Abstracted Trajectory Visualization for Explainability in Reinforcement Learning

Yoshiki Takagi, Roderick Tabalba, Nurit Kirshenbaum, and Jason Leigh
Information and Computer Sciences
University of Hawaii at Manoa
Honolulu, HI, USA
takagiyo@hawaii.edu, tabalbar@hawaii.edu, nuritk@hawaii.edu, leighj@hawaii.edu

***Abstract*—Explainable AI (XAI) for lay users who do not have expertise in Reinforcement Learning (non-RL experts) plays an essential role in helping them build mental models of an agent's behavior in increasingly common applications, such as self-driving cars. An intriguing approach involves utilizing the "familiarization effect", where exposure to an agent's behavior across various scenarios assists users in naturally forming these mental models. However, this method might be less effective when dealing with multiple agents or complex, lengthy tasks, due to the limitations of human short-term visual memory where holding visual information briefly in mind. We propose that visualizing abstracted trajectories, which illustrate transitions between major states of the RL agents, can aid non-RL experts in understanding the agents' behaviors. Preliminary findings suggest that this visualization enables non-RL experts to efficiently recognize RL agents' behaviors.**

***Index Terms*—Explaniable AI, Trajectory Visualization, Trajectory Abstraction**

I. Introduction

In recent years, deep Reinforcement Learning (RL) agents, employing deep neural networks in their policies, has outperformed skilled humans in fields like video games, chess, and Go [21]. However, DRL agents, characterized by their opaque, 'black box' policies, pose challenges in enabling humans to construct mental models of their behaviors.

Explainable AI (XAI) for users who do not have expertise in Reinforcement Learning (non-RL experts) becomes crucial, aiding them in comprehending deep RL agents' behaviors in increasingly common applications, such as self-driving cars. So far, various facets of explainability for non-RL experts has been illuminated. These include explaining the policy of an agent [9], [24], justifying the action of an agent with reward [13], [26], and explaining the dynamics of an environment [5], [14]. Recently, counterfactual explanations that explain "*If A did not happen, B would not have happened*" and contrastive explanations that answer "*Why B_1 rather than B_2?*" have been actively considered as good explanations that are understandable for a wide variety of users [17], [22], [29].

One notable approach is leveraging the 'familiarization effect', where exposure to an agent's behavior in diverse scenarios helps users intuitively grasp mental models of the agent's behavior. For instance, Dragan & Srinivasa [33] observed that watching a robot's trajectory in videos enables users to predict the robot's future trajectory. Another study pointed out that short video clips of an agents' game-play can effectively build mental models of the agents' performance [1]. However, this strategy may falter with an increase in the number of agents, or in complex, extended tasks, due to the limitations of human short-term visual memory where holding visual information briefly in mind.

We posit that visualizing abstracted trajectories – a method of visually representing RL agents' behaviors through major state transitions – offers a promising solution to these challenges. This approach aims to give users a condensed visual summary of various agents' behaviors by transforming videos of these behaviors into abstracted trajectories. This paper focuses on the question: *how should trajectories be abstracted and visualized to facilitate users' intuitive understanding of an agent's behavior?* To address this, we introduce a trajectory abstraction algorithm and propose an interface for visualizing these abstracted trajectories, applicable to diverse agent types. Our initial findings from a pilot study indicate that while the proposed interface could benefit from enhancements, the visualization of abstracted trajectories proved helpful for non-RL experts in deducing agents' behaviors.

This paper is structured as follows: the next section covers the technical background needed to understand an algorithm used to generate the abstracted trajectories; Section III presents a literature review of relevant research; From section IV through VII, outline of proposed interface, preliminary evaluation, result, and discussions are provided; Finally, section VIII and IX give limitations and a conclusion.

II. Background

This section delves into the technical background of machine learning models, particularly focusing on extracting transitions between major states in the RL agents. For details on how these extracted transitions are visualized, refer to section IV. As depicted in Fig. 1, our algorithmic process comprises three steps: trajectory extraction, trajectory abstraction, and visualization of the abstracted trajectory. In the following two subsections, the trajectory extraction and trajectory abstraction will be explained.

This project is funded in part by the National Science Foundation awards: 2004014, 2003800, 2003387, 2149133, and the Academy for Creative Media System.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00023

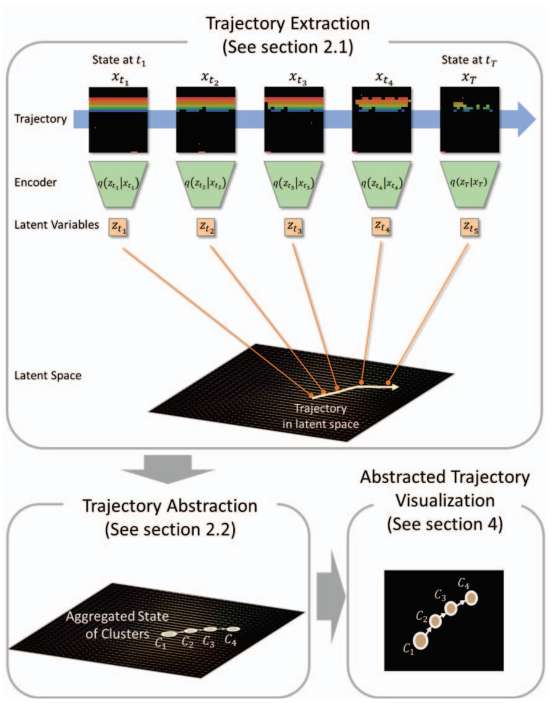


Fig. 1. The process of generating abstracted trajectories from input images that a RL agent observed. The features z (latent variables) are extracted from the input images x with VAE (See section II-A). The extracted features z are abstracted by spatio-temporal clustering (See section II-B).

A. Trajectory Extraction

In this study, we employed a machine learning model known as the Variational Autoencoder (VAE) to extract interpretable series of features without human supervision. Given an input x, the VAE's objective is to learn a vector z that captures the features of x in a disentangled manner (see Fig. 2 (a)). For example, when the VAE trained with images of Breakout, a game in which the player breaks rainbow-colored blocks by moving the paddle left and right and hitting the ball back to the block, the VAE learns to represent an input image of the game using z. Since the dimension of the vector z is usually high, dimensional reduction techniques, such as PCA [6], t-SNE [28] and UMAP [19], are used to project learned representation in 2D space. An example of this can be seen with the feature representing the number of remaining blocks in Breakout, visualized using PCA in a 2D space (Fig. 2 (b)).Ideally these independent factors are expected to be captured in separated dimensions of z (i.e., semantic dimensions), however, in practice, some dimensions of z will not learn to have semantic meanings. To address this, we introduce the β-VAE [10] which can facilitate z to learn more disentanglement representation than vanilla VAE. The β-VAE is trained to minimize the following loss function:

$$\mathcal{L}_\beta(\theta, \phi|x) = -\mathbb{E}_{q_\phi(z|x)}[log\left(p_\theta(x|z)\right)] + \beta D_{\mathbb{KL}}\left(q_\phi(z|x)||p_\theta(z)\right) \tag{1}$$

The first term of this equation represents the reconstruction loss, while the second term is a regularization term for enhancing disentanglement. With $\beta > 1$, β-VAE encourages a more disentangled z by imposing constraints on the latent bottleneck. As a result, similar images are positioned closely together in the latent space. With the continuous state changes due to the agent's actions, these transitions are depicted as trajectories in the latent space. In this research, the architecture of the β-VAE follows the approach presented by Ha and Schmidhuber [7].

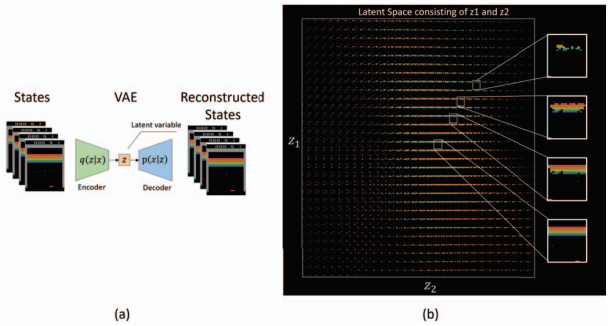


Fig. 2. a) The architecture of VAE, input images are fed to the decoder of VAE and the decoder is trained to compress inputs to latent variable z. The encoder of VAE is also trained at the same time to reconstruct the inputs from corresponding latent vectors. b) The reconstructed images projected on the latent space formed z_1 and z_2.

B. Trajectory Abstraction

To abstract the trajectories generated by the β-VAE, we utilized ST-DBSCAN [4], a clustering algorithm tailored for spatio-temporal data. ST-DBSCAN is a density-based clustering method, distinct from many clustering techniques that often rely on the independence and identically distributed (i.i.d.) assumption. This algorithm takes into account the temporal dynamics of data, forming clusters based on the density of points within a specified spatio-temporal radius. This approach is particularly effective in capturing the temporal structure inherent in the state transitions of our data.

ST-DBSCAN categorizes the features z, derived from the encoder of the β-VAE, into several clusters. To identify major states along a trajectory, we compute the median of the z distribution for each cluster and then input these medians into the decoder of the β-VAE. The abstracted trajectories are subsequently formed by connecting these major states in their chronological sequence.

III. Related Work

This study intersects with two significant areas of research: Visual Analytics (VA) for enhancing the explainability of RL agents, and the visualization of trajectories for temporal data. This section outlines contributions in these domains and differentiates our work from existing efforts.

VA has been recognized for its potential in facilitating hypothesis formation to explain visualized behavior patterns of RL agents, primarily through interactive interfaces. These interfaces have been primarily designed for RL practitioners to discover behavior patterns and potential bugs of an agent [12], [18], [20], [23], [30]. These methods enable the RL practitioners to construct accurate mental models about an agent by examining information in multiple charts about the agent. However, their complexity makes them less accessible to non-RL experts. Moreover, these methods are generally tailored for specific machine learning models (like RNN, LSTM, and DQN), limiting their applicability in comparing different types of models.

Trajectory visualization, on the other hand, focuses on representing agents' behavior patterns as trajectories in a singular visual format. For example, the Projection Path Explorer visualizes multiple Rubik's Cube solution strategies as trajectories, analyzing these as patterns [11]. Such visualizations can reveal agent behavior patterns through clusters or bundles of trajectories. Zahavy *et al.* proposed an interface that employs t-SNE to project RL model's latent vectors into two dimensions, offering case studies to illustrate how these trajectories can depict agent behaviors [32]. However, these studies often lack user-based evaluations on the practicality and comprehension of the interfaces. Additionally, their complexity, stemming from directly projecting high-dimensional states to 2D spaces, poses challenges for non-RL experts.

In contrast, our paper introduces a novel trajectory abstraction algorithm designed to simplify the trajectories of diverse RL agents. This abstraction allows non-experts to easily grasp a high-level overview of RL agents' behavior patterns. We also conduct a pilot study to assess users' understanding and interpretation when exposed to the abstracted trajectory visualizations.

IV. TRAJECTORY VISUALIZATION INTERFACE

This section outlines our interactive interface designed for visualizing abstracted trajectories. The interface, as illustrated in Fig. 3, is comprised of two main components: a map view, which presents the trajectories as a directed graph, and a slider view, which displays the trajectories horizontally. The subsequent subsections detail the functionalities of both the map view and the slider view.

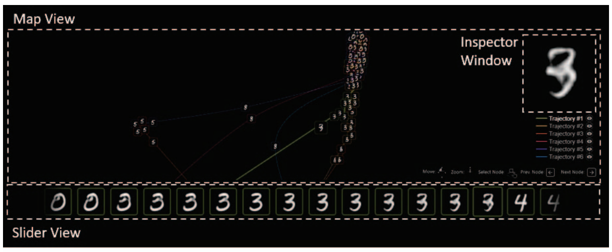


Fig. 3. Proposed interface for abstracted trajectory visualization: The map view is centrally placed, with the slider view positioned at the bottom. An enlarged image of a node, when hovered over, appears in the inspector window at the top right.

A. *Map View*

Located at the center of the interface, the map view visualizes abstracted trajectories as a directed graph. The nodes in this graph represent major states identified from clusters using ST-DBSCAN, as discussed in section II. These nodes are positioned based on a force simulation implemented with D3JS [3], where the Euclidean distances between latent vectors z of major states influence the linking strength, thereby forming clusters. The directed graph's edges, denoting temporal dependencies between nodes, are rendered using Bézier interpolation for enhanced readability, with time progression shown through animated dots along the edges.

Interaction with the map is facilitated through scrolling, dragging, hovering, and clicking. Users can pan and zoom on the map akin to navigating Google Maps. Hovering over nodes brings up an enlarged image in the top-right inspector window, allowing users to examine state transitions. Clicking on a node highlights its corresponding trajectory, and hovering over another node facilitates comparative analysis between trajectories. The highlighted trajectory is also synchronized with the slider view, which we will elaborate on next.

B. *Slider View*

The slider view arranges the nodes of abstracted trajectories horizontally in chronological order. Since the nodes in the slider view are sorted from left to right, users can easily see state transitions on the abstracted trajectories by comparing the adjacent images.

Users can interact with this view through scrolling, hovering, and clicking, similar to the map view. Scrolling horizontally navigates through the slider, while hovering over a node displays its enlarged image in the inspector window. Clicking on a node highlights its trajectory on the map, providing a dual perspective of both the trajectory's shape and its sequential states, as viewed in the map and slider views simultaneously.

V. PRELIMINARY EVALUATION

We conducted a preliminary comparative evaluation to assess the effectiveness of two visualization types: with and without trajectory abstraction. For the non-abstracted case, participants were shown animations like those in Fig. 4 (a), which we refer to as "complete trajectories" since they represent the full range of state transitions. In the abstracted case, as detailed in the previous section and shown in Fig. 4 (b), participants were provided with abstracted trajectories. In both scenarios, we measured task completion accuracy, the specifics of which are discussed in section V-B. This pilot study was designed to be conducted online to facilitate broad participation in future research.

A. *Questions*

To examine if abstracted trajectories aid non-RL experts in building mental models of agent behavior, we set the following guiding questions for this pilot study:

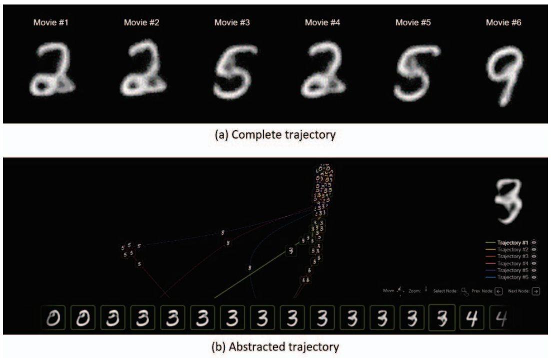


Fig. 4. Two visualization types used in a comparative evaluation

Q1- How well does a user's mental model obtained from the abstracted trajectory agree with a mental model obtained from complete trajectory?
Q2- Is the trajectory abstraction algorithm able to extract information that helps users to understand the behavioral pattern of RL agent?
Q3- What insights do users gain from the abstracted trajectories?

B. Task

We designed an analytical task, depicted in Fig. 5, to assess each visualization type's effectiveness. Participants were asked to observe a changing animation over time and identify the corresponding trajectory from six options, represented as either complete or abstracted trajectories. This task aimed to gauge participants' ability to generalize an agent's policy from each visualization type.

For this study, we designed tasks for 6 different applications, detailed in Table I. The Mnist application was used for a tutorial presented at the start of the pilot study. This tutorial showcased animations of handwritten digits transitioning smoothly (e.g., 0→4→3→2). In the complete trajectory scenario, participants observed six different animations (see Fig. 4 (a)), whereas in the abstracted trajectory scenario, they used the interface shown in Fig. 4 (b).

After the tutorial, participants were tested on the remaining five applications. A total of 10 questions (2 visualization types $\times$ 5 applications) were randomly presented. The detailed procedure following the tutorial is explained in section V-C.

The datasets of state transitions for each application were sourced from the work by Such *et al*. [25]. They trained six RL models including A2C, ApeX, DQN, ES, GA, and Rainbow, on various Atari games to support research that investigates the properties of these agents. In our study, replays of an Atari game played by these six RL models were represented as six trajectories. A participant never encountered a task with the same answer across visualization types.

In summary, we employed a within-subject design comparing two visualization types (with/without trajectory abstraction). The question blocks were counterbalanced to avoid bias, and participants never faced repeat answers across visualization types. Including the tutorial, the total number of trials amounted to 12N trials (2 visualization types $\times$ 6 applications $\times$ N participants).

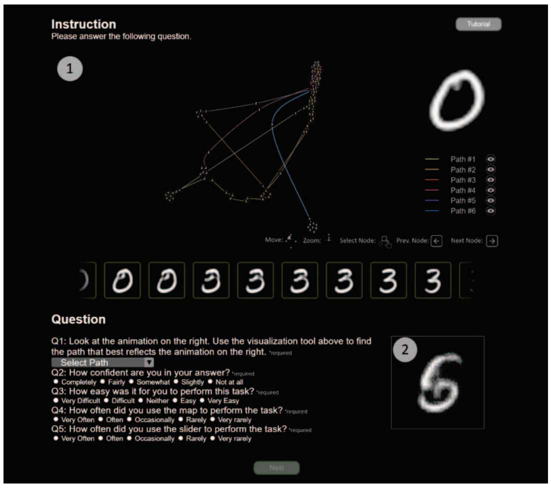
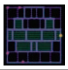


Fig. 5. Analytical task for abstracted trajectory: Participants are asked to match a trajectory visualized in the interface of abstracted trajectory (region (1) in the figure) with an animation (region (2)). In the complete trajectory scenario, region (1) displays complete trajectory visualizations as in figure 4 (a).

TABLE I
APPLICATIONS USED IN THIS PILOT STUDY

Mnist	Breakout	Qbert	Amidar	SpaceINvader	Boxing
					

C. Procedure

Each participant in the pilot study answered a total of 12 questions (2 visualization types $\times$ 6 applications including the tutorial). Participants were given unlimited time to complete each question and could not redo them. We assessed performance based on response accuracy. The accuracy of each visualization type was measured by the number of correct responses out of the total responses for each application. Comparing accuracy between the two types provided insight into how effectively participants could generalize an agent's policy from abstracted trajectories versus complete trajectories. For each question, participants rated their confidence in their answers and the question's difficulty using a 5-point Likert scale. For questions involving abstracted trajectories, we also asked participants to rate their usage frequency of the map and slider views using a 5-point Likert scale.

Upon completing all questions, participants rated the ease of use and usefulness of the abstracted trajectory interface on a 5-point Likert scale. Additionally, open-ended questions were provided to gather insights on their interpretation and challenges faced while using the interface of abstracted trajectories.

D. Experimental Setup

The pilot study was entirely conducted online. Participants used personal laptops or desktops for the task. Window size was recorded to exclude any who completed the task in an inadequately small window, though no exclusions were necessary. The study was approved by our internal IRB and lasted approximately 45 minutes on average.

E. Participants

We recruited 9 participants (5 male and 4 female) who self-declared as non-RL experts, aged between 22 and 45 years, from computer science. The participants rated the familiarity of the applications used in the study using a 4-point Likert scale. As shown in figure 6, most participants answered they have played the game at least once in Breakout and Space Invaders. For the other games, participants generally reported low familiarity.

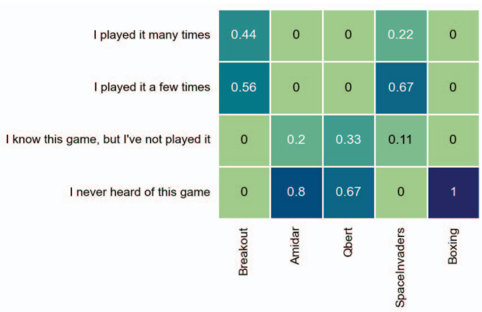


Fig. 6. Participants' familiarity with applications used in the pilot study, rated on a 4-point Likert scale.

VI. Results

This section presents the results of our study, encompassing both quantitative (accuracy) and qualitative (subjective ratings and responses to open-ended questions) aspects. We applied Fisher's exact test ($\alpha = .05$) for accuracy comparisons and the Mann-Whitney U test ($\alpha = .05$) for other comparisons.

A. Quantitative Results

The accuracy comparison between abstracted and complete trajectories is illustrated in figure 7. Due to some cell frequencies being less than 5, Fisher's exact test was employed. The test revealed no statistically significant differences in accuracy between the two groups.

B. Qualitative Results

1) Confidence and Difficulty of the task: Fig. 8 and 9 display participants' confidence in their answers and difficulty ratings of the task for both abstracted and complete trajectories. The Mann-Whitney U test indicated no significant differences in confidence or difficulty ratings between the two groups.

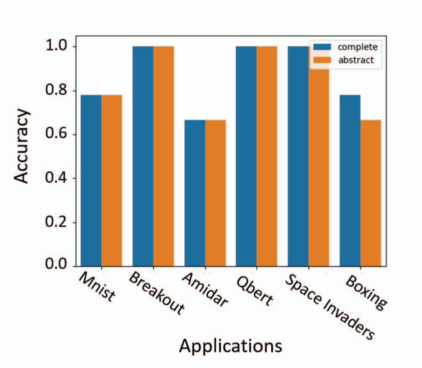


Fig. 7. Accuracy Results: No significant differences were observed between the two groups across all applications.

2) Usefulness of the interface: We gathered comments and feedback from participants after completing the study. Participants rated the question "Do you think the trajectory visualization tool is more useful than watching animations for performing tasks?" on a 5-point Likert scale and provided reasons for their ratings.

Most participants found the abstracted trajectory interface useful (see Fig. 10). However, opinions regarding the map view were mixed: about half found it generally useful, while the rest did not. One neutral participant commented, "(I am) not sure the map is for someone that has never dealt with data visualization." A participant who answered disagree suggested that "it would be nice to visualize nodes in the map with less cluttered, because tight clusters do not help clarity." Another participant who agreed on the rating scale left a positive comment about the map that "It helps compare and contrast what is similar and what is not."

Regarding the slider view, most participants found it helpful for task performance (see Fig. 10). When asked "What did you find most useful about the tool over watching the animation?", the majority of participants highlighted the slider's utility. For example, one participant noted, "The slider was really useful because it meant you could focus on certain parts of the animation." This aligns with the trend of participants using the slider more frequently than the map, as shown in Fig. 11.

3) Ease of use: We also collected comments about the interface's ease of use. Participants responded to "Do you think the trajectory visualization tool is easy to use?" on a 5-point Likert scale. Additionally, they provided feedback on a question "What was most difficult to understand from the visualization tool?".

Participants were divided on whether the interface was easy to use, with an equal number of participants saying it was more or less neutral. Those who found it easy to use commented, " It is not easy to use for finding a matching path but easy to use for eliminating similar paths." A participant who answered neutral stated that "(It) was hard to follow the map and at the same time compare it with the animations". A participant who strongly disagreed stated, "I did not find any purpose of using the map. Instead, I just toggled the path number on the right with the eye icons, then skimmed through the nodes to compare to the original animation.” Overall, participants mentioned the ease of use of the map as a criterion for evaluating the ease of use of the interface.

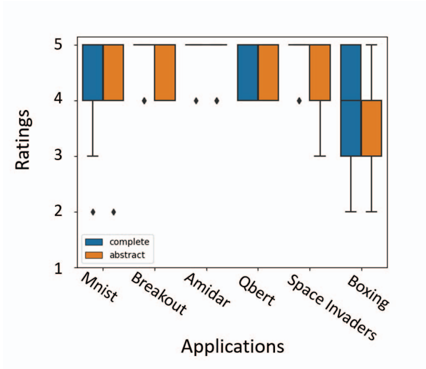


Fig. 8. Confidence Results: The vertical axis is the rating of 5-point Likert scale about participants’ confidence on their answer (1 - “Not at all”, 2 - “Slightly”, 3 - “Somewhat”, 4 - “Fairly” and 5 - “Completely”).

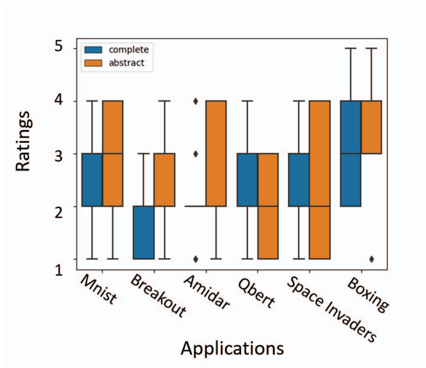


Fig. 9. Difficulty Results: The vertical axis is the difficulty rating of 5-point Likert scale about the task (1 - “Very easy”, 2 - “Easy”, 3 - “Neither”, 4 - “Difficult” and 5 - “Very difficult”).

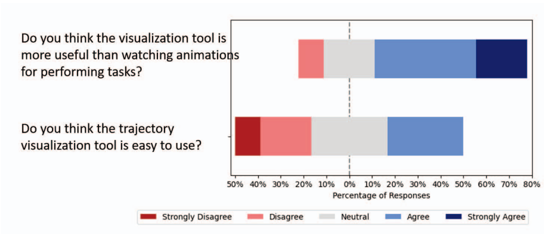


Fig. 10. Usefulness and Ease of Use Results.

VII. Discussions

The primary objective of our evaluation was to assess whether visualizing abstracted trajectories assists non-RL experts in forming mental models of agents’ behavior. The pilot study revealed comparable accuracy scores for trajectory identification tasks between abstracted and complete trajectories. Additionally, participants subjectively preferred the interface of abstracted trajectory. These preliminary findings support questions $Q1$ and $Q2$ from section V-A. However, some users raised concerns about the map view’s usefulness and ease of use, indicating a need for improvement in this area. In the following subsections, we discuss the pilot study results in relation to questions $Q1$, $Q2$, and $Q3$, and offer suggestions for enhancing the interface for abstracted trajectories.

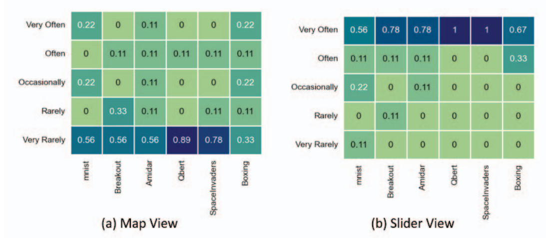


Fig. 11. Usage Results: The vertical axis shows a 5-point Likert scale rating of frequency of map and slider use. The slider was used more frequently than the map for task performance.

A. How well does a user’s mental model obtained from the abstracted trajectory agree with a mental model obtained from complete trajectory?

The task accuracies using abstracted trajectories were on par with those using complete trajectories, suggesting that the abstraction algorithm effectively distills crucial information about agents’ behavioral patterns. A participant supported the effectiveness of the trajectory abstraction by saying that ”You can see snapshots, you don’t need to wait for the animation to loop.” Another participant stated the benefit of abstracted trajectory on the short-term visual memory that ”Being able to go node by node, frame by frame, to compare the original animation to the different paths, is a lot easier than having to watch 6 other animations and comparing it to the original animation.” This comment emphasizes the advantage in visual memory that abstracted trajectory visualization allows for at glance comprehension of state transitions, whereas looking at complete trajectory necessitates refreshing their short-term visual memory.

B. Is the trajectory abstraction algorithm able to extract information that helps users to understand the behavioral pattern of RL agent?

While discussions on section VII-A suggest that the trajectory abstraction algorithm works to a degree, a participant’s comment about the need for a ”mental leap” to understand the state of each game suggests a gap between human intuition and the algorithm’s abstraction. The discrepancy between machine-learned spatial information and human conceptual understanding is a well-discussed topic in knowledge abstraction and representation learning [16], [27]. Designing interfaces that bridge these gaps remains a novel challenge in the CHI domain [31].

Applying models from video summarization in supervised and transfer learning to trajectory abstraction in RL might produce results more aligned with human intuition, as these methods have shown promise in creating condensed versions of longer videos [2], [8].

C. What insights do users gain from the abstracted trajectories?

The arrangement of node clusters and trajectory shapes on the map provided key insights for users in identifying trajectories similar to an animation. For instance, a user who analyzed trajectories commented, ”It helps compare and contrast what is similar and what is not. You can also refer back and forth to see which one looks more similar and keep it in mind or uncheck it so that you eliminate it as an option.” Another user referred to the benefit of the clusters of the nodes, ”clusters that were nicely isolated from others were easier to focus on finding the ”why are they far away”. Because they read such information, the map would be a more useful tool when performing tasks such as grouping similar behavior patterns of RL agents or counting the number of agents visiting similar states.

D. Improvements for the Interface

Feedback from the pilot study suggests the need for enhanced interaction and organization in the map view. Users found it challenging to trace trajectories with a mouse cursor. Implementing a direct manipulation technique designed for trajectory visualization [15] could allow for more intuitive navigation. Additionally, reducing node and trajectory overlap, perhaps through clustering techniques like k-means, would enhance visibility and encourage more effective use of the map.

Lastly, emphasizing temporal structures of the trajectory visualization in the map view may lead to user's better understanding for the visualization. Majority of participants in the pilot study preferred the slider rather than the map. This may be because the slider, of which nodes align from left to right based on temporal dependency, facilitates an intuitive grasp of state transitions. Thus, a visualization incorporating a branching tree structure could be advantageous for performing the task.

VIII. Limitations and Future Work

The findings of this paper should be considered preliminary due to the small sample size of the pilot study. Furthermore, the participant pool, being solely from a computer science background, highlights the need for diverse evaluations from individuals with varied backgrounds. However, the advantage of our evaluation framework's online nature presents an opportunity for employing crowdsourcing methods in future studies to address these limitations and broaden participant diversity.

IX. Conclusion

This paper introduced a novel XAI algorithm for generating abstracted trajectories from a range of RL agents' trajectories and proposed an interface for their visualization, aimed at helping non-RL experts understand RL agents' behavior patterns. We developed an online evaluation framework to assess the utility of abstracted trajectories for non-RL experts in forming mental models of agents. The results from our pilot studies indicate that the interface was effective for non-RL experts in identifying agent behavior patterns. These studies also highlighted a preference for the slider over the map view and provided insightful feedback for further enhancing the visualization of abstracted trajectories. Future work will focus on refining the interface and expanding the evaluation framework to include a broader and more diverse participant base.

References

[1] D. Amir and O. Amir, “Highlights: Summarizing agent behavior to people,“ In Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems, Stockholm Sweden, 2018, pp. 1168–1176.

[2] A. Bora and S. Sharma, “A review on video summarization approcahes: recent advances and directions,“ 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018, pp. 601–606.

[3] M. Bostock, V. Ogievetsky, and J. Heer, “D3 data-driven documents,“ *IEEE transactions on visualization and computer graphics*, vol. 17, num. 12, pp. 2301–2309, 2011.

[4] E. Cakmak, M. Plank, D. S. Calovi, A. Jordan, and D. Keim, “Spatio-Temporal Clustering Benchmark for Collective Animal Behavior,“ In Proceedings of the 1st ACM SIGSPATIAL International Workshop on Animal Movement Ecology and Human Mobility (HANIMOB '21), NY, USA, 2021, pp. 5–8.

[5] F. Elizalde, L. E. Sucar, A. Reyes, and P. Debuen, “An MDP Approach for Explanation Generation,“ AAAI Workshop, Vancouver, British Columbia, Canada, 2007, pp. 28–33.

[6] K. Pearson F.R.S., “LIII. On lines and planes of closest fit to systems of points in space,“ *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, num. 11, pp. 559–572, Nov. 1901.

[7] D. Ha and J. Schmidhuber, “Recurrent world models facilitate policy evolution,“ Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, Canada, 2018, pp. 2455–2467, (2018).

[8] H. B. Haq, M. Asif, and M. B. Ahmad, “Video summarization techniques: a review,“ *Int. J. Sci. Technol.* vol. 9, num. 11, pp. 146–153, Nov. 2020.

[9] B. Hayes and J. A. Shah, “Improving robot controller transparency through autonomous policy explanation,“ In Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction, Vienna Austria, 2017, pp. 303–312.

[10] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “beta-vae: Learning basic visual concepts with a constrained variational framework,“ In International conference on learning representations, Caribe Hilton, San Juan, Palais des Congrès Neptune, Toulon, France, 2017.

[11] A. Hinterreiter, C. Steinparz, M. Schöfl, H. Stitz, and M. Streit, “Projection path explorer: exploring visual patterns in projected decision-making paths,“ *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 11, num. 3-4, pp. 1–29, Dec. 2021.

[12] T. Jaunet, R. Vuillemot, and C. Wolf, “DRLViz: Understanding decisions and memory in deep reinforcement learning,“ *In Computer Graphics Forum*, vol. 39, num. 3, pp. 49–61, 2020.

[13] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, and F. D. Velez, “Explainable reinforcement learning via reward decomposition,“ In IJCAI/ECAI Workshop on explainable artificial intelligence, Macau, China, Jan. 2019.

[14] O. Khan, P. Poupart, and J. Black, "Minimal sufficient explanations for factored markov decision processes," In Proceedings of the International Conference on Automated Planning and Scheduling, vol. 19. pp. 194–200, Thessaloniki, Greece, 2009.

[15] B. Kondo and C. Collins, "Dimpvis: Exploring time-varying information visualizations by direct manipulation," *IEEE transactions on visualization and computer graphics*, vol. 20, num. 12, pp. 2003–2012, Nov. 2014.

[16] C. W. Lynn, A. E. Kahn, N. Nyema, and D. S. Bassett, "Abstract representations of events arise from mental errors in learning and memory," *Nature communications*, vol. 11, article Num. 2313, May 2020.

[17] P. Madumal, T. Miller, L. Sonenberg, and F. Vetere, "Explainable reinforcement learning through a causal lens," In Proceedings of the AAAI conference on artificial intelligence, vol. 34, num. 3, pp.2493–2500, New York, New York, USA, 2020.

[18] S. McGregor, H. Buckingham, T. G. Dietterich, R. Houtman, C. Montgomery, and R. Metoyer, "Facilitating testing and debugging of Markov Decision Processes with interactive visualization," In 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). pp. 53–61, Atlanta, Georgia, USA, 2015.

[19] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform Manifold Approximation and Projection," *Journal of Open Source Software*, vol. 3, num. 29, Feb. 2018.

[20] Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, and H. Qu, "Understanding hidden memories of recurrent neural networks," In 2017 IEEE conference on visual analytics science and technology (VAST), pp. 13–24, Phoenix, AZ, USA, 2017.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, num. 7540, pp. 529–533, Feb. 2015.

[22] M. L. Olson, R. Khanna, L. Neal, F. Li, and W. Wong, "Counterfactual state explanations for reinforcement learning agents via generative deep learning," *Artificial Intelligence*, vol. 295, pp. 103455, Jun. 2021.

[23] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE transactions on visualization and computer graphics*, vol. 24, num. 1, pp. 667–676, Aug. 2017.

[24] O. Struckmeier, M. Racca, and V. Kyrki, "Autonomous generation of robust and focused explanations for robot policies," In 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), New Delhi, India, 2019.

[25] F. P. Such, V. Madhavan, R. Liu, R. Wang, P. S. Castro, Y. Li, J. Zhi, L. Schubert, M. G. Bellemare, J. Clune, "An atari model zoo for analyzing, visualizing, and comparing deep reinforcement learning agents," arXiv preprint arXiv:1812.07069, 2018.

[26] A. Tabrez, S. Agrawal, and B. Hayes, "Explanation-based reward coaching to improve human performance via reinforcement learning," In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 249–257, Daegu, Korea, 2019.

[27] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. "How to grow a mind: Statistics, structure, and abstraction," *science*, vol. 331, num. 6022, pp. 1279–1285, 2011.

[28] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, num. 86, pp. 2579–2605, Nov. 2008.

[29] J. van der Waa, J. van Diggelen, K. van den Bosch, and M. Neerincx, "Contrastive explanations for reinforcement learning in terms of expected consequences," arXiv preprint arXiv:1807.08706, 2018.

[30] J. Wang, L. Gou, H. Shen, and H. Yang, "DQNViz: A visual analytics approach to understand deep q-networks," *IEEE transactions on visualization and computer graphics*, vol. 25, num. 1 , pp. 288–298, Jan. 2019.

[31] Q. Wang, S. L'Yi, and N. Gehlenborg, "DRAVA: Aligning Human Concepts with Machine Learning Latent Dimensions for the Visual Exploration of Small Multiples," In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Hamburg, Germany, 2023.

[32] T. Zahavy, N. Ben-Zrihem, and S. Mannor, "Graying the black box: Understanding DQNs," In International conference on machine learning. PMLR, pp. 1899–1908, New York City, NY, USA, 2016.

[33] A. Dragan and S. Srinivasa, "Familiarization to robot motion," In Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction, Bielefeld Germany, 2014, pp. 366–373.