Explainability via Responsibility

Faraz Khadivpour and Matthew Guzdial
Department of Computing Science, Alberta Machine Intelligence Institute (Amii)
University of Alberta, Canada
{khadivpour, guzdial}@ualberta.ca

Abstract

Procedural Content Generation via Machine Learning (PCGML) refers to a group of methods for creating game content (e.g. platformer levels, game maps, etc.) using machine learning models. PCGML approaches rely on black box models, which can be difficult to understand and debug by human designers who do not have expert knowledge about machine learning. This can be even more tricky in co-creative systems where human designers must interact with AI agents to generate game content. In this paper we present an approach to explainable artificial intelligence in which certain training instances are offered to human users as an explanation for the AI agent's actions during a co-creation process. We evaluate this approach by approximating its ability to provide human users with the explanations of AI agent's actions and helping them to more efficiently cooperate with the AI agent.

Introduction

In science and engineering, a black box is a component that cannot have its internal logic or design directly examined. In artificial intelligence (AI), "The black box problem" refers to certain kinds of AI agents for which it is difficult or impossible to naively determine how they came to a particular decision (Zednik 2019). Explainable artificial intelligence (XAI) is an assembly of methods and techniques to deal with the black box problem (Biran and Cotton 2017). Machine Learning (ML) is a subset of artificial intelligence that focuses on computer algorithms that automatically learn and improve through experience. (Goodfellow, Bengio, and Courville 2016). The current state-of-the-art models in ML, deep neural networks, are black box models. Intuitively, it is difficult to cooperate with an individual when you cannot understand them. This is critical in co-creative systems (also called mixed-initiative systems), in which a human and an AI agent work together to produce the final output. (Yannakakis, Liapis, and Alexopoulos 2014).

There is a wealth of existing methods in the field of XAI (Adadi and Berrada 2018). For example, those that draw

comparisons between the input and the output of a model (Cortez and Embrechts 2011; 2013; Simonyan, Vedaldi, and Zisserman 2013; Bach et al. 2016; Dabkowski and Gal 2017; Selvaraju et al. 2017), or analyze the output in terms of the model's parameters (Boz and Hillman 2000; García, Fernández, and Herrera 2009; Letham et al. 2015; Hara and Hayashi 2018). Alternatively, there is the strategy to attempt to simplify the model (Che et al. 2015; Tan et al. 2017; Xu et al. 2018). The major difference between our approach and these previous ones is that we present a method which makes it possible to explain an AI agent's action through a detailed inspection of what it has learned during the training phase.

Questions we might want to ask an AI agent include "How did you learn to do that action?" or "What did you learn that led you to make that decision?" (Cook et al. 2019). We sought to develop an approach that could answer these questions. Thus, our approach needed to find explanations for the AI agent's decisions based on its training data.

In this paper, we make use of the problem domain of a co-creative Super Mario Bros. level design agent. We use this domain since XAI is critical in co-creative systems. We introduce an approach to detect the training instance that is *most responsible* for an AI agent's action. We can then present the most responsible training instance to the human user as an answer to how the AI agent learned to make a particular decision. To evaluate this approach we compare the quality of these responsible training instances to random instances as explanations in two experiments on existing data.

Related Work

Our problem domain is generating explanations for a PCGML co-creative agent. Therefore we separate the prior related work into three main areas: Procedural Content Generation via Machine Learning (PCGML), co-creative systems, and Explainable Artificial Intelligence (XAI).

Procedural Content Generation via Machine Learning (PCGML)

Procedural Content Generation via Machine Learning (PCGML) is a field of research focused on the creation of game content by machine learning models that have been

trained on existing game content (Summerville et al. 2018). Super Mario Bros. level design represents the most consistent area of research into PCGML. Researchers have applied many machine learning methods such as Markov chains (Snodgrass and Ontanón 2016), Monte-Carlo Tree Search (MCTS) (Summerville, Philip, and Mateas 2015), Long Short-Term Recurrent Neural Networks (LSTMs) (Summerville and Mateas 2016), Autoencoders (Jain et al. 2016), Generative Adversarial Neural Networks (GANs) (Volz et al. 2018), and genetic algorithms through learned evaluation functions (Dahlskog and Togelius 2014) to generate these levels. In a recent work, Khalifa et al proposed a framework to generate game levels using Reinforcement Learning (RL), though they did not evaluate it in Super Mario Bros. (Khalifa et al. 2020). We also draw on reinforcement learning for our agent, however our approach differs from this prior work in terms of focusing on explainability.

Co-creative systems

There are numerous prior co-creative systems for game design. These approaches traditionally have not made use of ML, instead they rely on approaches like heuristics search, evolutionary algorithms, and grammars (Smith, Whitehead, and Mateas 2010; Liapis, Yannakakis, and Togelius 2013; Yannakakis, Liapis, and Alexopoulos 2014; Deterding et al. 2017; Baldwin et al. 2017; Charity, Khalifa, and Togelius 2020). ML methods have only recently been incorporated into co-creative game content generation. Guzdial et al. proposed a Deep RL agent for co-creative Procedural Level Generation via Machine Learning (PLGML) (Guzdial, Liao, and Riedl 2018). In another recent work, Schrum et al. presented a tool for applying interactive latent variable evolution to generative adversarial network models that produce video game levels (Schrum et al. 2020). The major difference between our approach and previous ones is that it explains an AI partner's actions based on what it learned during training.

It is important to note that we are not actually evaluating our approach in the context of co-creative interaction with a human subject study. We are only making use of data from prior studies in which humans interacted with ML and RL agents in co-creative systems.

Explainable Artificial Intelligence (XAI)

The majority of existing XAI approaches can be separated according to which of two general methods they rely on: (**A**) visualizing the learned features of a model (Erhan et al. 2009; Simonyan, Vedaldi, and Zisserman 2013; Nguyen, Yosinski, and Clune 2015; 2016; Nguyen et al. 2017; Olah, Mordvintsev, and Schubert 2017; Weidele, Strobelt, and Martino 2019) and (**B**) demonstrating the relationship between neurons (Zeiler and Fergus 2014; Fong and Vedaldi 2017; Selvaraju et al. 2017). Olah et al. developed a unified framework that included both (A) and (B) methods. (Olah et al. 2018).

There are a few prior works focused on XAI applied to game design and game playing. Guzdial et al. presented an approach to Explainable PCGML via Design Patterns in which the design patterns act as a vocabulary and mode of
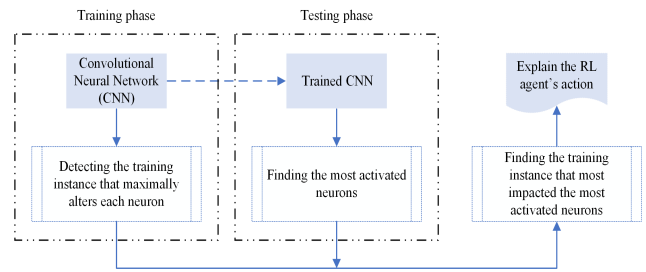

Figure 1: General steps of our approach

interaction between user and model (Guzdial et al. 2018). Ehsan et al. introduced AI rationalization, an approach for explaining agent behavior for automated game playing based on how a human would explain a similar behavior (Ehsan et al. 2018). Zhu et al. proposed a new research area of eXplainable AI for Designers (XAID) to help game designers better utilize AI and ML in their design tasks through co-creation (Zhu et al. 2018).

There exist a few approaches to explain RL agent's actions (Puiutta and Veith 2020). Madmul et al. presented an approach that learns structural causal models to derive causal explanations of the behavior of model-free RL agents (Madumal et al. 2019). Kumar et al. presented a deep reinforcement learning approach to control an energy storage system. They visualized the learned policies of the RL agent through the course of training and visualized the strategies followed by the agent to users (Kumar 2019). Cruz et al. proposed a memory-based explainable reinforcement learning (MXRL) where an agent explained the reasons why some decisions were taken in certain situations using an episodic memory (Cruz, Dazeley, and Vamplew 2019). In another recent paper, an approach was presented that employs explanations as feedback from humans in a human-in-the-loop reinforcement learning system (Guan, Verma, and Kambhampati 2020).

To the best of our knowledge, this is the first XAI work focused on the training data of a target ML model. Our approach differs from existing XAI work in detailed inspection and alteration of the training phase.

System Overview

In this paper, we present an approach for Explainable AI (XAI) that aims to answer the question "What did the AI agent learn during training that led it to make that specific action?". As is shown in Figure 1, the general steps of the approach are as follows: First, during training a DNN, we detect the training instance (or instances) that maximally alters each neuron inside the network. Secondly, during testing, we pass each instance through the network and find the neuron that is most activated (Erhan, Courville, and Bengio 2010). Then given the information from the first step, we can easily identify an instance (or instances) from the training data that maximally impacted the most activated neuron. We refer to this as "*the most responsible training instance*" for the AI agent's action. The intuition is that the user can take this explanation as something akin to the end goal of

the agent taking that action. Our hope is that it will be helpful in the user deciding whether to keep or remove some addition by the AI. For example in Figure 3, given the most responsible level as the explanation, the user might keep the lower of the two Goombas, despite the fact that it seems to be floating, if they can match it to the Goombas from the most responsible level.

For this purpose, we pre-trained a Deep RL agent using data from interactions of human users with three different ML level design partners (LSTM, Markov Chain, and Bayes Net) to generate the Super Mario Bros level. This is the same Deep RL architecture and data from prior work by Guzdial et al. (Guzdial, Liao, and Riedl 2018) for co-creative Procedural Level Generation via Machine Learning (PLGML), in which they made use of the level design editor from (Guzdial et al. 2017) which is publicly online.[1] The agent is designed to take in a current level design state and to output additions to that level design, in order to iteratively complete a level with a human partner.

Our training inputs are states and the outputs are the Q table values for taking a particular action for the particular state. The input comes into the network as a state of shape (40x15x34). The 40 is the width and 15 is the height of a level chunk. At each x,y location there are 34 possible level components (e.g. ground, goomba, pipe, mushroom, tree, Mario, flag, ...) that could be placed there. As is shown in the visualized architecture of the Convolutional Neural Network (CNN) in Figure 2, it has three convolutional layers and a fully connected layer followed by a reshaping function to make the output in the form of the action matrix which is (40x15x32). The player (Mario) and flag are the level entities that cannot be counted as an action, so there are 32 possible action components instead of the 34 state entities. Our activation function is "Leaky ReLu" for every layer and the loss function is "Mean Squared Error" and the optimizer is "Adam", with the network built in Tensorflow (Abadi et al. 2016). We make use of this existing agent and data since it is the only example of a co-creative PCGML agent where the data from a human subject study is publicly available.

During each training epoch we employ a batch size of one to track when each training instance passes through the network. We calculate and store the change of neuron weights between batches. After training, by summing over the changes of each neuron weight with respect to training data, we are able to identify which training instance maximally results in alteration of a neuron. Since positive and negative values can counteract each other's effects, it is important to not look at the absolute values until the end of the training. We can then sum and store this information inside eight arrays of shape (4x4x34) for the first convolutional layer, 16 arrays of shape (3x3x8) for the second convolutional layer, and 32 arrays of shape (3x3x16) for the third convolutional layer. These are the shapes of the filters in each layer. We name these arrays Most Responsible Instance for each Neuron in each Convolutional layer (MRIN-Conv1, MRIN-Conv2, and MRIN-Conv3). These data representations link neurons to IDs representing a particular instance

---

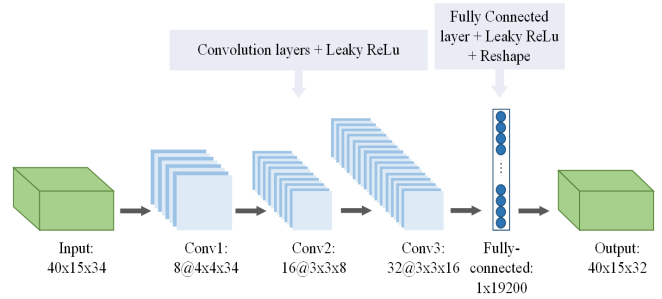[1] 1https://github.com/mguzdial3/Morai-Maker-Engine

Figure 2: Architecture of our Convolutional Neural Network (CNN).

of a human user working with the AI in the co-creative tool. We can then search these arrays and find the ID of a training instance that is the most responsible for changes to a particular weight.

Our end goal is to determine the most responsible training instance for a particular prediction made by our trained CNN. To do that, we need to find out what part of the network was most important in making that prediction. We can then determine the most responsible instance for the final weights of this most important part of the network. The most activated filter of each convolutional layer is a filter that contributes to the slice with the largest magnitude in the output of that layer. Hence the most activated filter can be considered the most important part of the convolutional layer for that specific test instance (Erhan, Courville, and Bengio 2010). For example, we pass a test instance into the network. A test instance is a (40x15x34) state that is a chunk of a partially designed level. Since the first convolutional layer has 8 4x4x34 filters with the same padding, the output would be in the shape of (40x15x8). Then we find the (40x15) slice with the largest values. The most activated filter is a (4x4x34) array in our convolutional layer which led to the slice with the greatest magnitude.

Finally, once we have the maximally activated filter we can identify *the most responsible* training instance (or instances) by querying the MRIN-Conv arrays we built during training. The most responsible training instance is the ID that most repeated in the MRIN-Conv array associated with the maximally activated filter. We chose the most repeated ID since it is the one that most frequently impacted the majority of the neurons in the filter during training.

Evaluation

In this section, we present two evaluations of our system. We call the first evaluation our "Explainability Evaluation" as it addresses the ability of our system to provide explanations that help a user predict an AI agent's actions. We call the second evaluation our "User Labeling Error Evaluation" as it addresses the ability of our system to help human users identify positive and negative AI additions during the co-creative process. Both evaluations approximate the impact of our approach on human partners by using existing data of AI-human interactions. Essentially, we act as though the pre-
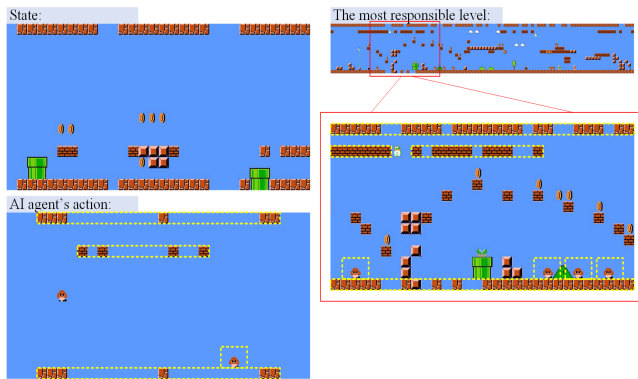
Figure 3: An example of explaining an AI agent's action by representing the most responsible level.

recorded actions of the AI agent were outputs from our Deep RL agent and identify the responsible training instances as if this were the case. Due to the fact that our system derives examples as explanations for the behavior of a co-creative Deep RL agent, a human subject study would be the natural way to evaluate our system. However, prior to a human subject study, we first wanted to gather some evidence of the value of this approach.

Explainability Evaluation

The first claim we made was that this approach can help human users better understand and predict the actions of an AI agent. In this experiment we use the most responsible level as an approximation of the AI agent's goal, in other words what final level the AI agent is working towards. The most responsible level refers to a level at the end of a human user's interactions with an AI agent. We identify this level by finding the most responsible training instance as above and identifying the level at the end of that training sequence. This experiment is meant to determine if this can help a user to predict the AI agent's actions. To do this, we passed test instances into our network and found the most responsible training instances. We then compared the most responsible level for some current test instance to the AI agent's action in the next test instance. If the most responsible level is similar to the action it would indicate that the most responsible level can be a potential explanation for the AI agent's action by priming the user to better predict future actions by the AI agent. In comparison, we randomly selected 20 levels from the training data and found their similarities to the AI agent's action in the next test instance. If our approach outperforms the random levels, it will support the claim that the responsible level is better suited to helping predict future AI agent actions compared to random levels.

We used two different sets of test data:

(A) Our first testset is derived from a study in which users interacted with pairs of three different ML agents as mentioned in our System Overview section (Guzdial, Liao, and Riedl 2018). We used the same testset identified in that paper.

(B) Our second testset is obtained from a study in which expert level designer users interacted with the trained Deep RL agent (Guzdial et al. 2019).

If we find success with the first testset then that would indicate that our trained Deep RL agent is a good surrogate for the original three ML agents, since we would be in effect predicting the next action of one of these agents. Good results for the second testset would demonstrate the capability for prediction of the Deep RL agent's actions itself. Since the first convolutional layer is the layer that most directly reasons over the level structure, we decided to find the most responsible training instance of just the first convolutional layer. However, this setup puts our approach at a disadvantage, since we are going to compare only one most responsible level to 20 random ones.

For comparing the most responsible level and the random levels to the actions, we needed to define a suitable metric. We desired a metric that detects local overlaps and represents the similarity between a level and action. We wanted to pick square windows which are not the same size as the first convolutional layer, to capture some local structures without biasing the metric too far towards our first convolutional layer. As a result, we found all three-by-three non-empty patches for both a given level and an action. Then we counted the number of exact matches of these patches on both sides, removing the matched ones from the dataset since we wanted to count the same patches only once. Finally, we divided the total number of the matched patches by the total number of patches in the action, since this was always smaller than the number from the level. We refer to this metric as the *local overlap ratio*.

Explainability Evaluation Results

We had 242 samples in the first testset and 69 samples in the second one. Since we wanted to compare instances in which the AI agent actually made some serious changes, we chose instances where the AI agent added more than 10 components in its next action. Thus we came to 38 and 46 instances from the first and second testsets, respectively.

Our approach outperforms the random baseline in 78.94 percent of 38 instances for the ML agents data and 67.29 percent of 46 instances for the Deep RL agent data. The average of the local overlap ratios is shown in Table 1 (higher is better). The minimum value here would be 0 for zero overlap and the maximum value would be 1 for complete overlap between the action and the most responsible level or the random level. This normalization means that even small differences in this metric represent large perceptual differences. For example, a 0.04 difference in the local overlap ratio between the most responsible level and the random levels in Table 1 indicates the most responsible level has 20 more three-by-three non-empty overlaps. We expect that the reason that the Deep RL agent values are generally lower is that the second study made use of published level designers rather than novices and an adaptive Deep RL Agent, meaning that there was more varied behavior compared with the three ML agents.

An example of explainability is demonstrated in Figure 3. As is shown in the figure, the AI agent made an action and

TestSet	Most Responsible Level	Random Levels
ML Agents	**0.4653**	0.3841
Deep RL	**0.2880**	0.2472

Table 1: A table comparing the average of the most responsible levels to the average of the random levels for both testsets.

added some components (e.g. goomba and ground) to the existing state. By looking at the chunk of the most responsible level, the user might realize that the AI agent wants to generate a level including some goombas as enemies and some blocks in the middle of the screen. The AI agent also added ground at the bottom and top of the screen, which the user could identify as being consistent with both their input to the agent and the most responsible level.

User Labeling Error Evaluation

For the second evaluation, we wanted to get some sense of whether this approach could be successful in terms of assisting a human user in better understanding good and bad agent actions during the co-creation process. To do this, we needed to identify specific instances where our tool could be helpful in the data we have available. We defined two such concepts: (A) false-positive decisions and (B) false-negative decisions, based on the interactions between users and AI partner during level generation:

(A) **False-positive decisions** are additions by the AI partner that the user kept at first but then deleted later.

(B) **False-negative decisions** are additions by the AI partner that the user deleted at first but then added later.

Given these concepts, if we could help the user avoid making these kinds of decisions, our approach could help a human user during level generation. We anticipated that one reason that users made these kinds of decisions was from a lack of context of the AI agent's action. Thus, if the user had context they may not delete or keep what they would otherwise keep or delete, respectively.

To accomplish this, we implemented an algorithmic way to determine false-positives and false-negatives among the two testsets described in the previous evaluation. In this algorithm, we first find all user decisions in terms of deleting or keeping an addition by the AI agent. Then we look at the level at the end of the user and the AI agent's interaction. If a deleted AI addition exists in the final level, it is counted as a false-negative example, and if a kept addition does not exist in the final level it is counted as a false-positive example.

Once we discovered all false-negative and false-positive examples, we found the state before the example was added by the AI agent and named it the Introduction-state (I-state). We found the state in which false-positivity or false-negativity occurred (i.e. when a user re-added a false-negative or deleted a false-positive) and named it the Contradiction-state (C-state). Since some change between the I-state and the C-state led to the user altering their decision, we wanted to see some sign that presenting the most responsible level to the user could change their mind before

they reached this point. Thus we compared these two states to find all the changes that the AI agent or the user made and named this the Difference-state (D-state).

We compared each D-state with the final generated level derived from the most responsible training instance. We also compared each D-state with 20 other randomly selected levels from the existing data. For the comparison, we used the local overlap ratio defined in the previous evaluation. If our approach outperforms the random baseline, we will be able to say that there is some support for the responsible level helping the user avoid false-positives and false-negatives in comparison to random levels.

User Labeling Error Evaluation Results

We found five false-negative and 24 false-positive examples in the first testset and five false-negative and 54 false-positive examples in the second one. The results of the evaluation are demonstrated in Figures 4.

For the first dataset which included the actions of the three ML agents, our approach outperformed the random baseline in 65.51 percent of the examples. The average of the local overlap ratio values for our approach was 0.1717 which is more than the 0.1647 for the random levels. For the second dataset obtained from the Deep RL agent, our approach outperformed the baseline in 59.32 percent of the examples. The average of the local overlap ratio values were 0.2665 and 0.2328 for the most responsible level and random levels, respectively. Again this represents a large perceptual difference of roughly 15 more non-empty 3x3 overlaps.

Interestingly, our approach outperforms the random levels in all of the false-negative examples in the second dataset, compared with just 20 percent of false-negatives in the first dataset. Further, our approach performs around 1.5 times better than the random levels in 15 false-positive examples in the second dataset. These instances come from the study that used the same RL agent as we used to derive our explanations, which could account for this performance.

Discussion

In this paper, we present an XAI approach for a pre-trained Deep RL agent. Our hypothesis was that our method could be helpful to human users. We evaluated it by approximating this process for two tasks using two existing datasets. These datasets are obtained from studies using three ML partners and an RL agent. Essentially, we used the XAI-enabled agent in this paper as if it were the agents used in these datasets. The results of our first evaluation demonstrates that our method is able to represent examples as explanations to help users predict an agent's next action. The results of our second evaluation support our hypothesis and give us an initial signal that this approach could be successful in order to help human users more efficiently cooperate with a Deep RL agent. This indicates the ability of our approach to help human designers by presenting an explanation for an AI agent's actions during a co-creation process.

A human subject study would be a more reasonable way to evaluate this system since human users might be able to derive meaning from the responsible level that our similarity metric could not capture. Our approach performs better
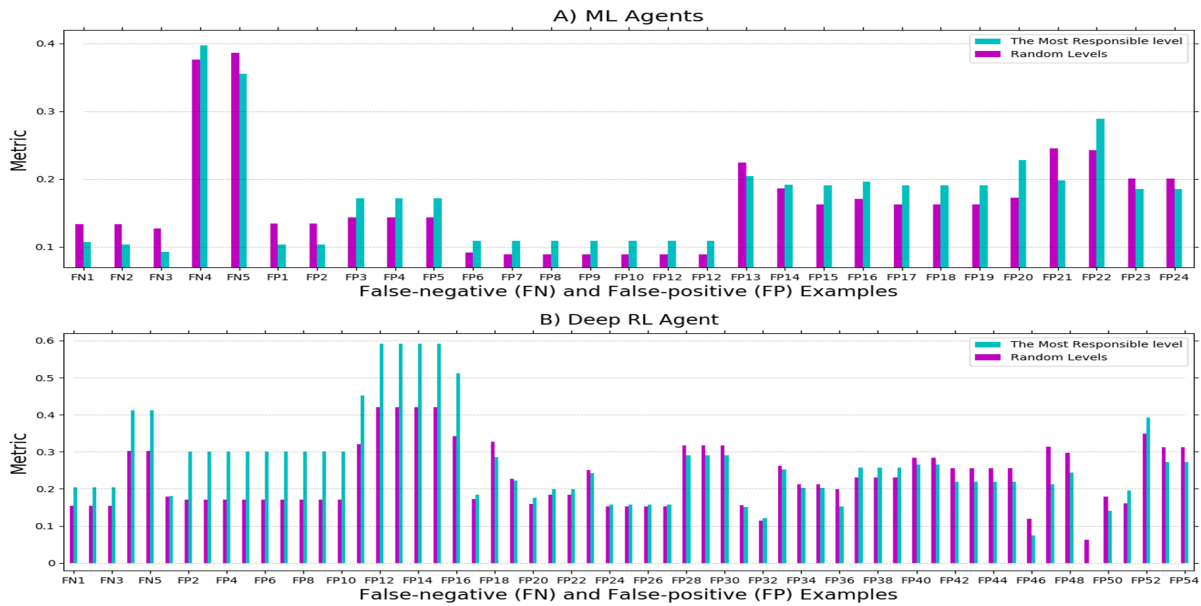
Figure 4: Results of the User Labeling Error Evaluation.

than our baseline of random levels in both evaluation methods and this presents evidence towards its value at this task. However, we look forward to investigating a human subject study in order to fully validate these results.

There could be other alternatives to a human subject study. For example, a secondary AI agent that predicts our primary AI agent's actions can play a human partner's role in the co-creative system. Thus making use of a secondary AI agent to evaluate our system before running a human subject study might be a simple next step.

It is important to mention that we only offer one most responsible level from only the first convolutional layer as an explanation. Looking into providing a user with multiple responsible levels or looking into the most responsible levels of the other layers could be a potential way to further improve our approach. Our metric for determining the most responsible training instance is based on finding the most repeated instance inside the MRIN-Conv arrays associated with the most activated filter. We identified the most activated filter by looking at the absolute values. We plan to investigate other metrics such as looking for the most activated neurons outside of the filters. In addition, considering negative and positive values separately in the maximal activation process could also lead to improved behavior. Negative values might indicate that an instance negatively impacted a neuron. It could be the case then that the filter might be maximally activated because it was giving a very strong signal against some action.

One quirk of our current approach is that the most responsible training instance depends on the order in which it was presented to the model during the training. Thus, this measure does not tell us about any inherent quality of a particular training data instance, only it's relevance to a particular model that has undergone a particular training regimen.

In the future, we intend to explore how more general representations of responsibility such as Shapely values might intersect with this approach (Ghorbani and Zou 2019).

Only the domain of a co-creative system for designing Super Mario Bros. levels is explored in this paper. Thus making use of other games will be required to ensure this is a general method for level design co-creativity. Beyond that, we anticipate a need to demonstrate our approach on different domains outside of games. We look forward to running another study to apply our approach to human-in-the-loop reinforcement learning or other co-creative domains.

Conclusions

In this paper we present an approach to XAI that provides human users with the most responsible training instance as an explanation for an AI agent's action. In support of this approach, we present results from two evaluations. The first evaluation demonstrates the ability of our approach to offer explanations and to help a human partner predict an AI agent's actions. The second evaluation demonstrates the ability of our approach to help human users better identify good and bad instances of an AI agent's behavior. To the best of our knowledge this represents the first XAI approach focused on training instances.

Acknowledgements

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al.

2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 265–283.

Adadi, A., and Berrada, M. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access* 6:52138–52160.

Bach, S.; Binder, A.; Müller, K.-R.; and Samek, W. 2016. Controlling explanatory heatmap resolution and semantics via decomposition depth. In *2016 IEEE International Conference on Image Processing (ICIP)*, 2271–2275. IEEE.

Baldwin, A.; Dahlskog, S.; Font, J. M.; and Holmberg, J. 2017. Mixed-initiative procedural generation of dungeons using game design patterns. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, 25–32. IEEE.

Biran, O., and Cotton, C. 2017. Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, volume 8.

Boz, O., and Hillman, D. 2000. *Converting a trained neural network to a decision tree dectext-decision tree extractor*. Citeseer.

Charity, M.; Khalifa, A.; and Togelius, J. 2020. Baba is y'all: Collaborative mixed-initiative level design. *arXiv preprint arXiv:2003.14294*.

Che, Z.; Purushotham, S.; Khemani, R.; and Liu, Y. 2015. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*.

Cook, M.; Colton, S.; Pease, A.; and Llano, M. T. 2019. Framing in computational creativity-a survey and taxonomy. In *ICCC*, 156–163.

Cortez, P., and Embrechts, M. J. 2011. Opening black box data mining models using sensitivity analysis. In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 341–348. IEEE.

Cortez, P., and Embrechts, M. J. 2013. Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences* 225:1–17.

Cruz, F.; Dazeley, R.; and Vamplew, P. 2019. Memory-based explainable reinforcement learning. In *Australasian Joint Conference on Artificial Intelligence*, 66–77. Springer.

Dabkowski, P., and Gal, Y. 2017. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, 6967–6976.

Dahlskog, S., and Togelius, J. 2014. A multi-level level generator. In *2014 IEEE Conference on Computational Intelligence and Games*, 1–8. IEEE.

Deterding, S.; Hook, J.; Fiebrink, R.; Gillies, M.; Gow, J.; Akten, M.; Smith, G.; Liapis, A.; and Compton, K. 2017. Mixed-initiative creative interfaces. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 628–635.

Ehsan, U.; Harrison, B.; Chan, L.; and Riedl, M. O. 2018. Rationalization: A neural machine translation approach to generating natural language explanations. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 81–87.

Erhan, D.; Bengio, Y.; Courville, A.; and Vincent, P. 2009. Visualizing higher-layer features of a deep network. *University of Montreal* 1341(3):1.

Erhan, D.; Courville, A.; and Bengio, Y. 2010. Understanding representations learned in deep architectures. *Department dInformatique et Recherche Operationnelle, University of Montreal, QC, Canada, Tech. Rep* 1355:1.

Fong, R. C., and Vedaldi, A. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, 3429–3437.

García, S.; Fernández, A.; and Herrera, F. 2009. Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. *Applied Soft Computing* 9(4):1304–1314.

Ghorbani, A., and Zou, J. 2019. Data shapley: Equitable valuation of data for machine learning. *arXiv preprint arXiv:1904.02868*.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.

Guan, L.; Verma, M.; and Kambhampati, S. 2020. Explanation augmented feedback in human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2006.14804*.

Guzdial, M. J.; Chen, J.; Chen, S.-Y.; and Riedl, M. 2017. A general level design editor for co-creative level design. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Guzdial, M.; Reno, J.; Chen, J.; Smith, G.; and Riedl, M. 2018. Explainable pcgml via game design patterns. *arXiv preprint arXiv:1809.09419*.

Guzdial, M.; Liao, N.; Chen, J.; Chen, S.-Y.; Shah, S.; Shah, V.; Reno, J.; Smith, G.; and Riedl, M. O. 2019. Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13.

Guzdial, M.; Liao, N.; and Riedl, M. 2018. Co-creative level design via machine learning. *arXiv preprint arXiv:1809.09420*.

Hara, S., and Hayashi, K. 2018. Making tree ensembles interpretable: A bayesian model selection approach. In *International Conference on Artificial Intelligence and Statistics*, 77–85.

Jain, R.; Isaksen, A.; Holmgård, C.; and Togelius, J. 2016. Autoencoders for level generation, repair, and recognition. In *Proceedings of the ICCC Workshop on Computational Creativity and Games*.

Khalifa, A.; Bontrager, P.; Earle, S.; and Togelius, J. 2020. Pcgrl: Procedural content generation via reinforcement learning. *arXiv preprint arXiv:2001.09212*.

Kumar, H. 2019. Explainable ai: Deep reinforcement learning agents for residential demand side cost savings in smart grids. *arXiv preprint arXiv:1910.08719*.

Letham, B.; Rudin, C.; McCormick, T. H.; Madigan, D.; et al. 2015. Interpretable classifiers using rules and bayesian

analysis: Building a better stroke prediction model. *The Annals of Applied Statistics* 9(3):1350–1371.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient sketchbook: computer-assisted game level authoring.

Madumal, P.; Miller, T.; Sonenberg, L.; and Vetere, F. 2019. Explainable reinforcement learning through a causal lens. *arXiv preprint arXiv:1905.10958*.

Nguyen, A.; Clune, J.; Bengio, Y.; Dosovitskiy, A.; and Yosinski, J. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4467–4477.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 427–436.

Nguyen, A.; Yosinski, J.; and Clune, J. 2016. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*.

Olah, C.; Satyanarayan, A.; Johnson, I.; Carter, S.; Schubert, L.; Ye, K.; and Mordvintsev, A. 2018. The building blocks of interpretability. *Distill* 3(3):e10.

Olah, C.; Mordvintsev, A.; and Schubert, L. 2017. Feature visualization. *Distill* 2(11):e7.

Puiutta, E., and Veith, E. 2020. Explainable reinforcement learning: A survey. *arXiv preprint arXiv:2005.06247*.

Schrum, J.; Gutierrez, J.; Volz, V.; Liu, J.; Lucas, S.; and Risi, S. 2020. Interactive evolution and exploration within latent level-design space of generative adversarial networks. *arXiv preprint arXiv:2004.00151*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Smith, G.; Whitehead, J.; and Mateas, M. 2010. Tanagra: A mixed-initiative level design tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, 209–216.

Snodgrass, S., and Ontanón, S. 2016. Learning to generate video game maps using markov models. *IEEE transactions on computational intelligence and AI in games* 9(4):410–422.

Summerville, A., and Mateas, M. 2016. Super mario as a string: Platformer level generation via lstms. *arXiv preprint arXiv:1603.00930*.

Summerville, A.; Snodgrass, S.; Guzdial, M.; Holmgård, C.; Hoover, A. K.; Isaksen, A.; Nealen, A.; and Togelius, J. 2018. Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games* 10(3):257–270.

Summerville, A. J.; Philip, S.; and Mateas, M. 2015. Mcmcts pcg 4 smb: Monte carlo tree search to guide platformer level generation. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*.

Tan, S.; Caruana, R.; Hooker, G.; and Lou, Y. 2017. Detecting bias in black-box models using transparent model distillation. *arXiv preprint arXiv:1710.06169*.

Volz, V.; Schrum, J.; Liu, J.; Lucas, S. M.; Smith, A.; and Risi, S. 2018. Evolving mario levels in the latent space of a deep convolutional generative adversarial network. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 221–228.

Weidele, D.; Strobelt, H.; and Martino, M. 2019. Deepling: Avisual interpretability system for convolutional neural networks. *Proceedings SysML*.

Xu, K.; Park, D. H.; Yi, C.; and Sutton, C. 2018. Interpreting deep classifier by visual distillation of dark knowledge. *arXiv preprint arXiv:1803.04042*.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity.

Zednik, C. 2019. Solving the black box problem: A normative framework for explainable artificial intelligence. *Philosophy & Technology* 1–24.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhu, J.; Liapis, A.; Risi, S.; Bidarra, R.; and Youngblood, G. M. 2018. Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–8. IEEE.