1 System Details

1.1 System Owner

This may be the designer deploying the system, a larger agency or body, or some combination of the two. The entity completing the report should also be indicated.

1.2 Dates

The known or intended timespan over which this reward function & optimization is active.

1.3 Feedback & Communication

Contact information for the designer, team, or larger agency responsible for system deployment.

1.4 Other Resources

Where can users or stakeholders find more information about this system? Is this system based on one or more research papers?

2 Optimization Intent

2.1 Goal of Reinforcement

A statement of system scope and purpose, including the planning horizon and justification of a data-driven approach to policy design (e.g. the use of reinforcement learning or repeated retraining). This justification should contrast with alternative approaches, like static models and hand-designed policies. What is there to gain with the chosen approach?

2.2 Defined Performance Metrics

A list of "performance metrics" included explicitly in the reward signal, the criteria for why these metrics were chosen, and from where these criteria were drawn (e.g. government agencies, domain precedent, GitHub repositories, toy environments). Performance metrics that are used by the designer to tune the system, but not explicitly included in the reward signal should also be reported here.

2.3 Oversight Metrics

Are there any additional metrics not included in the reward signal but relevant for vendor or system oversight (e.g. performance differences across demographic groups)? Why aren't they part of the reward signal, and why must they be monitored?

2.4 Known Failure Modes

A description of any prior known instances of "reward hacking" or model misalignment in the domain at stake, and description of how the current system avoids this.

3 Institutional Interface

3.1 Deployment Agency

What other agency or controlling entity roles, if any, are intended to be subsumed by the system? How may these roles change following system deployment?

3.2 Stakeholders

What other interests are implicated in the design specification or system deployment, beyond the designer? What role will these interests play in subsequent report documentation? What other entities, if any, does the deployed system interface with whose interests are not intended to be in scope?

3.3 Explainability & Transparency

Does the system offer explanations of its decisions or actions? What is the purpose of these explanations? To what extent is the policy transparent, i.e. can decisions or actions be understood in terms of meaningful intermediate quantities?

3.4 Recourse

Can stakeholders or users contest the decisions or actions of the system? What processes, technical or otherwise, are in place to handle this?

4 Implementation

4.1 Reward Details

How was the reward function engineered? Is it based on a well-defined metric? Is it tuned to represent a specific behavior? Are multiple terms scaled to make one central loss, and how was the scaling decided?

4.2 Environment Details

Description of states, observations, and actions with reference to planning horizon and hypothesized dynamics/impact. What dynamics are brought into the scope of the optimization via feedback? Which dynamics are left external to the system, as drift? Have there been any observed gaps between conceptualization and resultant dynamics?

4.3 Measurement Details

How are the components of the reward and observations measured? Are measurement techniques consistent across time and data sources? Under what conditions are measurements valid and correct? What biases might arise during the measurement process?

4.4 Algorithmic Details

The key points on the specific algorithm(s) used for learning and planning. This includes the form of the policy (e.g. neural network, optimization problem), the class of learning algorithm (e.g. model-based RL, off-policy RL, repeated retraining), the form of any intermediate model (e.g. of the value function, dynamics function, reward function), technical infrastructure, and any other considerations necessary for implementing the system. Is the algorithm publicly documented and is code publicly available? Have different algorithms been used or tried to accomplish the same goal?

4.5 Data Flow

How is data collected, stored, and used for (re)training? How frequently are various components of the system retrained, and why was this frequency chosen? Could the data exhibit sampling bias, and is this accounted for in the learning algorithm? Is data reweighted, filtered, or discarded? Have data sources changed over time?

4.6 Limitations

Discussion and justification of modeling choices arising from computational, statistical, and measurement limitations. How might (or how have) improvements in computational power and data collection change(d) these considerations and impact(ed) system behavior?

4.7 Engineering Tricks

RL systems are known to be sensitive to implementation tricks that are key to performance. Are there any design elements that have a surprisingly strong impact on performance? For example, state-action normalization, hard-coded curricula, model-initialization, loss bounds, or more?

5 Evaluation

5.1 Evaluation Environment

How is the system evaluated (and if applicable, trained) prior to deployment (e.g. using simulation, static datasets, etc.)? Exhaustive details of the offline evaluation environment should be provided. For simulation, details should include description or external reference to the underlying model, ranges of parameters, etc. For evaluation on static datasets, considering referring to associated documentation (e.g. Datasheets *[1]).*

5.2 Offline Evaluations

Present and discuss the results of offline evaluation. For static evaluation, consider referring to associated documentation (e.g. Model Cards *[2]). If applicable, compare the behaviors arising from counterfactual specifications (e.g. of states, observations, actions).*

5.3 Evaluation Validity

To what extent is it reasonable to draw conclusions about the behavior of the deployed system based on presented offline evaluations? What is the current state of understanding of the online performance of the system? If the system has been deployed, were any unexpected behaviors observed?

5.4 Performance standards

What standards of performance and safety is the system required to meet? Where do these standards come from? How is the system verified to meet these standards?

6 System Maintenance

6.1 Reporting Cadence

The intended timeframe for revisiting the reward report. How was this decision reached and motivated?

6.2 Update Triggers

Specific events (projected or historic) significant enough to warrant revisiting this report, beyond the cadence outlined above. Example triggers include a defined stakeholder group empowered to demand a system audit, or a specific metric (either of performance or oversight) that falls outside a defined threshold of critical safety.

6.3 Changelog

Descriptions of updates and lessons learned from observing and maintaining the deployed system. This includes when the updates were made and what motivated them in light of previous reports. The changelog comprises the central difference between reward reports and other forms of machine learning documentation, as it directly reflects their intrinsically dynamic nature.

References

[1] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. Iii, and K. Crawford, “Datasheets for datasets,” *Communications of the ACM*, vol. 64, no. 12, pp. 86–92, 2021.

[2] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru, “Model cards for model reporting,” in *Proceedings of the conference on fairness, accountability, and transparency*, 2019, pp. 220–229.