Advice Explanation in Complex Repeated Decision-Making Environments

Abstract

Humans that need to make decisions repeatedly in complex environments can gain from advice given by an automated assisting agent. However, due to the complexity of the environment and the long-term effect of a given advice, the decision maker may dismiss the advice and not take full advantage of its benefits. Advice explanation may improve the satisfiability and trust of the decision maker in the advice. We consider an automated assisting agent that integrates two deep learning-based models, an upstream prediction and a downstream Q-learning-based policy. As both models influence the advice, we propose to consider both when explaining it to the decision maker. We propose to reduce the state shown to the user, make the policy transparent through the precomputed policy, and compose them with an explanation of the upstream prediction model. We demonstrate our approach for idle taxi repositioning and show its effectiveness through computational experiments and a game-based user study. Although study participants do not follow the advice more often when compared to a baseline, they are significantly more satisfied, achieve a higher reward in the game, take less time to select an action, and use explanations of both models.

1 Introduction

Making decisions repeatedly in a dynamic environment is very challenging. An intelligent agent could improve human decision-making by providing advice. We consider an agent that provides advice through a learned policy that integrates two deep learning-based models, an upstream prediction and a downstream Q-learning-based policy. Humans are, in general, quite often not following machine-learning-based advice [**?**] and in particular, when the advice is based on two levels of deep learning black box models. Providing explanations may improve their acceptance and trust in the advice [**?**].

Most of the related work on eXplainable RL (XRL) focuses on the environment and algorithm-specific explanations, often not necessarily targeted at the general public but rather aimed at domain experts or researchers [Heuillet *et al.*, 2021; **?**]. Consequently, we focus on developing an explanation approach that is *generic* and *user-focused*. In particular, we propose an explanation approach that consists of four parts and their composition. First, we propose to way to choose the upstream prediction functions in a way that is closely related to the advice. Then, we propose a condensed representation of these functions to reduce the information load on the user. For presenting the policy, we propose to present future expected actions to help the user understand the long-term effect of his current advised action. Finally, we propose an explain the upstream prediction model via a classical local post-hoc perturbation-based eXplainable AI (XAI)-method like SHAP. Finally, we propose a visualization method to present all four components to the user in an easy-to-follow GUI.

In Section 4, we present our four component generalizable and modular approach towards explaining multi-black box Deep RL (DRL)-based systems to users. In Section 5, we apply it to idle taxi repositioning – along with matching and routing, one central function of ride-sharing. We select this application area because (1) it is an advising system that directly affects users – the drivers – (2) requires the latter to make repositioning decisions repeatedly, (3) uses DRL or more specifically typically Deep Q-learning (DQN) [Farazi *et al.*, 2021] – enables transferability to other cities and a longer time-horizon for optimization [Qin *et al.*, 2020] – and (4) additional upstream black-box models like a request estimator. We demonstrate the effectiveness of our approach via computational experiments (Section 6) and a game-based user study (Section 7). We discuss the major findings together with limitations and potential future work in Section 8.

Motivating example. Given an idle driver in a taxi service such as Uber or DiDi, a location advice might be provided to her: the service aims to redistribute its fleet proactively to future customers. To determine this advice, the taxi service can consider the future locations of its other taxi drivers – derived from the known schedules. However, the number of requests for each region can only be predicted via some potentially black–box model based on previously collected data. Both, the number of taxis and requests per region, can be fed into a DRL-based repositioner that computes the advice. As the driver loses time and money on the way to the proposed location and is not guaranteed to get a ride there, she might desire an explanation of the advice. As both models – request estimator and repositioner – influence the advice, the explanation needs to consider both.

2 Related Work

Although the field of Reinforcement Learning (RL) is heterogeneous but established, the field of XRL is also the former, but not the latter. [Puiutta and Veith, 2020] attempt to structure the literature in XRL by introducing two dimensions: In the first dimension they differentiate whether an approach is intrinsically explainable by using a transparent model or is explainable post-hoc; in the second dimension they distinguish approaches that explain locally or globally. As we explain advice given to a user for an existing model, we focus on *local post-hoc* explanations. However, none of the approaches included in [Puiutta and Veith, 2020] is composed of several deep learning-based models or explanations.

Very few works in XRL generate multiple explanations for one DRL agent. [Huber *et al.*, 2021] combine a local saliency map-based explanation with a global strategy summary explanation for an Atari agent. Both [Bayani and Mitsch, 2022] and [Sreedharan *et al.*, 2020] explain users an agent via a preset answer of questions with varying levels of abstractions in the answers. While [Bayani and Mitsch, 2022] explain DRL-based agents acting in toy environments, [Sreedharan *et al.*, 2020] explain multiple non-DRL-based components for a loan approval application. Other non DRL-based approaches that do generate multiple explanations are proposed by [Liao *et al.*, 2021]; the authors use multiple XAI methods such as feature importance to make the risk of hospital admission transparent and present their results side by side one another. To explain the recognition of vocal emotions, [Zhang and Lim, 2022] build five additional deep learning models and apply multiple XAI techniques, such as showing a saliency map. The only work we found that provides multiple explanations for multiple models is the one from [El-Sappagh *et al.*, 2021]: The authors first predict whether a person has Alzheimer's disease and attach another model to predict the stage of the disease; for explaining, they use SHAP, the feature importance of the underlying Random Forest (RF) models, and fuzzy rules to explain the predictions locally and globally.

In general, the number of approaches that generate multiple explanations for one or multiple deep learning models is very limited and heterogeneous. While some works provide advice – [Liao *et al.*, 2021; El-Sappagh *et al.*, 2021] – the majority explains some deep learning models not providing advice to users – [Huber *et al.*, 2021; Bayani and Mitsch, 2022; Sreedharan *et al.*, 2020; Zhang and Lim, 2022]. Some focus on explaining for end users – [Huber *et al.*, 2021; Sreedharan *et al.*, 2020; Zhang and Lim, 2022] – and others target expert users – [Bayani and Mitsch, 2022; Liao *et al.*, 2021; El-Sappagh *et al.*, 2021]. While the majority of the approaches considered evaluate the generated explanations without people – [Bayani and Mitsch, 2022; Sreedharan *et al.*, 2020; Liao *et al.*, 2021; El-Sappagh *et al.*, 2021] – only two evaluate with people – [Huber *et al.*, 2021; Zhang and Lim, 2022]. Also, most of the works focus on explaining non-DRL-based agents – [Sreedharan *et al.*, 2020; Liao *et al.*, 2021; Zhang and Lim, 2022; El-Sappagh *et al.*, 2021] – while two explain DRL-based agents – [Huber *et al.*, 2021; Bayani and Mitsch, 2022]; these works also explain agents in toy environments rather than those interacting in real-world applications.

Consequently, we consider the explanation of an *advising system with DRL agent and one or more upstream deep learning models* as an open research gap. To limit the scope of this paper, we will focus on *local post-hoc explanations for real-world applications* – like the idle taxi repositioning in our motivating example – and *end users* – e.g., taxi drivers – while developing our explanation approach. As regards the DRL approach, we focus on DQN which is commonly used for the repositioning of taxis [Farazi *et al.*, 2021] and in the field of autonomous driving.

3 Problem Definition

We consider a human user that can move in an undirected graph $G = (V, E)$ with V being a set of vertices and E a set of edges. The human goal is to maximize a reward. At every time step, the human is located at a location $l \in V$ and can take action $a \in A$ attempting to move on the graph G. A state $s \in S$ is associated with the properties of the entire environment and with the properties of the vertices in V. We use the notation $g_i(s), \forall s \in S$ for features that do not depend on the vertices and $f_j(s, v), \forall s \in S, \forall v \in V$ for features of the state that are relevant to a vertice v. $l(s) \in V$ indicates the location of the user in the state s. The state transition function $P(s, a, s'), \forall s, s' \in S, \forall a \in A$ from s to s' when taking action a is stochastic. The reward function $R(s, a, s'), \forall s, s' \in S, \forall a \in A$ depends on the state s, the action a, and the new state s'.

When considering the motivational example of idle taxi repositioning, G represents the road map of a city. At every point in time, the taxi driver selects a – like moving south from $l(s)$; this decision can be based on the state which is composed of a set of global features $\{g_1, g_2, ..., g_m\}$ like the weekday and another set of location-dependent features $\{f_1, f_2, ..., f_n\}$ such as the number of requests at the vs around $l(s)$. When collecting a passenger, the taxi driver receives a reward, e.g. 25 dollars.

To make a decision, the human can consider (1) its knowledge of the current state $s \in S$ and (2) advice provided through a learned policy $\pi : s \mapsto a, a \in A, \forall s \in S$ that maps each state s to action a. In particular, the policy has two levels: in the first level, there is a set of functions $\psi_j \in \Psi$; each function, given a state s and a vertice v, associates v with a value, that is, $\psi_j(s, v), \forall s \in S, \forall v \in V$. Some of these functions are estimated using deep learning. On the second level, the output of this first-level function is used by a Q value function that is learned via DRL: $Q_\Psi(s, l(s), a), \forall s \in S, \forall l(s) \in V, \forall a \in A$. The advice given to the human is $\arg\max_a Q_\Psi(s, l(s), a)$.

In idle taxi repositioning, we have two functions on the first level: ψ_d that extracts the demand for taxis and ψ_r that estimates the number of requests based on the previous number of requests via a neural network. Q_Ψ receives these outputs, $l(s)$, and an a; it is learned via deep Q-learning.

Explanation problem. Given the aforementioned sequential human-decision making problem in which a user u receives advice provided by a policy $\pi : s \mapsto a$, a user might have less information available – e.g., Ψ is not known by the user – or smaller computational capabilities. Consequently, the user's policy results in $\pi^u : s \mapsto a^u$ with $a \neq a^u$. The explanation problem tackled in this paper aims to produce an explanation ε so that $\pi^u : s \xrightarrow{\varepsilon} a$.

4 Explanation Approach

Understanding advice is challenging because (1) π is represented via Q_Ψ and both, Q and at least a subset of Ψ, are deep learning models – which are often hard to understand by users – (2) especially with a larger $|V|$ the size of the state $|s|$ might be overwhelming for users, and (3) users need to make decisions with a potential long-term effect repeatedly. Thus, in the following, we propose an explanation approach that consists of four parts and their composition.

4.1 Model Choices for Ψ

An important decision is to carefully choose the functions $\psi \in \Psi$. Previous approaches – like [Qin *et al.*, 2020; Haliem *et al.*, 2021] or the pipeline architecture described by [Grigorescu *et al.*, 2020] – compute the values of ψ simultaneously for all $v \in V$. That is, the functions are of the form $\psi(f_1, \ldots, f_n)$ which results in values for all $v \in V$. In this case, it is difficult to extract the contribution of each feature for the value associated with v. Thus, we propose to call ψ separately for each v, select features that are understandable by users, and make it return only one value for v – that is, $\psi(g_1, \ldots, g_m, f_1, \ldots, f_n)$.

E.g., when [Haliem *et al.*, 2021] reposition idle taxis, they make use of a function ψ to estimate the number of requests in the next time step in the whole city based on the previous demand. In this example, we propose to use an alternative ψ that estimates the number of requests on only one location based on fewer and more meaningful input features.

4.2 Condensed Representation of Ψ

Presenting all values that the functions $\psi_j \in \Psi$ associate with each vertice $v \in V$ can be overwhelming. Thus, we propose to integrate these values using some index I that compresses the number of values for each vertice. That is, $I(s, v) = \rho(\psi_1(s, v), \ldots, \psi_{|\Psi|}(s, v))$.

For example, in idle taxi repositioning, ρ could be the difference between the number of requests and taxis at v in state s; identifying a v with an undersupply becomes easier via ρ.

4.3 Transparent Policy

In order to reveal the long-term strategy of the policy, we propose to present the advice at any location $v \in V$ and not only at $l(s)$ to the user. Consequently, we compute the advice $\hat{a} = \arg\max_a Q_\Psi(s, l(s), a)$ for each location $v \in V$ and not only at $l(s)$. Similar to [Amir and Amir, 2018] we also make the certainty of the network in $\hat{a}$ transparent by computing the delta to the least promising action via $\hat{a} - \arg\min_a Q_\Psi(s, l(v), a)$. In addition, we compute a potential future path of limited length for the agent when following the advice while keeping everything in s fixed except for $l(s)$.

Realizing this part of our explanation in idle taxi repositioning is relatively straightforward via showing the advices via arrows for the whole city; the certainty of an advice can be incorporated into the color of the arrows.

4.4 Explaining Ψ

Another important component of the advising system is the subset of functions in Ψ that are represented via deep learning. For these ψs, we propose to present those features of s that contributed to ψ's value at vertices v. This is possible, given the way we defined ψ that outputs a value separately for each v. Such function ψ can be explained via a classical local post-hoc perturbation-based XAI-method like SHAP. We recommend to limit the number of vs for which the corresponding explanation is shown.

When we estimate the number of requests at a location v, we can show the most contributing features to a user to make the corresponding ψ more transparent

4.5 Compose the Explanation Parts

Besides carefully choosing Ψ, we present to the user of the advising system three aspects of the underlying policy: (1) the condensed representation of the ψ_is together, (2) the transparent policy, and (3) the explanations of the ψ_is. We propose to present (1) and (2) on the graph G; the former via arrows – advice – with different color intensity – certainty – and color each v via the index $I(s, v)$. Further, we propose to present the explanations of Ψ along the potential future path computed in (2) to limit the explanation size $|\varepsilon|$ shown to the user; the user can query only the locations available in this path.

	request estimation†	Repositioning‡
Haliem *et al.**	**1.22**	6.85
Ours	1.26	**7.24**

* adapted; † MAE in trips per cell; ‡ mean reward per step

Table 1: Agents performance; while for both – the request estimator and the repositioner – the test data is used for evaluation, for the repositioner, the mean reward per step is calculated over 100 runs.

5 Explaining Idle Taxi Repositioning

Before explaining idle taxi repositioning, we rebuild a repositioning approach orientating on one from the literature. Mostly, idle taxi repositioning is part of a system that also incorporates matching, scheduling, and routing. We favor the approach of [Haliem *et al.*, 2021] over others as it was developed over multiple papers, has – in contrast to most, like [Qin *et al.*, 2020] – made (at least most of) its source code available, and uses an accessible dataset. We show the results of approach adapted to our environment and the one we modified for explanation in Table 1; details of the implementation are described in Appendix A.

5.1 Rebuilding a Repositioning Agent

Dataset. We select the NYC taxi dataset. After outlier removal, around 186M trips between January 2015 and June 2016 remain. We generalize the degree-based start and end locations of trips to the indices of a grid; in particular, a 500m square grid. We use 26K 10-minute time steps. We separate the last two months for testing and split the remaining 16 month for training and validation with an 80/20 ratio; the latter two are split to enable learning Q based on Ψ.

Environment. In our environment, a taxi agent moves around in a city – represented as a 20×20 grid – aiming to serve requests. The taxi can move up to two cells in each direction or reside in its current location. The agent receives the state s which consists of the previous number of requests $r_{t-4:t}$ and the number of taxis d_{t+1} at every v as well as its location $l(s)$. Each episode lasts 54 ten-minute steps or a nine-hour shift. As regards the reward function R: When $r - t \geq 2$, the agent receives a reward of 20 (two passengers); $r - t = 1$ the reward is 10 (one passenger); if $r > 0$ and $r \leq d$ – the agent competes with other taxis – with a chance of $\frac{r}{t}$ a reward of 10 is given; in case the agent does (not) move the agent receives a reward of -1 (0). Whenever the reward is > 0, the agent is relocated to location randomly chosen from the distribution of drop-off locations. In each episode, the taxi starts at a random location and time. Our implementation of the environment is inspired by the OpenAI taxi environment.

Request estimation. [Haliem *et al.*, 2021] use ψ_d to extract the number of taxi from s and ψ_r to estimate the number of requests in 10 minutes at each v. ψ_r was learned via a three-layer convolutional neural network and achieved a Mean Absolute Error (MAE) of 1.22 trips per cell on the test data.

Repositioning. We train the repositioner via DRL in the repositioning environment. In particular, we use dueling double deep Q-learning as proposed by [Wang *et al.*, 2016] as it is closer to the state-of-the-art in RL than the double DQN approach used by [Haliem *et al.*, 2021]. After training, the repositioner – consumes $\psi_d, \psi_r, l(s)$ – achieves an average reward of 6.85 per step on the test data.

5.2 Explaining Repositioning Advice

Here, we apply our *composed explanation* approach proposed in Section 4 to explain advices in idle taxi repositioning to taxi drivers. Afterward, we also introduce a baseline explanation to which we compare ours. An example of both explanations is shown in Figure 1.

Replacing ψ_r. To explain the model ψ_r that estimates the number of requests at every $v \in V$ one could use a common XAI methods like SHAP – see [Lundberg and Lee, 2017] – producing a explanation of size $|\varepsilon| = 4 \times 20 \times 20 \times 20 \times 20 = 640K$. Besides being large, such explanation would be noisy and far from what a user expects. Thus, we reduce the number of output features heavily by making ψ_r only estimate the number of requests for one v. Further, we replace the original input features $r_{t-4:t}$ at every v by the location-dependent features index of v, $r_{t-4:t}$ at v, and the number of points of interest at v as well as location-independent time-related features like the weekday and weather-related ones. Next, we replace the convolutional neural network with a feed-forward fully-connected one. Thereby, we achieve a MAE of 1.26 trips per cell – which is only a slight increase of 0.04 – while reducing input size of ψ_r from 1600 to 20, the output size from 400 to 1, and $|\varepsilon|$ when applying a XAI method like SHAP from $64K$ to 20. After retraining the repositioner with the new ψ_r, the mean reward increases to 7.24 per step.

RT-index. To reduce the size of the input in Q with an intuitive representation, we propose the request-taxi index (RT-index). It combines the ratio between the estimated number of requests ψ_r and the number of taxis ψ_d as the all taxi drivers compete over the requests and the ratio between the mean number of requests $\bar{r}$ and ψ_r as the chance for getting a request is higher at locations with more requests. We weigh the two ratios via $\alpha \in [0, 1]$. We set alpha to 0.75 even though with another dataset a different value might be preferable. The corresponding formula is:

$$I_\Psi(s, v) = \psi_r(s, v) \left(\frac{\alpha}{\psi_d(s, v)} + \frac{1 - \alpha}{\bar{r}} \right) \quad \text{for } \alpha = 0.75$$

As a visual representation, we choose a heatmap that shows the RT-index for each location on a color scheme from red for 0 to green for values ≥ 3.

Transparent policy. To make the policy transparent, we iterate over all possible taxi locations $l \in V$ and pass the corresponding location with s to $\arg\max_a Q_\Psi(s, l, a)$. Thus, we collect the most promising action for each l. To visualize these, we plot an arrow from each location with the length and direction of the corresponding action. To incorporate the certainty of the agent, we also collect

$$\Delta_l = \max_a Q_\Psi(s, l, a) - \min_a Q_\Psi(s, l, a)$$

for each l. As a visual representation, we select black for arrows on top of the heatmap generated via the RT-index with a high action certainty and let the color fade out with decreasing certainty. To make the color consistent over all locations, we use min-max normalization with Δ_l for the local and Δ_g for the global delta:

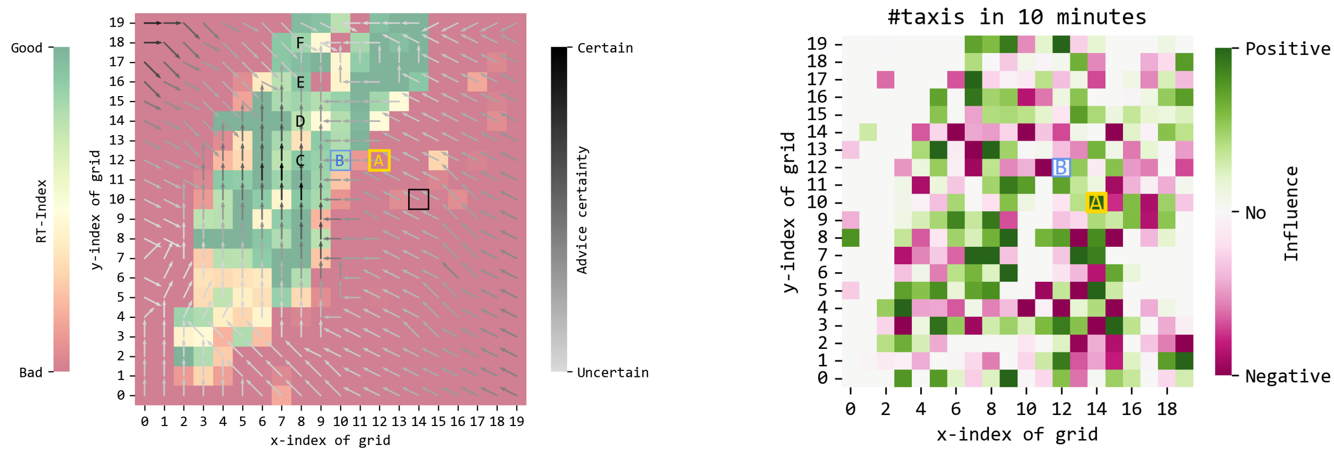
$$\frac{\Delta_l - \min \Delta_g}{\max \Delta_g - \min \Delta_g}$$

Further, we compute a potential future path for up to five locations. The resulting locations are plotted on the map via the letters $B, C, \ldots$ – A is reserved for the location of the taxi – and selectable via buttons that update a table with the six most important features.

Explaining ψ_r. After replacing ψ_r, we can simply apply SHAP to the single-cell request estimation model. To reduce the mental load of the users, list the six most important features as well as their order while omitting their actual values and influence. We generate this explanation for each v along the potential future path and offer the user to select one of the corresponding explanations via buttons.

5.3 Baseline

In our composed explanation, we have a compositional view of the advising system explaining each component of the advising system solely and then joining the explanations. In



(a) Composed explanation

(b) Baseline explanation

Figure 1: We show the composed explanation without its request estimation part in (a) and the baseline explanation for the number of taxis in 10 minutes – the explanations for the request over the last 40 minutes are of a similar kind – in (b)

contrast to our compositional view, related work generally has a one-model view that does not differentiate between $\psi_1, \psi_2, \ldots, \psi_{|\Psi|}$ and Q but takes the whole system as one function. In the following, we describe the selection of such baseline XAI method, the configuration of the selected method, and our chosen visual representation. An example explanation via the baseline is shown in Figure 1.

Selection. As we explain locally and post-hoc, we select a corresponding XAI method. Because our composed explanation is mainly visual, we select a corresponding baseline. As the state s is relatively big as well as image-like and others also use perturbation-based XAI methods to generate saliency maps for DRL – see e.g. [Huber *et al.*, 2022] – we select such. Based on the results of [Huber *et al.*, 2022] – who compare several potential XAI methods – we first tried Sarfa, a method proposed by [Puri *et al.*, 2020]. Unfortunately, these results were not reasonable with Q_Ψ. Another XAI method included by [Huber *et al.*, 2022] is LIME – see [Lundberg and Lee, 2017]. LIME allowed us to explain only the advice, produced more reasonable explanations than Sarfa, and takes reasonable time to explain.

Configuration. The explanation size is 2000 as we have one value for the number of taxis and four for the number of requests at each $v \in V$ and fix the taxi location as well as the advice. We select the number of perturbation samples considered for explaining to 1000 as this produces reasonable explanations in a decent time – Mean (M) of 10.35 seconds. The background data is taken from the dataset used for training and we select 25 samples at a similar hour and day as the time that shall be explained.

Visual representation. When using saliency maps, many approaches plot those on top of the state. As the saliency values would make the state invisible, we present the explanations beside the state. We decided to exclude the actual influence values and show a scale from *negative* to *positive* influence instead to reduce the mental load of the user; while a negative/positive value refers to a negative/positive influence of the corresponding state value on taking the advice when being at the given location.

6 Experimental Results

Here, we report the size of the networks – request estimator and repositioner – the number of input features given to the explanation models, the explanation size, and the execution time with several variants of the environment for idle taxi repositioning. In particular, we vary the size of the city in the environment and thereby indirectly the number of states $|S|$. As $|S| = 150^{10^2 \times 2} \approx 1.65 * 10^{4}35$ for $|V| = 100$, we only report the number of nodes $|V|$ instead of $|S|$. The highest $|V|$ we consider is 6400 which would corresponds to a grid cell size of 125m when we consider the same area. The second variation of the environment is the modification of the action size $|A|$. While $|A| = 9$ refers to the agent's ability to move one cell in each direction, $|A| = 25$ refers to moving up to two cells in each direction.

Network size, #input features, and explanation size. As shown in Table 2, the network size is primarly influenced by $|V|$ and neither by the explanation setting – composed or baseline – nor $|A|$. As the baseline uses a whole-city request estimator, the network size is slightly larger compared to the single-cell case. As the influence of $|A|$ on the network size is small and there is none on the number of input features and the explanation size, we do not list $|A|$ for $|V| > 100$ in Table 2. Obviously, the number of input features and the explanation size increases linearly with $|V|$. The size of the composed explanation is always smaller than that of the baseline. In all composed settings, the size is mainly driven by the RT-Index and the arrows – the table-based explanation of the upstream request estimator has a low influence on the number of input features and the explanation size. These results

		Network size		#input features		Explanation size	
$\|V\|$	$\|A\|$	Composed	Baseline	Composed	Baseline	Composed	Baseline
100	9	**3.31M**	3.35M	**0.32K** (0.20K, 0.20K, 0.12K)	0.50K	**0.24K** (0.10K, 0.10K, 36)	0.50K
100	25	**3.33M**	3.37M	**0.32K** (0.20K, 0.20K, 0.12K)	0.50K	**0.24K** (0.10K, 0.10K, 36)	0.50K
400	9	**21.14M**	21.18M	**0.52K** (0.80K, 0.80K, 0.12K)	2K	**0.84K** (0.40K, 0.40K, 36)	2K
1600	9	**120.23M**	120.27M	**3.32K** (3.20K, 3.20K, 0.12K)	8K	**3.24K** (1.60K, 1.60K, 36)	8K
6400	9	**361.14M**	361.18M	**12.92K** (12.8K, 12.8K, 0.12K)	32K	**12.84K** (6.40K, 6.40K, 36)	32K

Table 2: Network size, number of input features given to the explanation approach, and size of the explanation depending on the number of nodes $|V|$ and actions $|A|$ in the environment; for the number of input features and the explanation size, we show the values for the RT-index, the arrows, and the table separately in the brackets.

$\|V\|$	$\|A\|$	Composed (M±SD)	Baseline (M±SD)
100	9	**0.87±0.44**	7.20±0.86
100	25	**0.98±0.27**	7.42±0.52
400	9	**1.30±0.36**	10.00±0.71
1600	9	**5.89±0.31**	18.28±0.68
6400	9	**25.51±1.91**	41.18±1.13

Table 3: Execution time in seconds with varying number of nodes $|V|$ and actions $|A|$ for the composed and baseline explanation; M is the mean execution time in seconds over 10 runs and SD the corresponding standard deviation.

are limited because in reality the performance of an agent also depends on the network architecture; a larger state space might require more trainable parameters and therefore a network size larger than the one listed in the table.

Execution time. As shown in Table 3 (1) our approach can be applied to different environments, (2) its execution time is lower than that of the baseline in all considered cases, and (3) the size of our composed explanation is in all cases less than half compared to that of the baseline explanation. The execution time of the baseline depends on the number of samples considered for perturbation – 1000 in our case; the larger this number is chosen, the larger is the execution time of the baseline. Similar to before we omit more options for $|A|$ as the number of actions does only slightly depend on $|A|$.

7 Game-Based User Study with Questionnaire

7.1 Study Design

When designed appropriately, explanations have the potential to increase properties like the satisfaction of a user that interacts with an AI-based system. To evaluate the effectiveness of our explanation approach, we developed a game – see Figure 4 – in which participants of our study can drive through a city aiming to maximize their reward as taxi drivers. In this game, the participants receive advices provided by an agent that has learned Q_Ψ and an explanation – either ours or the baseline. At each time step a participant can either follow the advice or select one of the other actions. Besides observing the achieved reward, the degree to which advices are followed, and the time taken to select an action, we conduct a questionnaire with 31 questions.

Structure. During the study, participants go through the following steps: (1) Introduction of the study and the game, (2) ten steps of playing with one explanation method, (3) questions related to the subjective usage of the adivces, (4) ten steps of playing with the other explanation method, (5) questions related to the subjective usage of the adivces, (6) questions related to the explanations provided, and (7) demographic questions To ensure data quality, after the description of the game, we incorporate three attention-check questions about a participant's understanding of the environment.

Participants. We run our study with 27 participants that are fluent in English, over the age of 18, and do not have color blindness – the latter might affect their ability to see the generated explanations correctly. The M age of the participants is 28.81 years with a Standard Deviation (SD) of 8.39 years. 41% of the participants reported are female, 59% are male. 87% of the participants reported living in Germany. The study was conducted in December 2022 and January 2023.

Independent variables. Our within-subject study shows two explanation settings in one scenario – starting date and time of the day – to each participant. Consequently, each participant plays twice in the game before answering questions about both explanation settings. To half of the participants, the explanation is shown first and the baseline variant second; for the other half, the order is reversed. To gain better insights into the behavior of participants, we ask them to rate how confident they were to choose better than the provided advice and what their strategy was.

Dependent measures. Based on [Hoffman *et al.*, 2019], we evaluate the generated explanations via the *satisfaction* with each explanation presented, composed of *understanding*, *satisfaction*, *detail*, *completeness*, *usage*, *usefulness*, *accuracy*, and *trust*. We ask the participants to rate all questions related to explanation satisfaction on a five-point Likert scale. Further, we measure the achieved *reward*, the degree to which the participants *followed the advices*, and how much time they took to perform a step. Since as shown in Section 6 the execution time for creating the baseline explanation is on average 9.21 seconds higher than that of the composed one, we subtract $10.35 - 1.14 = 9.21$ seconds to enable a fair comparison between the two explanation settings.

Hypothesis. With the described study, we investigate the following hypotheses:

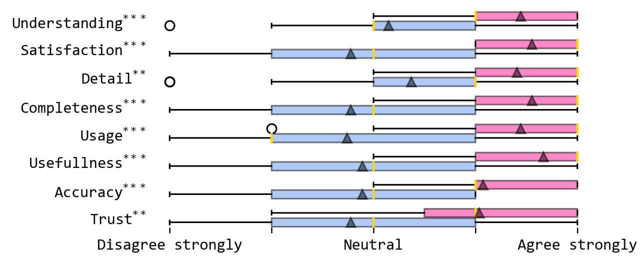


Figure 2: Questionnaire results for dimensions of the satisfaction scale by [Hoffman *et al.*, 2019] as boxplot for our composed explanation (pink) and the baseline (blue) – the median is represented via a gold line, the mean via a triangle; $^{*}*$ indicates $0.001 < p \leq 0.01$ and *** indicates $p \leq 0.001$.

- H1: The proposed composed explanation for repositioning achieves a *higher satisfaction* (see [Hoffman *et al.*, 2019]) than the baseline alternative.
- H2: Compared to the baseline explanation of repositioning, taxi drivers achieve a *higher reward* with the composed explanation.
- H3: Taxi drivers who are presented the composed explanation *follow the advices to a higher degree*, when compared to the baseline explanations.
- H4: Taxi drivers require *less time* when taking actions with the composed explanation compared to the baseline alternative.

7.2 Result Analysis

To investigate H1, we select a Wilcoxon signed-rank test; for H2 to H4, we select a paired sample t-test. For all tests, we set the significance level α to 0.05 because our sample size is relatively small.

H1 – Satisfaction. As shown in Figure 2, the null hypothesis of the tests can be rejected for all dimensions of the used satisfaction scale – highest p-value for trust with 0.0029. *Therefore, the data supports H1.*

H2 – Reward. While the participants achieved a M reward of around 89.89 with an SD of around 18.41 with the baseline explanation, they achieved a M reward of 97.78 (SD of 13.26) – the difference was higher when the participants first played with the composed setting. However, the difference was not statistically significant ($t = -1.7315$, $p = 0.0952$). As M is higher with the composed explanation, the SD is lower, and the difference is not significant, we argue that *the data partially supports H2.*

H3 – Degree of following. From the 27 participants, 13 followed more when presented with the baseline, ten more with the composed explanation, and four participants followed to the same degree in both settings. As the mean of following between baseline and composed also only slightly differs – 46% of following compared to 42% – the corresponding test could not underline the difference via statistical significance ($t = 0.9777$, $p = 0.3372$). *Consequently, the data does not support H3.*

H4 – Less time. On average, participants took less time to take actions when the composed explanation was provided ($M = 38.61$, $SD = 16.18$) compared to the baseline explanation ($M = 53.77$, $SD = 27.78$). This difference is also statistically significant ($t = 3.121$, $p = 0.0044$). *Thus, the data supports H4.*

Usage of explanation of upstream black-box. Overall, 70% of the participants used the explanation of the upstream black box or table. The usage spans over 20% of all game steps taken in the study. 41% of the participants used the table more than once. One person requested to see the table for more locations.

7.3 Discussion

Based on the satisfaction scale, people clearly favored our composed explanation over the baseline alternative. Even though with the former explanation, they achieved on average a higher reward, this result is not statistically significant. However, the comparison is slightly unfair as for the baseline the state is directly visible; this would be unrealistic as a taxi service is unlikely to want to disclose this knowledge to its taxi drivers. Most likely, not showing the state would change the results in favor of H2. Further, the reward does heavily dependent on a stochastic function.

The interpretation of the results as regards the degree of following the advices is not straightforward. On the one hand, the results might be blurred by the stochastic reward function leading to people following less/more based on the achieved reward. On the other hand, people might feel comfortable with the provided information and decide to make decisions on their own. The other way around this could mean that people feeling overwhelmed by the baseline follow the advices to reduce their mental load. This claim is in line with the fact that participants required more time to select an action with the baseline explanation. However, the aforementioned argumentation is weakened as the time required to take an action is only a proxy for the mental load of participants.

The results as regards the usage of the explanation for the upstream request estimation model indicate that making such explanations optionable – for instance by selecting which explanation aspect shall be shown – for each user. Another potential reason why the table-based explanation was not used more might be that the participants played so less that their mind was occupied by the other explanation aspects. Consequently, the table-based explanation might be more relevant once people are familiar with the game.

8 Conclusion and Future Work

A Details of Repositioning Agent

A.1 Dataset

A.2 Request Estimation

Original. The request estimator proposed by [Haliem *et al.*, 2021] consists of three convolutional layers that transform the previous number of requests per grid cell for the last four time steps – input shape of $4 \times 20 \times 20$ – into a prediction of the number of requests for taxis in the next 10 minutes – output

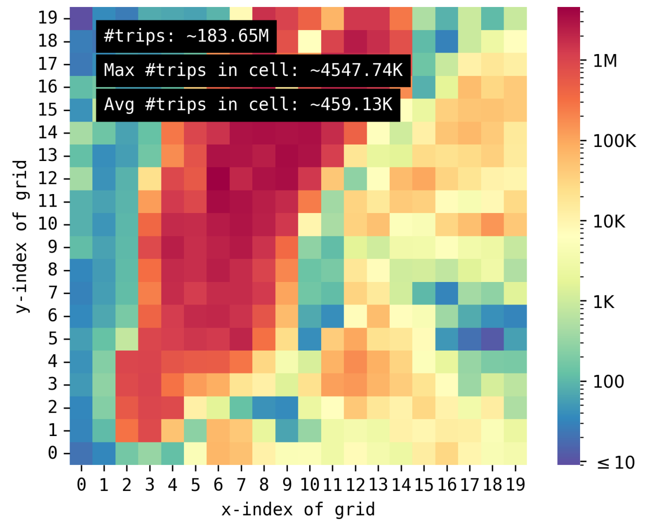


Figure 3: Distribution of the number of taxi trips in the NYC yellow taxi trip dataset in 2015 and 2016 visualized on a logarithmic scale via a 500m square grid.

shape of 20×20. The kernel sizes are 3, 5, and 7; the number of channels is set to 32 and 64. With a learning rate of 0.01 and 30 epochs of training, the request estimation model achieves a MAE of 1.22 trips per cell on our test data.

Modified. This request estimator consists of five fully-connected layers with 20, 128, 64, 32, and 16 neurons. With a learning rate of 0.001 and 15 epochs of training, we achieved a MAE of 1.26 trips per cell. As input features, we used: (1) x-index at v, (2) y-index at v, (3) #requests 30 minutes ago at v, (4) #requests 20 minutes ago at v, (5) #requests 10 minutes ago at v, (6) #requests now at v, (7) #points of interests at v, (8) hour, (9) minute, (10) weekday, (11) month, (12) temperature, (13) wind, (14) humidity, (15) air pressure, (16) view, (17) snow, (18) precipitation, (19) cloudy, and (20) holiday.

A.3 Repositioning

We train the repositioner in the taxi repositioning environment via reinforcement learning. Similar to [Haliem *et al.*, 2021] and related work in taxi repositioning, we use model-free off-policy Q-learning to train the repositioner in our environment. In particular, we use dueling double deep Q-learning as proposed by [Wang *et al.*, 2016] as it is closer to the state-of-the-art in RL than the double DQN approach used by [Haliem *et al.*, 2021]. Both networks – the policy and target one – consist of three convolutional layers with corresponding kernel sizes of 5, 5, and 3; the number of filters is set to 16, 32, and 64. The next layer is a fully connected one with 64 * 12 * 12 + 2 = 9218 input and 1024 output neurons. Both the value and advantage layers receive this as input. As we do not aim to outperform other repositioning approaches but to enable explaining them, we tune the hyperparameter manually, resulting in (1) a learning rate of 0.001, (2) a gamma of 0.99, (3) an episode decay of 675 to adjust the exploration–exploitation trade-off, (4) a target network update rate of 11,

Category	Content	Overall ($n = 27$)	
		n	**%**
Gender	Female	11	41
	Male	16	59
	No gender	-	-
	No answer	-	-
Age	< 21	3	11
	21 to 30	17	63
	31 to 40	4	15
	41 to 50	1	4
	51 to 60	-	-
	> 60	1	4
	No answer	1	4
Education	No training yet	-	-
	Secondary school	1	4
	High school diploma	3	11
	Vocational training	2	7
	Bachelor degree	8	30
	Master degree	10	37
	Doctorates	3	11
	Other	-	-
	No answer	-	-
Country	Germany	17	63
	Israel	6	22
	United States	3	11
	Finland	1	4
	No answer	-	-

Table 4: Profile of respondents

(5) and a replay memory size of 15K transitions. As shown in the first row of Table 1, the repositioner achieves an average reward of 6.85 per step.

B Details of User Study

B.1 Profile of Respondents

See Table 4.

B.2 Description of Game Given to Participants

Before each participant starts to play the game, we describe that he/she is a taxi driver that aims to maximize his/her reward. Further, we describe the following aspects: (1) the current location – yellow square – the advice – blue square – and the last location – black square – (2) that at each step a movement of up to two cells or staying at the current location is possible via the action buttons, (3) the reward function, (4) the available information fields like the accumulated reward, (5) the usage of the webpage – minimizing/maximizing of graphics and description pane – and (6) the description of the explanation configuration.

B.3 GUI of Game

Ethical Statement

This study described in Section 7 and Appendix B was approved by the internal review board of Bar-Ilan University prior to conducting our study.

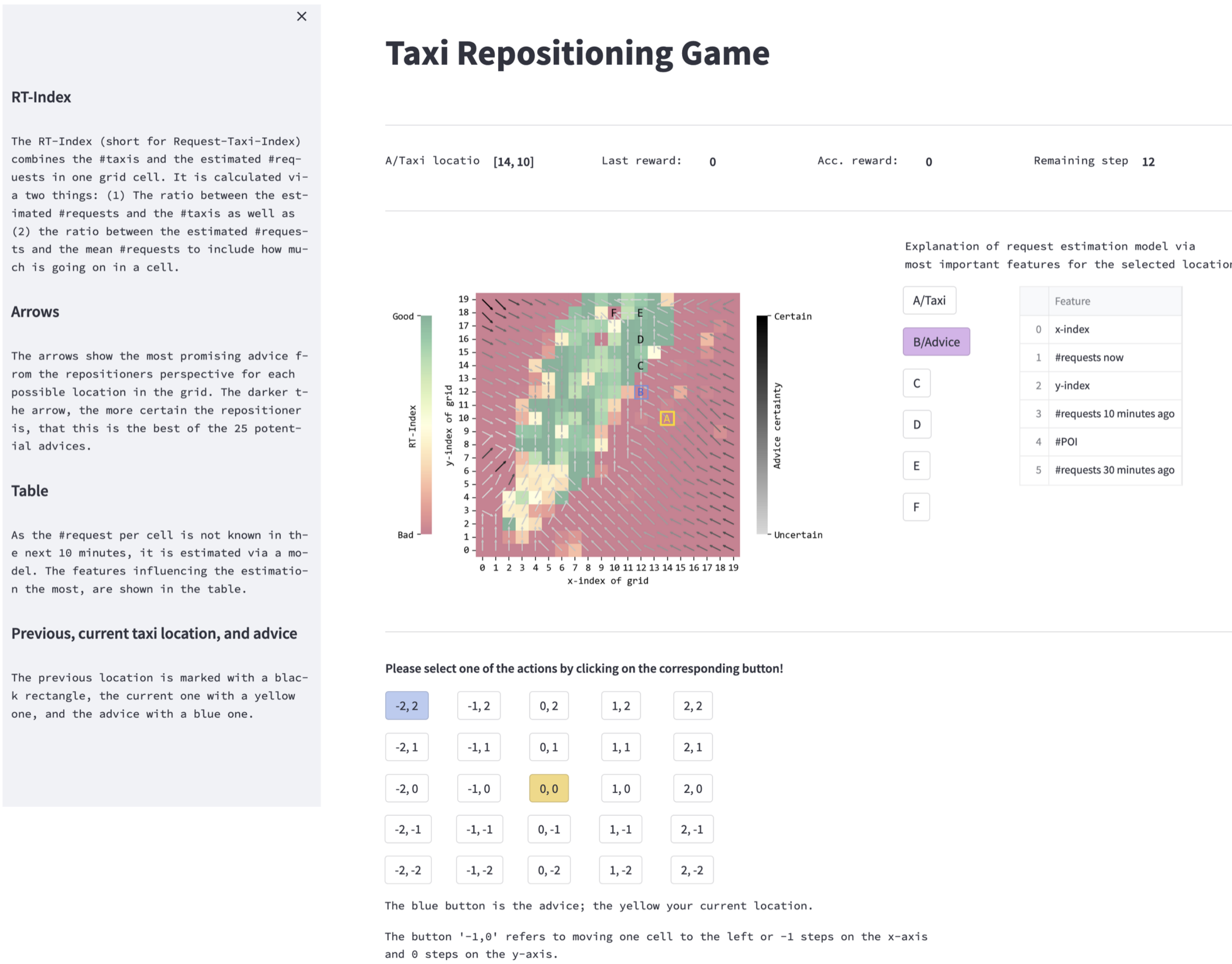


Figure 4: GUI of the game with the composed explanation method

Acknowledgments

References

[Amir and Amir, 2018] Dan Amir and Ofra Amir. HIGHLIGHTS: Summarizing agent behavior to people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 1168–1176, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[Bayani and Mitsch, 2022] David Bayani and Stefan Mitsch. Fanoos: Multi-resolution, multi-strength, interactive explanations for learned systems. In *Lecture Notes in Computer Science*, pages 43–68. Springer International Publishing, 2022.

[El-Sappagh *et al.*, 2021] Shaker El-Sappagh, Jose M. Alonso, S. M. Riazul Islam, Ahmad M. Sultan, and Kyung Sup Kwak. A multilayer multimodal detection and prediction model based on explainable artificial intelligence for Alzheimer's disease. *Scientific Reports*, 11(1), January 2021.

[Farazi *et al.*, 2021] Nahid Parvez Farazi, Bo Zou, Tanvir Ahamed, and Limon Barua. Deep reinforcement learning in transportation research: A review. *Transportation Research Interdisciplinary Perspectives*, 11:100425, September 2021.

[Grigorescu *et al.*, 2020] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, April 2020.

[Haliem *et al.*, 2021] Marina Haliem, Ganapathy Mani, Vaneet Aggarwal, and Bharat Bhargava. A Distributed Model-Free Ride-Sharing Approach for Joint Matching, Pricing, and Dispatching Using Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(12):7931–7942, December 2021.

[Heuillet *et al.*, 2021] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214:106685, February 2021.

[Hoffman *et al.*, 2019] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. Metrics for Explainable AI: Challenges and Prospects, February 2019.

[Huber *et al.*, 2021] Tobias Huber, Katharina Weitz, Elisabeth André, and Ofra Amir. Local and global explanations of agent behavior: Integrating strategy summaries with saliency maps. *Artificial Intelligence*, 301:103571, December 2021.

[Huber *et al.*, 2022] Tobias Huber, Benedikt Limmer, and Elisabeth André. Benchmarking Perturbation-Based Saliency Maps for Explaining Atari Agents. *Frontiers in Artificial Intelligence*, 5:903875, July 2022.

[Liao *et al.*, 2021] Q. Vera Liao, Milena Pribić, Jaesik Han, Sarah Miller, and Daby Sow. Question-driven design process for explainable AI user experiences. April 2021.

[Lundberg and Lee, 2017] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[Puiutta and Veith, 2020] Erika Puiutta and Eric M. S. P. Veith. Explainable reinforcement learning: A survey. In *Lecture Notes in Computer Science*, pages 77–95. Springer International Publishing, 2020.

[Puri *et al.*, 2020] Nikaash Puri, Sukriti Verma, Piyush Gupta, Dhruv Kayastha, Shripad Deshmukh, Balaji Krishnamurthy, and Sameer Singh. Explain your move: Understanding agent actions using specific and relevant feature attribution. In *International Conference on Learning Representations*, 2020.

[Qin *et al.*, 2020] Zhiwei (Tony) Qin, Xiaocheng Tang, Yan Jiao, Fan Zhang, Zhe Xu, Hongtu Zhu, and Jieping Ye. Ride-Hailing Order Dispatching at DiDi via Reinforcement Learning. *INFORMS Journal on Applied Analytics*, 50(5):272–286, September 2020.

[Sreedharan *et al.*, 2020] Sarath Sreedharan, Tathagata Chakraborti, Yara Rizk, and Yasaman Khazaeni. Explainable composition of aggregated assistants. November 2020.

[Wang *et al.*, 2016] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1995–2003, New York, NY, USA, 2016. JMLR.org.

[Zhang and Lim, 2022] Wencan Zhang and Brian Y Lim. Towards Relatable Explainable AI with the Perceptual Process. In *CHI Conference on Human Factors in Computing Systems*, pages 1–24, New Orleans LA USA, April 2022. ACM.