts and an image with various states. A more advanced agent that is animated, voiced, and overall more prominent could have a bigger impact on intrinsic motivation, as previous work has shown [19, 20]. We have thus shown that even a simple pedagogical agent consisting of text and an image can enhance the beneficial effects of serious games with improved knowledge retention.

	Overall	**Agent**	**Popup**	**df**	**t**	**p**	**Effect Size**
	M (SD)	*M (SD)*	*M (SD)*				*d*
Learning Effects							
First Knowledge Test	1.80 (1.32)	1.86 (1.57)	1.75 (1.16)	10.97	.15	.885	.08
Second Knowledge Test	2.67 (1.23)	2.57 (1.27)	2.75 (1.28)	12.76	-.27	.791	-.14
Third Knowledge Test	2.23 (1.59)	2.67 (1.86)	1.86 (1.35)	8.98	.89	.399	.51
Change between Test 2 and 3	-.38 (1.04)	.17 (.75)	-.86 (1.07)	10.67	2.02	.035*	1.09
*Correct Solutions**	19.93 (3.97)	20.43 (5.21)	19.50 (5.21)	6.24	.15	.885	.15
Motivation (IMI)							
Interest/Enjoyment	4.46 (.86)	4.69 (1.00)	4.25 (.71)	10.70	.98	.175	.52
Pressure/Tension	2.56 (1.18)	2.54 (1.09)	2.57 (1.32)	12.96	-.05	.960	-.03
Agent (API)							
Informational Usefulness	3.79 (.62)	3.66 (.73)	3.91 (.53)	10.84	-.77	.459	-.41
Affective Interaction	2.73 (1.16)	3.47 (1.23)	2.09 (.62)	8.67	2.71	.013*	1.46
Task Load (NASA-TLX)							
Mental Demand	73.00 (14.74)	76.43 (16.51)	70.00 (13.36)	11.59	.82	.428	.43
Overall Task Load	41.44 (6.13)	43.69 (5.35)	39.48 (6.41)	12.98	1.39	.189	.71
User Experience (UEQ)							
Attractivity	.40 (.31)	.45 (.21)	.35 (.39)	10.93	.61	.551	.31
Perspicuity	-.32 (.41)	-.43 (.31)	-.22 (.47)	12.22	-1.03	.32	-.52
Efficiency	.05 (.52)	.14 (.50)	-.03 (.56)	12.99	.64	.534	.33
Dependability	.17 (.40)	.14 (.40)	.19 (.42)	12.83	-.21	.837	-.11
Stimulation	-.22 (.45)	-.25 (.41)	-.19 (.51)	12.91	-.26	.797	-.13
Novelty	-.33 (.56)	-.18 (.59)	-.47 (.53)	12.18	1.00	.337	.52
Flow (FSS)							
Overall Score	4.47 (.91)	4.69 (.88)	2.67 (1.17)	12.93	.88	.396	.45
Perceived Importance	3.11 (1.76)	3.62 (2.26)	2.67 (1.17)	8.73	1.01	.342	.54
Fit of Demand and Skills	4.56 (.85)	4.48 (.90)	4.63 (.86)	12.56	-.33	.750	-.17

Table 1: The means, standard deviations, and comparisons between the conditions for each measure used in the experiment.

6.2 Limitations and Future Work

We evaluated the agent with machine learning experts and with currently enrolled students, which allowed us to incorporate the feedback of both experts and novices, but imposes the limitation that the study's sample size is quite small. Thus, we plan to incorporate the game into the course and evaluate it over the whole semester. Despite high flow ratings, our results showed that participants rated the game as below average. Qualitative feedback we received after the study shows that participants found the game repetitive and boring. A way to increase motivation could be to include competition [9]. Players could see the ranking of their company compared to other (static) competitors on a leaderboard. To make the gameplay less repetitive, we will also switch from entering the formula for the same neuron every time, to adding the formula once, and then updating it whenever a new neuron is unlocked. This shifts the gameplay to be more user-driven and is akin to factory games like *Learning Factory* [27], where production lines are established and updated each time a new product is added. It would also resemble having to program a neural network from scratch, which is the end goal of typical machine learning courses. Further work will expand the role and capabilities of the pedagogical agent, e.g., by switching to a conversational agent that can react to the game's current state and answer specific questions about machine learning, which should be perceived as more humanlike and is more likely to be accepted as a social interaction partner, thus increasing the positive impact on motivation and learning outcomes.

7 CONCLUSION

We presented *Traversing the Pass*, a game about machine learning that uses a pedagogical agent to enhance the beneficial effects of serious games. In the game, students learn how to calculate the forward and backward pass - the mechanism neural networks use to learn new information - and practice it repeatedly. The mentor-like agent explains every underlying concept, gives feedback, and supports players during the practice sessions. We conducted a pre-study with machine learning experts to iteratively improve the game and to decide on the agent's appearance and evaluated the game with machine learning students, who had to solve 26 learning challenges either accompanied by a pedagogical agent or a depersonalized popup. Overall, the game was received well. Although the user experience was rated as below average, we found that the game generated high amounts of flow and a medium task load. We found no direct effect on learning outcomes and no effect on intrinsic motivation. However, participants performed better in the same knowledge test two weeks later if they used the pedagogical agent compared to a depersonalized popup. Thus, our work indicates that pedagogical agents could be used to enhance the beneficial effects of serious games with improved knowledge retention.

ACKNOWLEDGMENTS

This research was funded by the German Federal Ministry of Labour and Social Affairs [DKI.00.00030.21].

REFERENCES

[1] Adobe. 2023. Adobe Stock. https://stock.adobe.com/ [Online; accessed 17-January-2023].

[2] Amy Baylor. 2000. Beyond Butlers: Intelligent Agents as Mentors. *Journal of Educational Computing Research - J EDUC COMPUT RES* 22 (Sept. 2000), 373–382. https://doi.org/10.2190/1EBD-G126-TFCY-A3K6

[3] Amy Baylor and Jeeheon Ryu. 2003. The API (Agent Persona Instrument) for assessing pedagogical agent persona. In *EdMedia+ innovate learning*. Association for the Advancement of Computing in Education (AACE), 448–451.

[4] Amy Baylor and Jeeheon Ryu. 2005. The API (Agent Persona Instrument) for Assessing Pedagogical Agent Persona. (2005), 448–451.

[5] Amy L Baylor. 2011. The design of motivational agents and avatars. *Educational Technology Research and Development* 59 (2011), 291–300.

[6] Ted Boren and Judith Ramey. 2000. Thinking aloud: Reconciling theory and practice. *IEEE transactions on professional communication* 43, 3 (2000), 261–278.

[7] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. 2018. Generative adversarial networks: An overview. *IEEE signal processing magazine* 35, 1 (2018), 53–65.

[8] Mihaly Csikszentmihalyi and Mihaly Csikzentmihaly. 1990. *Flow: The psychology of optimal experience*. Vol. 1990. Harper & Row New York.

[9] Adrian A. De Freitas and Troy B. Weingart. 2021. I'm Going to Learn What?!?: Teaching Artificial Intelligence to Freshmen in an Introductory Computer Science Course. *SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (2021), 198–204. https://doi.org/10.1145/3408877.3432530

[10] Sara De Freitas and Fotis Liarokapis. 2011. Serious games: a new paradigm for education? In *Serious games and edutainment applications*. Springer, 9–23.

[11] Christopher B Eiben, Justin B Siegel, Jacob B Bale, Seth Cooper, Firas Khatib, Betty W Shen, Barry L Stoddard, Zoran Popovic, and David Baker. 2012. Increased Diels-Alderase activity through backbone remodeling guided by Foldit players. *Nature biotechnology* 30, 2 (2012), 190–192.

[12] James Paul Gee. 2003. What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)* 1, 1 (2003), 20–20.

[13] Google Scholar. 2023. Google Scholar Search for 'Machine Learning' from 2017 to 2023. https://scholar.google.com/scholar?q=machine+learning&hl=en&as_sdt=0%2C5&as_ylo=2017&as_yhi=2023 [Online; accessed 07-January-2023].

[14] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.

[15] Morten Hertzum. 2021. Reference values and subscale patterns for the task load index (TLX): a meta-analytic review. *Ergonomics* 64, 7 (2021), 869–878.

[16] Ross Ihaka and Robert Gentleman. 1996. R: a language for data analysis and graphics. *Journal of computational and graphical statistics* 5, 3 (1996), 299–314.

[17] W. Lewis Johnson, Jeff W. Rickel, and James C. Lester. 2000. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial InTELligence in Education* 11 (2000), 47–78.

[18] Kristian Kiili. 2005. Content creation challenges and flow experience in educational games: The IT-Emperor case. *The Internet and higher education* 8, 3 (2005), 183–198.

[19] Yanghee Kim and Amy L. Baylor. 2016. Research-Based Design of Pedagogical Agent Roles: A Review, Progress, and Recommendations. *International Journal of Artificial Intelligence in Education* 26, 1 (March 2016), 160–169. https://doi.org/10.1007/s40593-015-0055-y

[20] Gonca Kizilkaya and Petek Askar. 2008. The Effect of an Embedded Pedagogical Agent on the Students' Science Achievement. *Interactive Technology and Smart Education* 5, 4 (Jan. 2008), 208–216. https://doi.org/10.1108/17415650810930893

[21] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and usability engineering group*. Springer, 63–76.

[22] Niklas Lavesson. 2010. Learning machine learning: a case study. *IEEE Transactions on Education* 53, 4 (2010), 672–676.

[23] Bobby Law. 2016. Puzzle games: A metaphor for computational thinking. *Proceedings of the European Conference on Games-based Learning* 2016-Janua (2016), 344–353.

[24] LimeSurvey GmbH. 2023. LimeSurvey. https://www.limesurvey.org/ [Online; accessed 17-January-2023].

[25] Chiung Lin Liu. 2017. Using a video game to teach supply chain and logistics management. *Interactive Learning Environments* 25, 8 (2017), 1009–1024. https://doi.org/10.1080/10494820.2016.1242503

[26] Luden.io. 2019. While true: Learn(). https://store.steampowered.com/app/619150/while_True_learn

[27] Luden.io. 2021. Learning Factory. https://store.steampowered.com/app/1150090/Learning_Factory

[28] Ati Suci Dian Martha and Harry Santoso. 2019. The Design and Impact of the Pedagogical Agent: A Systematic Literature Review. *The Journal of Educators Online* 16, 1 (Jan. 2019). https://doi.org/10.9743/jeo.2019.16.1.8

[29] Edward McAuley, Terry Duncan, and Vance V Tammen. 1989. Psychometric properties of the Intrinsic Motivation Inventory in a competitive sport setting: A confirmatory factor analysis. *Research quarterly for exercise and sport* 60, 1 (1989), 48–58.

[30] Kristen N Moreno, Natalie K Person, Amy B Adcock, Richard N Van Eck, G Tanner Jackson, and Johanna C Marineau. 2002. Etiquette and Efficacy in Animated Pedagogical Agents: The Role of Stereotypes. (2002), 4.

[31] Chelsea M. Myers, Jiachi Xie, and Jichen Zhu. 2020. A Game-Based Approach for Helping Designers Learn Machine Learning Concepts. (2020). arXiv:2009.05605 http://arxiv.org/abs/2009.05605

[32] Sebastian Oberdörfer, David Heidrich, and Marc Erich Latoschik. 2019. Usability of gamified knowledge learning in VR and desktop-3D. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[33] Sebastian Oberdörfer, Philipp Krop, Samantha Straka, Silke Grafe, and Marc Erich Latoschik. 2022. Fly My Little Dragon: Using AR to Learn Geometry. In *2022 IEEE Conference on Games (CoG)*. IEEE, 528–531.

[34] Sebastian Oberdörfer and Marc Erich Latoschik. 2019. Predicting learning effects of computer games using the Gamified Knowledge Encoding Model. *Entertainment Computing* 32 (2019), 100315.

[35] Cathy O'neil. 2017. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown.

[36] Kayur Patel, James Fogarty, James A Landay, and Beverly L Harrison. 2008. Examining Difficulties Software Developers Encounter in the Adoption of Statistical Machine Learning.. In *AAAI*. 1563–1566.

[37] Shruti Priya, Shubhankar Bhadra, and Sridhar Chimalakonda. 2021. ML-Quest: A Game for Introducing Machine Learning Concepts to K-12 Students. 1, 1 (2021), 1–13. arXiv:2107.06206 http://arxiv.org/abs/2107.06206

[38] Jeffrey S Racine. 2012. RStudio: a platform-independent IDE for R and Sweave.

[39] Falko Rheinberg, Regina Vollmeyer, and Stefan Engeser. 2003. Die erfassung des flow-erlebens. (2003).

[40] Martin Schrepp. 2019. User experience questionnaire handbook Version 8. *All you need to know to apply the UEQ successfully in your projects* (2019).

[41] Erin Shaw, W Lewis Johnson, and Rajaram Ganeshan. 1999. Pedagogical agents on the web. In *Proceedings of the third annual conference on Autonomous Agents*. 283–290.

[42] Nicole Sintov, Debarun Kar, Thanh Nguyen, Fei Fang, Kevin Hoffman, Arnaud Lyet, and Milind Tambe. 2016. From the lab to the classroom and beyond: Extending a game-based research platform for teaching ai to diverse audiences. *30th AAAI Conference on Artificial Intelligence, AAAI 2016* (2016), 4107–4112.

[43] Daniel Smilkov, Shan Carter, D. Sculley, Fernanda B. Viégas, and Martin Wattenberg. 2017. Direct-Manipulation Visualization of Deep Networks. (2017). arXiv:1708.03788 http://arxiv.org/abs/1708.03788

[44] Elisabeth Sulmont, Elizabeth Patitsas, and Jeremy R. Cooperstock. 2019. Can You Teach Me To Machine Learn? (2019), 948–954. https://doi.org/10.1145/3287324.3287392

[45] Tarja Susi, Mikael Johannesson, and Per Backlund. 2007. Serious games: An overview. (2007).

[46] Unity. 2021. Unity Engine. https://unity.com

[47] Akhila Sri Manasa Venigalla and Sridhar Chimalakonda. 2020. G4D - a treasure hunt game for novice programmers to learn debugging. *Smart Learning Environments* 7, 1 (2020). https://doi.org/10.1186/s40561-020-00129-4

[48] Iro Voulgari, Marvin Zammit, Elias Stouraitis, Antonios Liapis, and Georgios Yannakakis. 2021. Learn to Machine Learn: Designing a Game Based Approach for Teaching Machine Learning to Primary and Secondary Education Students. *Proceedings of Interaction Design and Children, IDC 2021* (2021), 593–598. https://doi.org/10.1145/3459990.3465176

Received 28 April 2023; revised 18 July 2023; accepted 18 July 2023