Verification and Validation of AI Systems Using Explanations

Saaduddin Mahmud[1], **Sandhya Saisubramanian**[2], **Shlomo Zilberstein**[1]

[1]University of Massachusetts Amherst, USA
[2]Oregon State University, USA

smahmud@umass.edu, sandhya.sai@oregonstate.edu, shlomo@umass.edu

Abstract

Verification and validation of AI systems, particularly learning-enabled systems, is hard because they often lack formal specifications and rely instead on incomplete data and human subjective feedback. Aligning the behavior of such systems with the intended objectives and values of human designers and stakeholders is very challenging, and deploying AI systems that are misaligned can be risky. We propose to use both existing and new forms of explanations to improve the verification and validation of AI systems. Toward that goal, we present a framework, in which the agent explains its behavior and a critic signals whether the behavior and the explanation are acceptable. In cases where either of them is not accepted, the agent gathers feedback about the behavior and the explanation, which is then used to improve the system's alignment. This approach represents a shift from traditional AI methods, where feedback typically focuses solely on behavior without considering the underlying explanations. We discuss examples of this approach that proved to be effective, and how to extend the scope of explanations and minimize human effort involved in this process.

The Alignment Problem

Value alignment is aimed at creating AI agents whose behaviors and goals align with the intended objectives and values of human designers and stakeholders. Developing verification methods for value alignment is critical as AI systems become prevalent in complex domains (Zilberstein 2015; Dietterich 2017). What complicates value alignment is that AI systems often optimize an unspecified set of competing objectives, which it must balance based on limited amount of data. The data captures the true objectives only implicitly and can be noisy. For example, *inverse reinforcement learning* (IRL) (Sutton and Barto 1998)—a common approach for *learning from demonstrations* (LfD) (Argall et al. 2009)—is designed to retrieve a reward function that motivates some observed behavior, allowing agents to generalize observed behavior to unseen situations.

Assuring reward alignment is difficult, particularly when the reward function is acquired from sample trajectories. For example, consider situations where the reward function is learned from ranked sub-optimal trajectories, and the trajectory dataset only covers a subset of states. Moreover, the dataset may contain spurious state feature correlations, causing the agent to learn a misaligned reward. This can occur, for example, when training an autonomous vehicle (AV) using data collected at a particular geo-location and then deploying it in areas with different characteristics.

Techniques for improving value alignment include efforts to avoid *negative side effects* of AI systems (Saisubramanian, Kamar, and Zilberstein 2022), the *inverse reward design* approach (Hadfield-Menell et al. 2017), and *value alignment verification* (VAV) with a minimum number of queries (Brown, Schneider, and Niekum 2021). Our approach adds important capabilities, most notably using explanations (e.g., rankings of reward explanations) as feedback, the ability to correct misaligned reward and to verify performance in novel situations.

We propose a novel approach called *explanations for value alignment*, where an agent not only explains its behavior and objectives but also collects feedback on both its behavior and the explanations themselves. This represents a significant shift from traditional AI methods, which typically focus only on gathering feedback about behavior. The types of explanations that can be utilized in this framework range from feature attributions to natural language explanations. A critic—whether human or an automated function—can evaluate these explanations either quantitatively, using similarity metrics, or qualitatively through more abstract reasoning, providing valuable feedback.

Rich forms of explanations can be used in conjunction with this approach, particularly the commonly used explanations that rely on feature attribution to uncover the relationship between input features and output decisions. Examples include LIME (Ribeiro, Singh, and Guestrin 2016), Gradient as Explanation (GaE) (Tayyub, Sarmad, and Schönborn 2022), and saliency maps (Simonyan, Vedaldi, and Zisserman 2014). Beyond feature attribution, other automated explanation generation methods exist, such as model reconciliation (Chakraborti et al. 2017) and policy summarization (Amir, Doshi-Velez, and Sarne 2019). Although some of these methods have been used for value alignment verification (Huang et al. 2018; Tabrez, Agrawal, and Hayes 2020), none have been employed to actively *improve* alignment. Moreover, natural language explanations could harness rapid advancements in language-based abstraction to further improve value alignment through this framework.

Figure 1 offers a simple example of the proposed approach in which an AV approaches a pedestrian walking

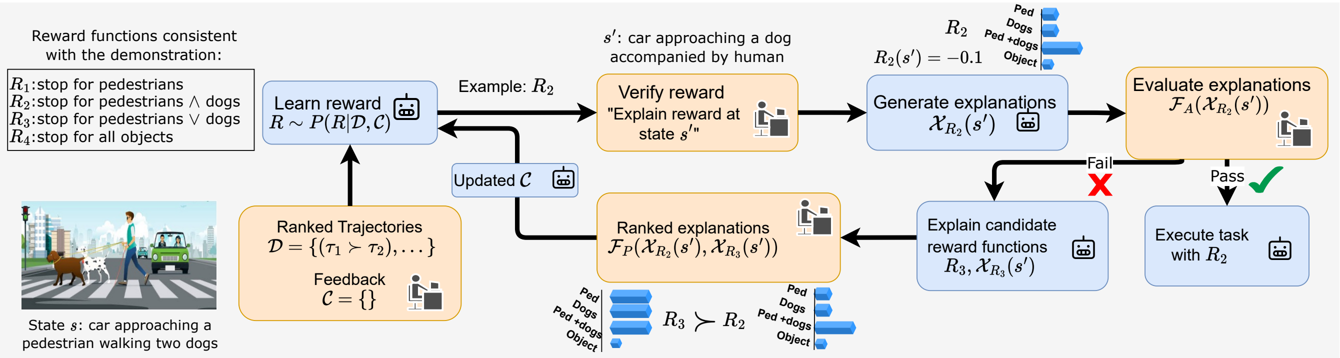
Reward functions consistent with the demonstration:
R_1:stop for pedestrians
R_2:stop for pedestrians $\wedge$ dogs
R_3:stop for pedestrians $\vee$ dogs
R_4:stop for all objects
Learn reward $R \sim P(R|\mathcal{D}, \mathcal{C})$
Example: R_2
s': car approaching a dog accompanied by human
Verify reward "Explain reward at state s'"
R_2
$R_2(s') = -0.1$
Ped
Dogs
Ped +dogs
Object
Generate explanations $\mathcal{X}_{R_2}(s')$
Evaluate explanations $\mathcal{F}_A(\mathcal{X}_{R_2}(s'))$
Fail
Pass
Updated $\mathcal{C}$
Explain candidate reward functions $R_3, \mathcal{X}_{R_3}(s')$
Execute task with R_2
Ranked explanations $\mathcal{F}_P(\mathcal{X}_{R_2}(s'), \mathcal{X}_{R_3}(s'))$
Ranked Trajectories $\mathcal{D} = \{(\tau_1 \succ \tau_2), \dots\}$
Feedback $\mathcal{C} = \{\}$
State s: car approaching a pedestrian walking two dogs
Ped
Dogs
Ped +dogs
Object
$R_3 \succ R_2$
Ped
Dogs
Ped +dogs
Object

Figure 1: An example of reward verification and learning using explanations.

two dogs. Suppose that the training data includes trajectories in which the driver always stops when a human is crossing the street with dogs. Since dogs are often accompanied by humans, the rare case of encountering dogs alone might be missing from the dataset. Consider four different reward functions consistent with the trajectory dataset, $(R_i, 1 \leq i \leq 4)$, each with the same negative reward for not stopping in this case. R_1 does not account for dogs, R_2 rewards stopping for pedestrians with dogs, R_3 rewards stopping for pedestrians or dogs, and R_4 rewards stopping for all objects, including leaves or a plastic bag on the road. Without additional information, the AV may randomly learn one of these reward functions (say R_2), however, R_3 represents the intended reward. When operating based on R_2, the AV may not stop for dogs unaccompanied by humans. This example illustrates the reward ambiguity stemming from the incomplete trajectory dataset and the danger of using misaligned rewards. While increasing the diversity of the trajectories in the dataset may help to some extent, it is often impractical or risky to demonstrate certain trajectories.

Our framework can utilize verification tests in the form of a query: "explain the reward at state s," and explanations in the form of feature attribution. The verification test may be "explain the reward when the AV encounters a dog accompanied by humans," to which the agent may respond with its reward value and feature attributions (for R_2) indicating a low weight for the 'dogs' feature, such as "$R(s) = -0.1$; *Feature attributions*: pedestrian=0.25, dogs=0.15, pedestrian&dogs=0.5, object=0.1". This reveals a potential weakness of the model in the counterfactual scenario in which the dog is not accompanied by a human (missing from the dataset). When this verification test fails, the agent explains another candidate reward function (for example, R_3). The critic then selects an explanation that is similar to the explanation that their intended reward would generate (R_3 in this case), indicating that R_3 is ranked over R_2 and the desired behavior is to stop for pedestrians or dogs. This example highlights two key advantages of our approach: (1) it *exposes wrong reward estimation* in novel situations that do not appear in the dataset, and (2) it *improves the alignment* of the reward function offline without requiring additional trajectory samples containing the novel situation, and before the agent encounters that situation.

This approach to value alignment departs from the more common prediction of competency (e.g., based on accuracy or F1 score) that can often hide important shortcomings in agent behavior. The assessment can be both qualitative—how good the explanations are, and quantitative—how similar the explanations are using some metric (Mahmud et al. 2023). Explanation-based assessments are simple and effective for non-expert users to evaluate agent alignment, unlike the expertise required for generating formal specifications. This form of verification is not only useful for identifying misalignment, but also for building trust with humans and helping them to identify the system's strengths and weaknesses, or acquiring competence models (Basich et al. 2023).

Case Study: REVEALE

We present a case study called REVEALE that demonstrates the application of explanations for human-assisted alignment verification, while simultaneously guiding the learner to improve alignment using feedback on those explanations.

Problem Formulation

REVEALE is designed for sequential decision problems modeled as a Markov decision process (MDP) M, represented by a tuple $M = (S, A, T, R, S_0, \gamma)$, where S is a set of states, A is a set of actions, $T : S \times A \times S \rightarrow [0, 1]$ is the transition function, $R : S \rightarrow \mathbb{R}$ is the reward function (bounded by $R_{\max}$), S_0 is the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi : S \rightarrow A$ maps states to actions. The state values of a policy π are defined as $V^\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0, \pi], \forall s \in S$. The optimal values are denoted by $V^*(s) = \max_\pi V^\pi(s)$.

REVEALE addresses settings in which the reward function R is not known a priori and must be inferred offline from a dataset before policy optimization. The dataset includes a limited number of pairwise rankings of sub-optimal trajectories, $\mathcal{D} = \{(\tau_1^1 \succ \tau_1^2), \dots, (\tau_n^1 \succ \tau_n^2)\}$, similar to (Brown et al. 2019). Here, τ_i^1 and τ_i^2 denote two different trajectories, where $\tau_i^1 \succ \tau_i^2$ indicates that τ_i^1 is preferred over τ_i^2. The learned reward is later used to solve different instances of the domain. The quality of the retrieved reward depends on the composition of the dataset, which, in turn, depends on the source of the dataset. Increasing the size of the training data does not guarantee learning the intended reward, as the additional trajectories generated from the same source may not provide novel information critical for learning an aligned reward. Furthermore, it is practically infeasible to foresee and construct a dataset that contains information pertaining to novel states that the agent will encounter when deployed.

REVEALE does not make any assumptions about the composition or the source of the trajectory dataset and utilizes explanations and a critic for evaluation of explanations. The critic is assumed to be an entity capable of reasoning about reward estimation (e.g., identifying a feature that makes a state good or bad). The critic is not required to be aware of all possible novel scenarios missing from the dataset a priori. Instead, REVEALE exposes wrong reward estimation in novel states using explanations of states that appear in the dataset. An alternative to REVEALE would be to train and deploy the agent using the unverified reward function and collect additional data about undesirable behavior. However, in real-world scenarios, this approach can be unsafe, prohibitively costly, and time-consuming.

Method

REVEALE uses explanations to verify and improve reward alignment. Explaining the learned model reveals not only what the agent knows but also potential errors in its reasoning. Initially, the posterior over the reward function is calculated from the trajectory dataset $\mathcal{D}$ using a Bayesian IRL method (Brown et al. 2020). Specifically, REVEALE uses Bayes' rule to calculate $\mathcal{P}(R|\mathcal{D}) \propto \mathcal{P}(\mathcal{D}|R)\mathcal{P}(R)$. Then REVEALE uses the Bradley-Terry model to define $\mathcal{P}(\mathcal{D}|R)$:

$$\mathcal{P}(\mathcal{D}|R) = \prod_{(\tau_i^1 \succ \tau_i^2) \in \mathcal{D}} \frac{e^{\beta R(\tau_i^1)}}{e^{\beta R(\tau_i^1)} + e^{\beta R(\tau_i^2)}} \tag{1}$$

where $R(\tau) = \sum_{s \in \tau} R(s)$ and $\beta \in [0, \infty)$. In the verification phase, the critic verifies the maximum a posteriori (MAP) reward function using verification tests in the form of queries to the agent. The agent responds by explaining its reward, and the critic signals whether the explanation passes the verification test. If it fails, the agent presents additional explanations from an alternative sample of the current posterior. The critic provides feedback by selecting the explanation that most closely matches the correct reasoning. This is followed by the improvement phase, in which the agent updates its posterior based on the additional feedback.

The input to REVEALE consists of a set of ranked trajectories $\mathcal{D}$ and the verification test states S_V. The test states can be selected randomly from $\mathcal{D}$ or by a human critic, who possesses a broader scope of knowledge and can identify critical states that affect performance. The algorithm begins by initializing an empty set of feedback $\mathcal{C}$ and retrieves the initial reward function R_m using Equation 1. Then the algorithm alternates between the verification and improvement phases until a reward function is found that passes all the verification tests. Figure 2 summarizes this process.

Verification phase In this phase, for each verification test state $s_V \in S_V$, the reward value $R_m(s_V)$ and the corresponding explanation $\mathcal{X}_{R_m}(s_V)$ are shown to the critic for approval. If approved, then $\mathcal{X}_{R_m}(s_V)$ is added to $\mathcal{C}$ as an Oracle explanation. If disapproved, additional feedback is requested from the critic. When possible, the critic can provide the correct explanation, i.e., $\mathcal{X}_O(s_V)$. Otherwise, the agent generates an alternative explanation $\mathcal{X}_{R'}(s_V)$ and collects the critic's ranking over $\mathcal{X}_{R_m}(s_V)$ and $\mathcal{X}_{R'}(s_V)$. Here, R' is a different reward function sampled from the posterior. Finally, all the feedback is added to $\mathcal{C}$. Note that if the critic fails to distinguish between the two explanations, additional alternative explanations can be generated to help the critic. If the agent does not pass all the tests, the algorithm proceeds to the improvement phase.

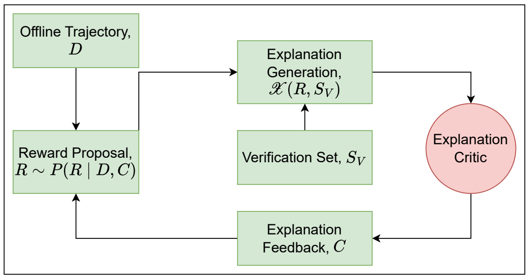
Offline Trajectory, D
Explanation Generation, $\mathscr{X}(R, S_V)$
Reward Proposal, $R \sim P(R \mid D, C)$
Verification Set, S_V
Explanation Critic
Explanation Feedback, C

Figure 2: Overview of the REVEALE Framework

Improvement phase In the improvement phase, a new posterior distribution over the reward function is calculated by combining $\mathcal{D}$ and $\mathcal{C}$ using Equation 2,

$$P(R|\mathcal{D}, \mathcal{C}) \propto P(\mathcal{C}|\mathcal{D}, R)P(\mathcal{D}|R)P(R). \tag{2}$$

Since explanations only depend on the reward function, $P(\mathcal{C}|\mathcal{D}, R) = P(\mathcal{C}|R)$. Based on this posterior distribution, a new MAP reward R_m is calculated. Then the algorithm goes back to the verification phase. Thus, our algorithm iteratively guides the reward alignment using explanations.

We now discuss how REVEALE defines $P(\mathcal{C}|R)$ for linear reward models[1], which are described by a linear weighted combination of features describing the state, $R(s) = \mathbf{w}^T \phi(s)$, $\mathbf{w} \in \mathbb{R}^n$.

REVEALE considers three forms of feature-attribution based explanations, namely LIME (LM), Gradient-as-Explanation (GaE), and saliency maps (SM). GaE is defined as the gradient of the reward function with respect to the input state features, i.e., $\partial R(s)/\partial \phi(s)$, and SM as $|\partial R(s)/\partial \phi(s)|$. Here, $|\cdot|$ indicates the absolute value. LIME approximates the local behavior of a complex model by fitting a simpler, interpretable model to the predictions in the vicinity of the input instance, however, it yields the same formula in the linear case as GaE.

Now, given an oracle feedback explanation $\mathcal{X}_O(s_i)$ for state s_i and a distance measurement $D(\cdot)$ (e.g., L2, cosine similarity), the learned reward function R must satisfy the following constraint:

$$D(\mathcal{X}_O(s_i), \mathcal{X}_R(s_i)) = 0, \tag{3}$$

so that the learned reward function produces the same explanation as the Oracle. Similarly, given the ranking over two

[1] In this scenario REVEALE implicitly assumes that the intended reward function can be represented with a vector in $\mathbb{R}^n$.

explanations $\mathcal{X}_{R_1}(s_i) \succ \mathcal{X}_{R_2}(s_i)$, the learned reward function R must satisfy the following constraint:

$$D(\mathcal{X}_{R_1}(s_i), \mathcal{X}_R(s_i)) < D(\mathcal{X}_{R_2}(s_i), \mathcal{X}_R(s_i)). \quad (4)$$

This constraint causes the explanation generated by the learned reward function to look more similar to the higher-ranked explanation. We henceforth use $\mathcal{C}$ to denote the set of all such constraints constructed from Equations 3-4 using the feedback. Since an aligned reward function will satisfy all the constraints, we can define $P(\mathcal{C}|R)$ as:

$$P(\mathcal{C}|R) = \frac{1}{Z}\mathbb{I}(\mathcal{C}, R), \quad (5)$$

with, $\mathbb{I}(\mathcal{C}, R) = 1$ when R satisfies all the constraints in $\mathcal{C}$ and 0 otherwise. When using $P(\mathcal{C}|R)$ from Equation 5 in Equation 2, the posterior probability of all the reward functions that do not produce the correct explanations is evaluated to 0 in the improvement phase. The posterior probability of all the reward functions that do produce correct explanations remains proportional to the original probability derived by BREX (Brown et al. 2020). Intuitively, the key role of REVEALE is to eliminate deceptive reward functions—those that accurately estimate rewards for the training dataset and therefore are selected by BREX, but would fail to produce correct estimates in novel situations. It is worth noting that REVEALE also provides a gradient-based solution for non-linear R by converting this constraint-satisfying 0,1 distribution to a softer constraint optimization distribution. We refer to Mahmud, Saisubramanian, and Zilberstein (2023) for more details.

Findings and Discussion

In (Mahmud, Saisubramanian, and Zilberstein 2023), the authors provide several theoretical insights for the linear reward case. It is shown that REVEALE can correctly return a solution from the solution set, $R \in \Delta(\mathcal{D}) \cap \Delta(\mathcal{X}^{S_V})$. The explanation-consistent reward set, denoted $\Delta(\mathcal{X}^{S_V})$, is a set of reward functions whose corresponding explanations are approved by the critic:

$$\Delta(\mathcal{X}^{S_V}) = \{R \in \mathcal{R} \mid \mathcal{F}_A(\mathcal{X}_R(s_V)) = 1, \forall s_V \in S_V\}. \quad (6)$$

The data-consistent reward set, denoted $\Delta(\mathcal{D})$, is a set of reward functions under which the higher ranked trajectories have a higher reward than the lower ranked trajectories:

$$\Delta(\mathcal{D}) = \{R \in \mathcal{R} \mid R(\tau_i^1) > R(\tau_i^2), \forall(\tau_i^1 \succ \tau_i^2) \in \mathcal{D}\}. \quad (7)$$

It is also shown that a single Oracle-generated GaE explanation feedback is sufficient to produce the optimal intended reward. Proposition 3 indicates that to reduce reward function ambiguity by $x\%$, it suffices to have ranked feedback over $k = \log_2(1/(1 - x/100))$ randomly generated GaE explanation pairs. The analysis also implies that GaE can be more effective than SM in reducing reward ambiguity under certain conditions.

The effectiveness of learning aligned linear and non-linear rewards with REVEALE was evaluated using three explanation generation techniques: gradient as explanations, LIME, and saliency map. To test this, five proof-of-concept domains were used, with training data generated by sub-optimally solving a set of training instances, and the learned reward evaluated on test instances that differ in start state distribution and risky region locations. Reward learning is challenging due to the limited state coverage in trajectories and spurious feature correlations. Metrics for evaluation include accuracy in predicting trajectory ranking in test instances, reward estimation quality in unseen states, and average reward achieved by executing a policy computed using the learned reward in test instances. The performance of our approach is compared with the true reward function policy (Optimal) and recent IRL algorithms: BREX (for linear rewards) and TREX (for non-linear rewards).

The results show that REVEALE significantly improves prediction accuracy in novel states by utilizing explanation feedback. While traditional IRL methods like REX often suffer from spurious correlations and inaccurate reward estimation in test scenarios, explanation-guided alignment in REVEALE provides more accurate and safer reward functions. The accuracy of ranking prediction improves with GaE explanations, especially when feedback is incorporated. In terms of avoiding risky states, GaE and LIME-based methods produce safer trajectories in most domains, with REVEALE outperforming REX and other methods in avoiding penalties from risky regions. Overall, our results demonstrate that REVEALE, with explanation-guided feedback, can learn reward functions that align better with true rewards, resulting in safer and more effective policies.

Conclusion

The verification and validation of AI systems, particularly those that are learning-enabled, present significant challenges. The proposed framework leverages *explanations for value alignment* and offers several distinct advantages. By utilizing a critic to assess the agent's explanations and iteratively refining the reward functions based on feedback, the approach ensures better alignment of AI behavior with human values and objectives.

The REVEALE case study underscores the critical role of explanations in identifying and correcting misaligned rewards, thereby improving the safety and reliability of AI systems in novel situations. The case study demonstrates the practical application of this framework, showcasing its effectiveness in various domains through empirical evaluations. Explanation-guided reward alignment not only enhances the transparency and trustworthiness of AI systems but also minimizes the need for extensive trajectory data, which is often impractical or risky to obtain.

In future work, we plan to explore new types of explanations, particularly explanations in natural language that are more comprehensible and intuitive for users. By leveraging the significant advancements in natural language processing through large language models (LLMs), we aim to enhance value alignment using language-based abstraction. Additionally, we intend to develop mechanisms to improve the sample efficiency of verification tests by incorporating more informative examples. Lastly, we seek to create methods that offer formal guarantees for explanation-guided verification, accounting for different assumptions regarding the accuracy of the critic.

Acknowledgments

This work was supported in part by the National Science Foundation grants 2205153, 2326054, 2321786, and 2416459.

References

Amir, O.; Doshi-Velez, F.; and Sarne, D. 2019. Summarizing agent strategies. *Autonomous Agents and Multi-Agent Systems*, 33: 628–644.

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5): 469–483.

Basich, C.; Svegliato, J.; Wray, K. H.; Witwicki, S.; Biswas, J.; and Zilberstein, S. 2023. Competence-Aware Systems. *Artificial Intelligence*, 103844.

Brown, D. S.; Goo, W.; Prabhat, N.; and Niekum, S. 2019. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *Proceedings of the 36th International Conference on Machine Learning*, 783–792.

Brown, D. S.; Niekum, S.; Coleman, R.; and Srinivasan, R. 2020. Safe imitation learning via fast Bayesian reward inference from preferences. In *Proceedings of the 37th International Conference on Machine Learning*, 1165–1177.

Brown, D. S.; Schneider, J. J.; and Niekum, S. 2021. Value alignment verification. In *Proceedings of the 38th International Conference on Machine Learning*, 1105–1115.

Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. *arXiv preprint arXiv:1701.08317*.

Dietterich, T. G. 2017. Steps toward robust artificial intelligence. *AI Magazine*, 38(3): 3–24.

Hadfield-Menell, D.; Milli, S.; Abbeel, P.; Russell, S. J.; and Dragan, A. 2017. Inverse reward design. In *Advances in Neural Information Processing Systems*.

Huang, S. H.; Bhatia, K.; Abbeel, P.; and Dragan, A. D. 2018. Establishing appropriate trust via critical states. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3929–3936.

Mahmud, S.; Nashed, S. B.; Goldman, C. V.; and Zilberstein, S. 2023. Estimating Causal Responsibility for Explaining Autonomous Behavior. In Calvaresi, D., ed., *International Workshop on Explainable and Transparent AI and Multi-Agent Systems (EXTRAAMAS)*, 78–94. Springer.

Mahmud, S.; Saisubramanian, S.; and Zilberstein, S. 2023. Explanation-Guided Reward Alignment. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, 473–482.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

Saisubramanian, S.; Kamar, E.; and Zilberstein, S. 2022. Avoiding negative side effects of autonomous systems in the open world. *Journal of Artificial Intelligence Research*, 74: 143–177.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *CoRR*, abs/1312.6034.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

Tabrez, A.; Agrawal, S.; and Hayes, B. 2020. Explanation-Based Reward Coaching to Improve Human Performance via Reinforcement Learning. In *Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction*, 249–257. ISBN 9781538685556.

Tayyub, J.; Sarmad, M.; and Schönborn, N. 2022. Explaining Deep Neural Networks for Point Clouds Using Gradient-based Visualisations. *arXiv preprint arXiv:2207.12984*.

Zilberstein, S. 2015. Building strong semi-autonomous systems. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 4088–4092.