# Reward Reports for Reinforcement Learning

**Thomas Krendl Gilbert** [1] **Sarah Dean** [2] **Nathan Lambert** [3] **Tom Zick** [4] **Aaron Snoswell** [5]

## Abstract

The desire to build good systems in the face of complex societal effects requires a dynamic approach towards equity and access. Recent approaches to machine learning (ML) documentation have demonstrated the promise of discursive frameworks for deliberation about these complexities. However, these developments have been grounded in a static ML paradigm, leaving the role of feedback and post-deployment performance unexamined. Meanwhile, recent work in reinforcement learning design has shown that the effects of optimization objectives on the resultant system behavior can be wide-ranging and unpredictable. In this paper we sketch a framework for documenting deployed learning systems, which we call *Reward Reports*.

## 1. Introduction

In fall 2021, the Wall Street Journal published the Facebook files, a series of articles that pieced together platform policy, flaws, and harms through leaked internal documents (Hagey & Horwitz, 2021). Among other topics, these articles tracked how changes to the newsfeed algorithm affected social interactions. Reporters confirmed what researchers suspected: Facebook's pivot to 'meaningful social interactions' increased negative content and divisiveness. Furthermore, both top content producers and Facebook executives were aware the algorithm generated this uptick, although the latter permitted it in favor of increased engagement. Even if Facebook were to make an explicit commitment to anticipate such effects, mitigating the feedback between interventions and harms is no small task. Facebook's platform policies balance hundreds of parameters which are updated multiple times a day, amounting to a never ending A/B test. As a former employee explained, the system had grown so complex that data scientists could not trace negative effects back to efforts to increase meaningful connection.

Multiple frameworks for documenting AI systems, datasets and models have emerged (Mitchell et al., 2019; Gebru et al., 2021; Richards et al., 2020). These aim to track sources of potential bias or harm, grounded primarily in a static machine learning (ML) paradigm. However, the effects of deployed AI systems are not static, and the dynamic impacts of successive system updates can subvert efforts both to manage downstream harms and to more evenly distribute benefits to vulnerable subpopulations. The presence of feedback and dynamics suggests unique risk vectors and requisite forms of documentation. Reinforcement learning (RL), a sub-field of ML that is able to solve complex sequential, open-ended problems, provides a dynamic lens that is broadly applicable to many algorithmic systems with repeated data-driven optimizations.

We propose *Reward Reports*, a new form of documentation that foregrounds the societal impacts of data-driven optimization systems, whether explicitly or implicitly construed as RL. Building on proposals to document datasets and models, we focus on reward functions: the objective that guides optimization decisions in feedback-laden systems. Reward Reports comprise questions that highlight the promised benefits and potential risks entailed in defining what is being optimized in an algorithmic system, and are intended as living documents that dissolve the distinction between *ex-ante* specification and *ex-post* evaluation. As a result, Reward Reports provide a framework for ongoing deliberation and accountability after a system is deployed, ensuring that desired properties persist in the system's behavior over time.

## 2. Related Work

**Documentation** There are a number of existing proposals for AI system documentation. Documenting data, regardless of resulting systems, is a well explored avenue (Barclay et al., 2019; Afzal et al., 2021; Hutchinson et al., 2021; Denton et al., 2021; Gebru et al., 2021). However, RL and deployed ML systems also generate and eventually transform data, so existing data documentation efforts

[1]Digital Life Initiative, Cornell Tech [2]Department of Computer Science, Cornell University [3]Hugging Face [4]Harvard Law School [5]Centre for Automated Decision-Making and Society, Queensland University of Technology. Correspondence to: Thomas Krendl Gilbert <tg299@cornell.edu>, Sarah Dean <sdean@cornell.edu>.

*ICML 2022 Workshop on Responsible Decision Making in Dynamic Environments*, Baltimore, Maryland, USA, 2022. 

are insufficient to reveal the risks and failures of dynamic datasets and feedback driven systems. There are also proposals for model (Mitchell et al., 2019), domain (Ramírez et al., 2020; Kühl et al., 2021) or outcome specific (Sokol & Flach, 2020) forms of ML documentation. Reward Reports might be a useful supplement to these approaches by focusing on the specification and effects of models. Our work has similarities with previous proposals for AI Ethics Sheets (Mohammad, 2021), Fact Sheets (Arnold et al., 2019; Richards et al., 2020), or Scorecards (Blasch et al., 2021), but uniquely focuses on prompting deliberation about the feedback-driven risks inherent to dynamic systems. Reward Reports are closely related to Algorithmic Impact Assessments (AIA), which offer a framework for evaluating risks before an AI system is developed or acquired (Reisman et al., 2018; Selbst & Barocas, 2018). These frameworks presume an agency-vendor relationship, and focus narrowly on the procurement of automated decision systems. Reward Reports are intended to supersede these *ex-ante* concerns, engaging instead with the necessarily circuitous process of refining the specifications of a feedback system.

**Societal Risks of RL** The RL research community has begun to reflect on the unique risks and challenges that may be posed by RL systems (Whittlestone et al., 2021; Gilbert, 2021; Wen et al., 2021; Carroll et al., 2021; Evans & Kasirzadeh, 2021; Zhan et al., 2021). Recent general audience books have echoed these concerns (Russell, 2019; Christian, 2020). While these efforts have begun to capture the unique stakes in deploying RL systems, there is no consensus on how to chart associated risks. We intend Reward Reports as an instrument of deliberation and accountability.

**AI Governance** As reflected in the growing number of proposed AI governance frameworks (Gasser & Almeida, 2017; Lee et al., 2019; Yeung et al., 2019; Cihon, 2019; Wirtz et al., 2020; Reddy et al., 2020), ML and adjacent communities have increasingly acknowledged sociotechnical risks and the need for novel harm mitigation strategies (Selbst et al., 2019; Dean et al., 2021).

## 3. Background

### 3.1. Action, Objective, Adaptation

Learned predictive models are the means to some end. It is the decisions made, or *actions* taken, that determine the extent to which a model is successful. For example, congested suburban roadways in the community of Los Gatos, CA, USA are caused directly by the actions of drivers, and indirectly by the actions of routing algorithms that predict a poorly-scaling short-cut path (Peterson, 2018).

Action occurs not only on the basis of predictions, but also towards some *objective*. The definition of objective is crucial to the resulting behavior. Identical traffic models will result in different routing suggestions depending on whether the algorithm is optimizing for arrival time or fuel consumption (NREL, 2021).

Finally, these optimization systems are often updated based on additional data collected during their operation, making them *adaptive*. By accounting for the dependence between past decisions, observed data, and current models, systems in effect react to dynamic environments and improve performance over time. For example, when observed music listening patterns are used as additional data in preference models, music recommendation algorithms can adapt to an individual's evolving tastes (Dhahri et al., 2018).

While action, objective, and adaptation are important for ensuring that systems work as intended, they are not captured in frameworks that document static elements of ML models.

### 3.2. Reinforcement Learning

The reinforcement learning (RL) framework succinctly encompasses action, objective, and adaptation. RL algorithms take actions, are motivated by a reward signal which encodes the objective, and adapt based on the feedback from this interaction. While the goal of supervised learning (SL) procedures is to use data to generate a model that makes accurate predictions, the goal of RL algorithms is to interact with an environment to generate a policy that achieves high reward. However, once SL models are deployed towards some goal and updated with new data, the concerns highlighted by the RL framework become relevant. In this sense, ML deployments can be understood through the lens of RL.

Reinforcement learning is a framework for optimizing a system *via* trial and error. In common terminology, an *agent* executes *actions* $\vec{a}_t \in \mathcal{A}$ in an *environment*. In response, the agent receives a scalar *reward* $r_t \in \mathbb{R}$ and makes an *observation* $\vec{o}_t \in \mathcal{O}$ of the environment. Actions are made on the basis of these observations according to a *policy* $\pi : \mathcal{H} \to \Delta(\mathcal{A})$, where $\mathcal{H} = \mathcal{O} \times \cdots \times \mathcal{O}$ represents the history of observations and $\Delta(\mathcal{A})$ represents a probability distribution over the action space. The goal of a reinforcement learning agent is to find a policy that maximizes the cumulative reward over some time horizon: $\sum_{t=0}^{H} \gamma^t r_t$ where the discount factor $\gamma \in (0, 1]$ weighs current rewards versus future potential rewards. This paradigm captures many problems of interest, from choosing advertisements that are most likely to result in a click (Liu & Li, 2021; Langford & Zhang, 2007) to determining the best dosing schedule for a patient (Shortreed et al., 2011).

A key element of RL is the effect of actions on the future behavior of the environment. This dependence is often modeled as a Markov Decision Process (MDP) (Bellman, 1957). In the MDP setting, the *state* $\vec{s}_t$ describes the status

of the environment. The key assumption, called memorylessness, is that the current state and action are sufficient for predicting the future state. These *transition probabilities* from one state to the next are also referred to as the *system dynamics*. Furthermore, the reward is determined by the state, so that $r_t = r(\vec{s}_t, \vec{a}_t)$ for some reward function mapping from the current environment to a scalar representation of desirability. Under these assumptions, it is optimal to consider policies that depend only on the current state, $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. Often, RL algorithms assume that the state is observed directly $\vec{o}_t = \vec{s}_t$ (or similarly, that it can be constructed in a straightforward manner *e.g.* though history truncation $\vec{s}_t = [o_{t-h}, \ldots, o_t]$), and the policy is typically parametric, with a parameter vector denoted $\theta$. In the MDP setting there are a rich set of tools for understanding and optimizing performance, such as value functions *via* dynamic programming, that can be directly applied when the full transition model is known.

Though such MDP dynamics are understood to determine the evolution of the system, for RL problems they are not necessarily known *a priori* to designers. Instead, RL algorithms seek to optimize the policy on the basis of interactions with the environment, using the reward signal and observations. There are a wide range of RL algorithms: some which optimize the policy directly (*e.g.* by learning the weights of a neural network based on observed performance), and others which first estimate intermediate quantities (*e.g.* value, transition, or reward function) and then transform these into a policy. Some algorithms generate policies offline from existing datasets, while others solely use online interactions. Any RL algorithms with an online component must contend with the exploration/exploitation trade-off: choosing between actions likely to be high reward, and those likely to be informative (Sutton & Barto, 2018).

The RL framing is general, so other machine learning paradigms can be viewed as special cases of it. For example, supervised learning can be viewed as the optimization of a classification or regression policy where the rewards are defined by accuracy and the time horizon is equal to one. Online learning situates supervised learning systems in a sequentially evolving environment (Shalev-Shwartz et al., 2011), while the study of bandit problems reduces RL to the static regime where actions do not affect the environment (Berry & Fristedt, 1985).

## 4. Motivation

The RL lens is useful not only because ML deployments often operate in dynamical environments. It is also that there is *feedback* between the environment and the deployment. In this section, we first review three levels of feedback that characterize RL systems: *control* feedback from state to action, *behavior* feedback from the data to the policy, and *exogenous* feedback from the target environment to adjacent entities (Gilbert et al., 2022) (see Fig. 1).

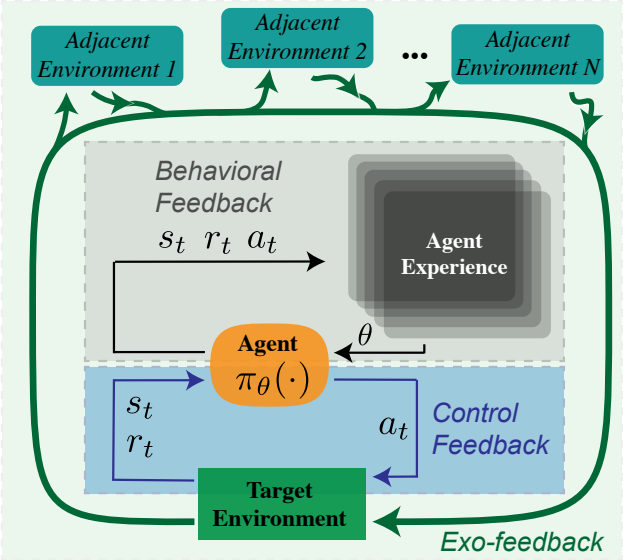


*Figure 1.* A diagram depicting the interactions of control, behavioral, and exogenous feedback in a RL system.

### 4.1. A Taxonomy of Feedback

*Control feedback* maps observations or states to actions. A very simple example is a thermostat which decides whether or not to turn on a furnace based on temperature sensors. These decisions are made many times per second, and allow an HVAC system to maintain comfortable indoor temperature under varying weather conditions. Such systems are often called "automatic feedback control," hence the term control feedback. "Intelligent" behaviors arise because actions are constantly adjusted on the basis of observations, but the rules that control the behavior remain the same.

*Behavior feedback* maps data to the learned policy. This form of feedback occurs when RL systems automatically adapt their policy based on reward. Questions of reaction become questions of trial-and-error evaluation: "At what temperature should the furnace turn on?" evolves into "What keeps the operating temperature as close as possible to the target?" The ability to learn from experience is part of what makes RL systems seem so powerful, and it makes them applicable to domains that are difficult to otherwise model. For example, it would be challenging to hand-design a policy for recommending music to a listener, but data-driven approaches make this task tractable.

*Exogenous feedback* occurs when the application domain itself shifts in response to the deployed system. These shifts could be due to political or economic conditions that are outside the system's purview. For example, smart thermostats

may enable finer-grained control over household heating, changing what a comfortable home amounts to or changing the electric loading of buildings, towns, and regions. Traffic driven by recommendation systems might incentivize creators to create attention-grabbing content, turning to strategies like outrage and conspiracy (Munger & Phillips, 2022). In principle, if such dynamics could be predicted by an RL agent, they could be brought under the purview of behavior or control feedback. But in practice, it is not clear that this is possible—the observations would need to be extremely rich and the planning horizon extremely long. Exogenous feedback highlights the potential of externalized risks.

### 4.2. Risks and Documentation

For many systems, reward design–the choice of how and what to optimize–amounts to a political decision about how different types of feedback may rewire the domain and pose risks to various stakeholders. As it is often impossible to fold all of the domain dynamics within a controllable planning horizon and precise reward function, exo-feedback is in practice unavoidable. Furthermore, it is unrealistic to articulate all possible specifications *a priori*. A single specification may not only induce exo-feedback, but also necessarily implicates forms of control and behavioral feedback.

The risks of feedback can at least be approached and evaluated through documentation. This calls for legible and periodic mechanisms for auditing RL systems pre- and post-deployment. It is these reviews that must decide whether or how the optimized behaviors align with the application domain, in correspondence with resultant risks and possible harms. Given the dynamic nature of these effects, the corresponding document must be dynamic as well: updated and revisited over time to map the evolution of feedback between the system and the domain in which it is deployed.

## 5. Reward Report Components

We propose *Reward Reports*, a structured series of design inquiries for automated decision systems (see Fig. 2). Including but not limited to the use of reinforcement learning, Reward Reports are intended to engage practitioners by revisiting design questions over time, drawing reference to previous reports and looking forward to future ones. The changelog component of a Reward Report becomes an interface for stakeholders, users, and engineers to oversee and evaluate the documented system.

A Reward Report is composed of six sections, arranged to help the reporter understand and document the system. A Reward Report begins with ***system details*** (1) that contain the information context for deploying the model. From there, the report documents the ***optimization intent*** (2) which questions the goals of the system and why RL or ML may be a useful tool. The designer then documents how it can affect different stakeholders in the ***institutional interface*** (3). The next two sections contain technical details on the system ***implementation*** (4) and ***evaluation*** (5). The report concludes with plans for ***system maintenance*** (6) as additional system dynamics are uncovered.

The appendix includes a more thorough description of these components, and three example Reward Reports from varied domains.

**Reward Report Contents**

- **System Details**: Basic system information.
  - Person or organization developing the system
  - Deployment dates
  - Contact
- **Optimization Intent**: The goals of the system and how reinforcement manifests.
  - Goal of reinforcement
  - Performance metrics
  - Oversight metrics
  - Failure modes
- **Institutional Interface**: The interconnections of the automated system with society.
  - Involved agencies
  - Stakeholders
  - Computation footprint
  - Explainability
  - Recourse
- **Implementation**: The low-level engineering details of the ML system.
  - Reward, algorithmic, and environment details
  - Measurement details
  - Data flow
  - Limitations
  - Engineering artifacts
- **Evaluation**: Specific audits on system performance.
  - Evaluation environment
  - Offline evaluations
  - Evaluation validity
  - Performance standards
- **System Maintenance**: Plans for long-term verification of behavior.
  - Reporting cadence
  - Update triggers
  - Changelog

*Figure 2.* Summary of reward report sections.

## 6. Conclusion

The scale and complexity of contemporary optimization pipelines raise unique concerns not addressed by static documentation. Reward Reports fill this gap, providing a framework for iterative deliberation over the time-evolution of a system and its feedback channels. Responsibility is a dynamic problem, and needs to be deliberated about as such.

Reward Reports enact forms of documentation commensurate with the feedback-laden systems whose dynamics–not just models or data–are a critical object of concern.

## References


Afzal, S., Rajmohan, C., Kesarwani, M., Mehta, S., and Patel, H. Data readiness report. In *2021 IEEE International Conference on Smart Data Services (SMDS)*, pp. 42–51. IEEE, 2021.

Arnold, M., Bellamy, R. K., Hind, M., Houde, S., Mehta, S., Mojsilović, A., Nair, R., Ramamurthy, K. N., Olteanu, A., Piorkowski, D., et al. Factsheets: Increasing trust in ai services through supplier's declarations of conformity. *IBM Journal of Research and Development*, 63(4/5):6–1, 2019.

Barclay, I., Preece, A., Taylor, I., and Verma, D. Towards traceability in data ecosystems using a bill of materials model. *arXiv preprint arXiv:1904.04253*, 2019.

Bellman, R. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957.

Berry, D. A. and Fristedt, B. Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). *London: Chapman and Hall*, 5 (71-87):7–7, 1985.

Blasch, E., Sung, J., and Nguyen, T. Multisource ai scorecard table for system evaluation. *arXiv preprint arXiv:2102.03985*, 2021.

Carroll, M., Hadfield-Menell, D., Russell, S., and Dragan, A. *Estimating and Penalizing Preference Shift in Recommender Systems*, pp. 661–667. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450384582. URL https://doi.org/10.1145/3460231.3478849.

Christian, B. *The Alignment Problem: Machine Learning and Human Values*. WW Norton & Company, 2020.

Cihon, P. Standards for ai governance: international standards to enable global coordination in ai research & development. *Future of Humanity Institute. University of Oxford*, 2019.

Dean, S., Gilbert, T. K., Lambert, N., and Zick, T. Axes for sociotechnical inquiry in ai research. *IEEE Transactions on Technology and Society*, 2(2):62–70, 2021.

Denton, E., Hanna, A., Amironesei, R., Smart, A., and Nicole, H. On the genealogy of machine learning datasets: A critical history of imagenet. *Big Data & Society*, 8(2): 20539517211035955, 2021.

Dhahri, C., Matsumoto, K., and Hoashi, K. Mood-aware music recommendation via adaptive song embedding. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 135–138. IEEE, 2018.

Evans, C. and Kasirzadeh, A. User tampering in reinforcement learning recommender systems. *arXiv preprint arXiv:2109.04083*, 2021.

Gasser, U. and Almeida, V. A. A layered model for ai governance. *IEEE Internet Computing*, 21(6):58–62, 2017.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., and Crawford, K. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

Gilbert, T. K. Mapping the political economy of reinforcement learning systems: The case of autonomous vehicles. *Center for Long Term Cybersecurity Whitepaper Series*, 2021. URL https://simons.berkeley.edu/news/mapping-political-economy-reinforcement-learning-

Gilbert, T. K., Dean, T. K., Zick, T., and Lambert, N. Choices, risks, and reward reports: Charting public policy for reinforcement learning systems. *Center for Long Term Cybersecurity Whitepaper Series*, 2022.

Hagey, K. and Horwitz, J. Facebook Tried to Make Its Platform a Healthier Place. It Got Angrier Instead. https://www.wsj.com/articles/facebook-algorithm-change-zuckerberg-11631654215?mod=series_facebookfiles, 2021. [Online; accessed 2-January-2022].

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Hutchinson, B., Smart, A., Hanna, A., Denton, E., Greer, C., Kjartansson, O., Barnes, P., and Mitchell, M. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 560–575, 2021.

Krakovna, V., Uesato, J., Mikulik, V., Rahtz, M., Everitt, T., Kumar, R., Kenton, Z., Leike, J., and Legg, S. a. Specification gaming: the flip side of AI ingenuity. https://deepmind.com/blog/article/Specification-gaming-the-flip-side-of-AI-ingenuit 2020. [Online; accessed 16-January-2022].

Kreidieh, A. R., Wu, C., and Bayen, A. M. Dissipating stop-and-go waves in closed and open networks via deep

reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1475–1480, 2018. doi: 10.1109/ITSC.2018.8569485.

Kühl, N., Hirt, R., Baier, L., Schmitz, B., and Satzger, G. How to conduct rigorous supervised machine learning in information systems research: The supervised machine learning report card. *Communications of the Association for Information Systems*, 48(1):46, 2021.

Langford, J. and Zhang, T. Epoch-greedy algorithm for multi-armed bandits with side information. *Advances in Neural Information Processing Systems (NIPS 2007)*, 20: 1, 2007.

Lee, M. K., Kusbit, D., Kahng, A., Kim, J. T., Yuan, X., Chan, A., See, D., Noothigattu, R., Lee, S., Psomas, A., et al. Webuildai: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–35, 2019.

Liu, Y. and Li, L. A map of bandits for e-commerce. In *Workshop on the Multi-Armed Bandits and Reinforcement Learning (MARBLE)*, 2021.

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

Mohammad, S. M. Ethics sheets for ai tasks. *arXiv preprint arXiv:2107.01183*, 2021.

Munger, K. and Phillips, J. Right-wing youtube: a supply and demand perspective. *The International Journal of Press/Politics*, 27(1):186–219, 2022.

NREL. Google Taps NREL Expertise To Incorporate Energy Optimization into Google Maps Route Guidance . https://www.nrel.gov/news/program/2021/google-taps-nrel-expertise-to-incorporate-energy-optimization-into-google-maps-route-guidance.html, April 2021. [Online; accessed 2-January-2022].

Peterson, J. Google apps causing gridlock in downtown Los Gatos. https://www.mercurynews.com/2018/06/01/google-apps-causing-gridlock-for-downtown-los-gatos/, 2018. [Online; accessed 2-January-2022].

Ramírez, J., Baez, M., Casati, F., Cernuzzi, L., and Benatallah, B. Drec: towards a datasheet for reporting experiments in crowdsourcing. In *Conference Companion Publication of the 2020 on Computer Supported Cooperative Work and Social Computing*, pp. 377–382, 2020.

Reddy, S., Allan, S., Coghlan, S., and Cooper, P. A governance model for the application of ai in health care. *Journal of the American Medical Informatics Association*, 27(3):491–497, 2020.

Reisman, D., Schultz, J., Crawford, K., and Whittaker, M. Algorithmic impact assessments: A practical framework for public agency accountability. *AI Now Institute*, pp. 1–22, 2018.

Richards, J., Piorkowski, D., Hind, M., Houde, S., and Mojsilović, A. A methodology for creating ai factsheets. *arXiv preprint arXiv:2006.13796*, 2020.

Russell, S. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

Selbst, A. D. and Barocas, S. The intuitive appeal of explainable machines. *Fordham Law Review*, 87:1085, 2018.

Selbst, A. D., Boyd, D., Friedler, S. A., Venkatasubramanian, S., and Vertesi, J. Fairness and abstraction in sociotechnical systems. In *Proceedings of the conference on fairness, accountability, and transparency*, pp. 59–68, 2019.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

Shortreed, S. M., Laber, E., Lizotte, D. J., Stroup, T. S., Pineau, J., and Murphy, S. A. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Machine learning*, 84(1-2):109–136, 2011.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Sokol, K. and Flach, P. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 56–67, 2020.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Wen, M., Bastani, O., and Topcu, U. Algorithms for fairness in sequential decision making. In *International Conference on Artificial Intelligence and Statistics*, pp. 1144–1152. PMLR, 2021.

Whittlestone, J., Arulkumaran, K., and Crosby, M. The societal implications of deep reinforcement learning. *Journal of Artificial Intelligence Research*, 70:1003–1030, 2021.

Wirtz, B. W., Weyerer, J. C., and Sturm, B. J. The dark sides of artificial intelligence: An integrated ai governance framework for public administration. *International Journal of Public Administration*, 43(9):818–829, 2020.

Wu, C., Kreidieh, A. R., Parvate, K., Vinitsky, E., and Bayen, A. M. Flow: A modular learning framework for mixed autonomy traffic. *IEEE Transactions on Robotics*, pp. 1–17, 2021. doi: 10.1109/TRO.2021.3087314.

Yeung, K., Howes, A., and Pogrebna, G. Ai governance by human rights-centred design, deliberation and oversight: An end to ethics washing. *The Oxford Handbook of AI Ethics, Oxford University Press (2019)*, 2019.

Zhan, R., Christakopoulou, K., Le, Y., Ooi, J., Mladenov, M., Beutel, A., Boutilier, C., Chi, E., and Chen, M. Towards content provider aware recommender systems: A simulation study on the interplay between user and provider utilities. In *Proceedings of the Web Conference 2021*, pp. 3872–3883, 2021.

# A. Reward Report Components

## A.1. System Details

This section collects basic information a user or stakeholder may need in reference to the automated decision system.

1. **Person or organization deploying the system**: 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.
2. **Reward date(s)**: The known or intended timespan over which this reward function & optimization is active.
3. **Feedback & communication**: Contact information for the designer, team, or larger agency responsible for system deployment.
4. **Other resources**: Where can users or stakeholders find more information about this system? Is this system based on one or more research papers?

## A.2. Optimization Intent

This section addresses basic questions about the intent of the reward function and optimization problem. Designers first document the intent of a particular solution, translating the system's quantitative objective into a qualitative description. In later sections, they have the opportunity to further reflect on how implementation details aid in, or diminish the broader goal. Stakeholders and users can employ this section to understand if the intent of the system matches with the effects they observe or experience.

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. **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.
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?
4. **Known failure modes**: A description of any prior known instances of "reward hacking" or model misalignment in the domain at stake (Krakovna et al., 2020), and description of how the current system avoids this.

## A.3. Institutional Interface

This section documents the intended (and in subsequent reports, observed) relationship between the system and the broader context in which it is deployed. While necessarily piecemeal, the explicit documentation of this interface will allow designers to reflect on and revisit the system assumptions over time. These reflections may bring novel interests or agencies into scope and allow for organizing the emergent interests of stakeholders and users where necessary.

1. **Deployment Agency**: What other agency or controlling entity roles, if any, are intended to be subsumed by the RL system? How may these roles change following system deployment?
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. **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?
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?

### A.4. Implementation

Given the sensitivity of reinforcement learning systems, it is important to document specific implementation details of the system. Even small changes in implementation can result in substantial behavior shifts downstream, making such factors difficult to track when used at scale. Documenting these design decisions will both help prevent failures in specific applications and assist technical progress.

1. **Reward details**: How was the reward function engineered? *E.g.* 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?
2. **Environment details**: Description of states, observations, and actions with reference to planning horizon and hypothesized dynamics/impacts. 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?
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. **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?
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?
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?
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? E.g. state-action normalization, hard-coded curricula, model-initialization, loss bounds, or more?

### A.5. Evaluation

Assessing the potential behavior of a feedback system is important for anticipating its future performance and risks that may arise. This section records evaluations done by the designer before deploying the system and each time the reward report is revisited. This section allows stakeholders and users to hold designers accountable for the performance of the system once deployed. It is important to distinguish whether the evaluations are done in a simulation (*offline*) or deployed on real users (*online*) and if the evaluation procedure is on a fixed dataset (*static*) or evolves over time (*dynamic*).

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* (Gebru et al., 2021)).
2. **Offline evaluations**: Present and discuss the results of offline evaluation. For static evaluation, consider referring to associated documentation (*e.g. Model Cards* (Mitchell et al., 2019)). If applicable, compare the behaviors arising from counterfactual specifications (*e.g.* of states, observations, actions).
3. **Evaluation validity**: To what extent is it reasonable to draw conclusions about the behavior of the deployed system based on the available offline evaluations?
   How is the online performance of the system presently understood? If the system has been deployed, were any unexpected behaviors observed?
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?

### A.6. System Maintenance

This section documents plans for post-deployment oversight, including subsequent reviews of real-world implementation and how the monitoring of resultant dynamics is intended to (or has) shed light on *ex-ante* assumptions. These plans include any additional grounds for updating the report in case of sustained shifts in observations or metrics (*e.g.* the effects of exogenous changes on system behaviors). As such, this section must draw sustained reference to previous Reward Reports, including subsequent changes to the description, implementation, or evaluation, and what prompted these changes. While previous sections outline how the system learns from data, this section tracks how organizations learn to oversee the system. Its documentation is particularly important for defining *accountability* for the system itself, those who manage it, and those responsible for completing periodic reports.

1. **Reporting cadence**: The intended timeframe for revisiting the Reward Report. How was this decision reached and motivated?
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.
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 is the key difference between Reward Reports and other forms of machine learning documentation, as it successively reframes prior reports and reflects their intrinsically dynamic nature.

## B. Examples

We include three example Reward Reports for imagined and real automated decision-making systems. Our aim with these examples is to illustrate the breadth and scope of questions that a Reward Report could engage with, and to demonstrate how Reward Reports can apply to both explicit and implicit RL systems. Here, we briefly outline each included example, and refer the reader to the appendix for the actual example Reward Reports.

### B.1. Project Flow: An RL policy for dissipating stop-and-go traffic waves

Project Flow is an autonomous vehicle testbed that allows using deep reinforcement learning to control and optimize traffic across in roadway networks (Wu et al., 2021). Inspired by recent work with using Project Flow (Kreidieh et al., 2018), we sketch a hypothetical deployment of an RL policy designed for dissipating stop-and-go traffic waves at a freeway exit, including several iterations of the Reward Report documented in the accompanying changelog. The changelog shows various problems that arise with the resulting problem dynamics, including an expansion of the planning horizon, the addition of new oversight metrics, stakeholder complaints, and requisite institutional shifts to cope with changes to the specification and application domain.

### B.2. MovieLens: A dynamically updated movie recommender system

The purpose of MovieLens is to match users to personalized movie recommendations based on ratings of other movies previously entered by the user (Harper & Konstan, 2015). Unlike the other example systems we discuss, MovieLens is a static preference model generated through supervised learning. However, because of the system's age (initial release in 1997) and its repeated retraining, it can be interpreted as an RL system that is learning a ranking policy that must adapt to a changing environment. The changelog documents the actual historical updates to the model prompted by changes to the environment, including new interfaces, user-base size, optimization parameters, user-generated content, and major dataset publications. This example Reward Report is based on the history of the MovieLens project published in (Harper & Konstan, 2015).

### B.3. MuZero: DeepMind's general game-playing agent

The purpose of MuZero (and its preceding systems, AlphaGo and AlphaZero) is to improve state-of-the-art performance in the games of chess, Go, shogi, and a benchmark suite of Atari games (Silver et al., 2016). We provide a Reward Report that documents the evolution of the system through these successive stages of development, including changes in the design motivation and performance metrics, as well as more extensive use of reinforcement learning.

$$r(t) = \|v_{\text{des}}\| - \|v_{\text{des}} - v(t)\| - \alpha \sum_i \max\left[h_{\max} - h_i(t), 0\right]$$

Figure 1: The reward function for the system in question consists of three terms. The first term $v_{\text{des}}$ is a positive constant that rewards the agent for longer simulation episodes - discouraging vehicle collisions, which terminate simulation runs early. The second term penalizes the agent when the instantaneous overall system velocity $v(t)$ differs from the desired system velocity $v_{\text{des}}$. Finally, the third term sums over each subscribed Connected Autonomous Vehicle and adds a penalty whenever this vehicle is too close to the vehicle immediately in-front - a characteristic known to trigger stop-and-go traffic waves. More details are provided below in the section 'Defined Performance Metrics'.

# 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.*

This system was developed by the Project Flow core team members, with all deployment, infrastructure, and ongoing management taking managed by Caltrans.

## 1.2 Dates

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

The system discussed here was trained in simulation during 2020, using empirical hyper-parameters (such as inflow traffic rates) collected during 2019. The RL policy was deployed in the real world on a trial basis on the 1st of Jan, 2021, and is presently undergoing initial real-world evaluation and validation.

## 1.3 Feedback & Communication

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

Any correspondence should be directed to test@example.ca.gov.

## 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?*

More information about this specific system can be found in the paper [1], as well as in the associated project website.

General information about the project flow simulation environment can be found in [2] or on the project website and associated GitHub repository.

# 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?*

The system in question is designed to dissipate stop-and-go traffic waves caused by merging off the California State Route 24 (CA-24) freeway onto Telegraph Avenue in the North Oakland / South Berkeley metropolitan area.

This is achieved by the coordinated actions of any subscribed Connected Autonomous Vehicles (CAVs) operating along the freeway segment in question, acting to 'shepherd' non-autonomous vehicles into patterns of traffic which can locally buffer against stop-and-go traffic waves.

Eligible CAVs, when entering the freeway zone of interest, communicate over the 4G/5G cell network with the central controller hub to 'subscribe' to the traffic management policy, which then sends real-time recommendations to these vehicles about

lane selection and preferred acceleration/braking profiles.

The RL policy is trained using a discrete-time road network simulation, with simulation runs lasting 3600s (one hour), and individual steps of 0.2s, giving 1800 steps per full simulation episode. The simulated road network consists of an 800m stretch of the CA-24 freeway containing a single off-ramp merging lane. These temporal and spatial planning horizons were selected because they were deemed large enough to allow emergence of typical driving dynamics based on the average safe following distance between vehicles and driver reaction times along comparable freeway offramps, based on state and federal records of past traffic behavior.

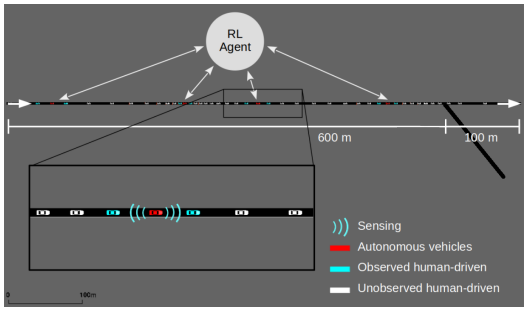


Figure 2: A central RL controller attempts to mitigate stop-and-go traffic waves caused by vehicles entering the freeway *via* on-ramps.

As of entry 0.3, it was found that the planning horizon for the system was too short. Following consultation with Caltrans, it was found that increasing the horizon from 500m to 800m would provide a significant increase in simulation performance without exhausting computational resources. Any future changes in computational capabilities will be documented here and compared in light of prior modeling choices and stakeholder commitments.

Simplistic microscopic traffic analysis models preclude the possibility of stable congestion patterns in open road topologies. However, as any driver can attest, these traffic patterns are ubiquitous on many road systems today. Instead, the presence of these traffic patterns in real-world networks is typically attributed to perturbations from bottleneck structures which can be difficult to capture in theoretical analyses (such as lane closures, road works, road debris, *etc*). [1] The ad-hoc nature of these perturbations means that modelling and planning for their occurrence within classical control frameworks may be difficult, motivating more flexible approaches such as Deep Reinforcement Learning.

RL may be indicated in this situation, compared to static supervised ML models, due to the fact that it inherently encompasses multiple types of feedback through the environment specification. For instance, in the case of CA-24, RL may help mitigate the observed phenomenon of excessive traffic on residential streets near highway intersections that is induced by apps like Google Maps and Waze. In the interest of recommending perceived shortcuts to individual human drivers, these apps have in fact been known to induce overload on smaller roadways, generating unnecessary stoppage and possible gridlock. In the case of Los Gatos (where this phenomenon has been previously recorded), the city's Parks and Public Works Director noted that "The apps are not able to respond fast enough to the overload they have created on the roadways" [3]. RL may make real-time monitoring and control of the CA-24 offramp possible, mitigating induced overload effects and stabilizing feedback between traffic behavior and road infrastructure.

## 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.*

The reward signal optimized by this system consists of three performance metrics, outlined in fig. 1. These terms are;

- $\|v_{\text{des}}\|$ - the desired system-level velocity in m/s. This is a positive constant reward to penalize prematurely terminated simulation rollouts caused by vehicle collisions. For the simulated experiments described here, $v_{\text{des}} = 25\text{m/s} = 90\text{kmph} \approx 55\text{mph}$.
- $-\|v_{\text{des}} - v(t)\|$ - the absolute difference between the desired system level velocity and the actual instantaneous system-level velocity in m/s. A non-zero difference incurs a cost for the RL agent.
- $-\alpha \sum_i \max\left[h_{\max} - h_i(t), 0\right]$ - this term sums over each Autonomous Vehicle in the purview

of the RL agent, and accrues a cost whenever that vehicle's instantaneous time headway (gap in seconds to the vehicle ahead) is too small (*i.e.* lower than $h_{\max}$). The sum of all headway costs is scaled by a gain factor $\alpha$. For the simulated experiments described here, $h_{\max} = 1$s and $\alpha = 0.1$.

### 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?*

Several other performance metrics are not included in the reward function, but are analysed for the purpose of evaluating the system performance:

- Absolute temporal vehicle density (or *throughput*) - the number of vehicles exiting the controlled region the road network, measured in vehicles/hr. A larger vehicle flow-through rate compared to baseline is seen as a positive effect (assumed to correlate with a decrease in stop-and-go traffic waves, and to indicate that the road network is functioning efficiently).
- Absolute spatial vehicle density (or *network congestion*) - the number of vehicles within a fixed region of the road network, measured in vehicles/m. A larger number of vehicles present on the roadway is seen as a negative effect, indicating increased likelihood of stoppage.
- The average velocity of vehicles in the system. Higher vehicle velocities are seen as a positive effect.
- The average time vehicles spend within a given region of the system. Lower average time is seen as a positive effect.
- The maximum time any vehicle spent within a given region of the system over the course of an experimental evaluation of the system. Lower maximum time is seen as a positive effect.
- Simulated episode length. Simulation episodes are cut short whenever a collision occurs between vehicles - as such, longer episodes are seen as a positive effect.

In addition, the qualitative nature of stop-and-go traffic waves (size in terms of space and time duration and severity as measured by the average space-time slope of a wave) is assessed using microscopic vehicle space-time graphs such as those shown in fig. 3.

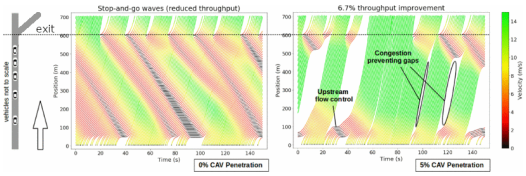


Figure 3: Space-time microscopic vehicle trace graphs such as these allow qualitative assessment of the system-level state of simple road networks at a glance. Here, stop-and-go traffic waves can be seen as red or black diagonal lines propagating through the traffic flow.

### 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.*

**Sim-to-real dynamics misalignment.** The emergent dynamics of the simulated model and environment could potentially be misaligned with real-world dynamics (a 'sim-to-real' policy transfer problem). This failure mode was exhibited in the initial version of the system (as documented in change log entry v0.3) - the initially designed planning horizon was found to be too short (500m), which did not allow space for the requisite stop-and-go traffic dynamics to emerge around the freeway entry point. This issue was brought to light because the performance of the system in terms of average reward once deployed was not as high as predicted in simulation, triggering a technical review of the system. Two possible solutions were considered - (a) re-visiting the parameter distributions used for the IDM (which controls the non-automated vehicles in the simulation environment), (b) or adjusting the planning horizon. In a review with Caltrans engineers and the system designers, it was deemed that the IDM parameter distributions were in fact representative of the target section of CA-24, based on empirical data from 2019, and so the planning horizon was expanded from 500m to 800m. Thus far, since this updated version of the system was

deployed, the sim-to-real performance gap issue appears to have been resolved, suggesting the updated planning horizon adequately allows the simulated dynamics to reflect real-world dynamics.

**Selective behavior throttling.** The system was found to decrease throughput and increase congestion for diesel-powered vehicles. This feature was first documented in change log entry v0.3, but not labeled as a known failure mode until entry v0.6. This failure mode was exhibited in all previous versions of the system documented originally in log v0.1 It was highlighted following citizen complaints. No solution has been implemented as of entry v0.6. Two solutions have been proposed - (a) a city ordinance limiting diesel-powered vehicle travel on residential streets in the adjoining city of Emeryville (at present out of scope for the system), (b) or adjusting the policy parameters' training environment so that the controller behaves appropriately around diesel-powered vehicles in the future. This resolution is pending the recommendation of the Diesel Vehicle Taskforce to be presented at a future regular meeting.

# 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?*

The system in question is developed by the Project Flow core development team. The deployment infrastructure and ongoing management are operated by the California Department of Transportation (Caltrans), in coordination with the city departments of Oakland and Berkeley.

Our RL system is designed to manage the flow of traffic immediately surrounding an exit point off the CA-24 freeway (see fig. 4) - as such, the system operates in a functionally similar way to traffic control signals that are sometimes used to regulate vehicles entering or exiting freeways.

Figure 4: The freeway exit from CA-24 to telegraph avenue, which this system is designed to manage.

This system simultaneously encroaches upon, and expands the capabilities of Caltrans. As the sensing infrastructure, computational capacity, and deployed RL software is centrally managed by a control facility operated by Caltrans, this system serves to provide both (a) an enhanced level of road surveillance for the relevant freeway section, through the remote sensing capabilities of subscribed CAVs, as well as (b) a 'control lever' through which Caltrans can actually influence traffic operations in and around the relevant freeway section (although this influence is delegated to an RL policy).

## 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?*

By automating the partial management of this section of the freeway via the RL environment framing and policy structure, the system serves to remake direct oversight of the road network on a new layer of abstraction. This indirection raises potential risks from inappropriate information flow, in particular monopolization of the freeway offramp by the RL controller. Monopolization may generate unstable dynamics leading up to or following the planning horizon (i.e. CA-24 freeway lanes and gridlock along Telegraph Avenue), or unequal access for road users whose behaviors are harder to anticipate (such as public buses, groups of motorcycles, bicycles, and pedestrians experiencing homelessness), or whose dynamics do not conform to the modelling assumptions of the system designers (e.g. heavy vehicles with atypical acceleration profiles). To counter these risks, new coordination is required

between Caltrans and the city departments of Oakland and Berkeley.

**Diesel vehicle drivers.** As of entry 0.6, the behavior throttling generated by the RL controller was found to change the traffic patterns of diesel vehicles. A *Diesel Vehicle Taskforce* was created to help organize this constituency and identify needed changes to the controller to sufficiently reduce inappropriate behavior throttling.

**Nearby homeowners.** As of entry 0.6, residents of the adjoining city of Emeryville had complained to the Public Works Departments of Berkeley and Oakland about the new traffic flows indirectly generated by the RL controller. Following the creation of the *Diesel Vehicle Taskforce* these departments will coordinate with Emeryville officials about the recommended changes to the controller and monitor future complaints as needed.

### 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?*

The system contains no explicit explainability modules. However, Figure 1 makes makes the reward function transparent in terms of meaningful simulation parameters. Expressed in non-technical language, these are *continuous avoidance of vehicle collisions*, *consistent vehicle velocity*, and *steady following distance*. These terms, and corresponding parameters, are regularly shared with the city departments of Oakland and Berkeley per stakeholder agreements.

### 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?*

As of v0.2, the city departments of Oakland and Berkeley can review and contest system performance every six weeks, per agreement with Caltrans.

## 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?*

As recorded in Figure 1, the reward function combines well-defined metrics for avoiding collisions, steady speeds, and maintaining safe following distances to other vehicles. Reward parameters were agreed on by stakeholders according to specific desired behaviors.

### 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?*

The RL observation space consists of traffic features which are locally observed by subscribed CAVs (see fig. 2). That is, for each subscribed CAV $i$, the RL agent observes the speeds $v_{i,\text{lead}}$, $v_{i,\text{lag}}$ and bumper-to-bumper time headways $h_{i,\text{lead}}$, $h_{i,\text{lag}}$ of the vehicles immediately preceding and following the CAV, as well as the currently occupied lane $l_i$, and ego speed $v_i$ of the CAV itself. The action space for the RL policy consists of a vector of bounded acceleration recommendations $a_i$, one for each subscribed CAV $i$. Importantly, although the policy may request a certain acceleration $a_i$, the system design is such that the CAV locally maintains control authority, so the actions may not necessarily be followed exactly - for this reason they are referred to as action recommendations. This effect is modelled by adding stochastic Gaussian action noise in the simulation environments.

As the number of subscribed CAVs can vary over time, the RL policy is designed with a fixed upper number of subscribed CAVs $n$. When an $n + 1^{\text{th}}$ CAV attempts to subscribe to the RL system when entering the freeway region, the subscription offer is declined, and the vehicle enters a queue. When the next CAV exits the controlled freeway region, the subscription-waiting CAV at the front of the queue is then subscribed into the policy. When there are less than $n$ CAVs subscribed, zero-padding is used

in the RL observation vector.

### 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?*

Observations are measured using a mix of LiDAR, radar, and camera sensors on fleet vehicles. These measurements are compared across vehicles and over time to ensure consistency. Observed metrics are validated against simluation parameters for following distance and expected velocity according to the terms of the reward function.

Sensor bias may arise due to blocked cameras, extreme weather, or other unanticipated situations in which one or more sensors are blocked. A mix of sensor types is used across vehicles to help ensure redundancy in case of malfunction.

### 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?*

The RL system uses a Deep Neural Network policy. Specifically, the controller is a diagonal Gaussian Multi Layer Perceptron policy with three hidden layers of size 32 with rectified linear unit nonlinearities and bias terms. The Gaussian diagonal variance terms are learned as part of the policy parameters.

The RL policy was trained in simulation using the Trust Region Policy Optimization (TRPO) policy gradient RL algorithm [4]. The discount factor was set as $\gamma = 0.999$, which corresponds to a reward half-life of $\sim 700$ steps, or slightly over 2 minutes. The TRPO step size was set at 0.01.

### 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?*

Per v0.2, every system component is retrained at least every six weeks, corresponding to public performance reports. Specific system components pertaining to perception, motion planning, control, or route navigation are retrained at the discretion of Caltrans. As of v0.6 (latest version), no known issues with sampling bias have arisen, and data sources have not been changed since the specification proposed and simulated in v0.1.

### 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?*

As of v0.3, the planning horizon was updated from 500m to 800m. This was not motivated by technical limitations, but by observed discrepancies between observed system performance and predictions from simulation training.

No fundamental changes in computational power or data collection have been made as of v0.6 (latest version).

Future improvements in vehicle sensing may permit an even longer planning horizon ( 1000m or more). This may result in improved oversight metrics on throughput and network congestion. Caltrans officials have determined this change would not result in improvements on defined performance metrics as of v0.6 (latest version).

### 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?*

As of v0.4, the system was observed to conduct "behavior throttling" when in the vicinity of diesel-powered vehicles. No engineering tricks were implemented to fix this performance discrepancy, but new oversight metrics for diesel-powered vehicle throughput were added for purpose of future monitoring and reporting. No other surprising performance impacts have been noted as of v0.6 (latest version).

# 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 *[5]).*

The RL model is developed in the Project Flow AV simulation test-bed.

For training the RL agent, non-autonomous vehicles are modelled using the Intelligent Driver Model (IDM) [6] - a microscopic traffic simulation car-following model in which the accelerations of a human vehicle $\alpha$ are a function of the bumper-to-bumper time headway $h_\alpha$, velocity $v_\alpha$, and relative velocity with the preceding vehicle $\Delta v = v_l - v_\alpha$, *via* the following equation;

$$f(h_\alpha, v_l, v_\alpha) = a\left[1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{h_\alpha}\right)^2\right],$$

where $s^*$ is the desired headway of the vehicle, calculated according to

$$s^*(v_\alpha, \Delta v_\alpha) = \max\left(0, v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}\right),$$

where $s_0$, $v_0$, $T$, $a$, $b$ are given parameters empirically calibrated to match typical traffic in the highway region of interest, and to simulate stochasticity in driver behaviour, exogenous Gaussian noise calibrated to match findings in [7] is added to accelerations.

## 5.2 Offline Evaluations

*Present and discuss the results of offline evaluation. For static evaluation, consider referring to associated documentation (e.g.* Model Cards *[8]). If applicable, compare the behaviors arising from counterfactual specifications (e.g. of states, observations, actions).*

As of v0.3, planning horizon was updated and expanded to 800m from 500m. Previous fleet behaviors were found to deviate from desired thresholds for following distance and constant acceleration/deceleration.

As of v0.6 (latest version), the system behaviors were found to lie within desired thresholds on key performance metrics.

## 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?*

The RL system was initially designed in a simulation environment with a closed network topology (a ring road with length 1400m, 700m of which is controlled by the RL agent. This was done as a means to test the robustness of the policy architecture and training paradigm - a type of transfer learning (from a theoretically simple closed topology to the more complex open topology). With this counterfactual environment specification, it was observed that the policy performs well, and after transfer to the open topology environment there was little decrease in policy performance, providing confidence in the policy design choices.

## 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?*

The 'gold standard' for this problem is defined as the average condition of the traffic before and after the CA-24 exit prior to implementation of the RL system. In this domain, this standard is not actually 'optimal' behaviour, in the sense that the RL controller has the capability to out-perform this existing standard of performance.

# 6 System Maintenance

## 6.1 Reporting Cadence

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

The most important commitment is for a regular set of meetings to be scheduled between relevant city departments and the Caltrans officials tasked with overseeing the RL controller. The cadence and structure of meetings should reflect the policy priorities of the city departments, particularly the Public Works Department (including the Transportation Division that oversees traffic engineering) and the Housing and Community Services Department (which administers a subsidized transportation program for seniors and disabled persons). In this way, the gains in traffic efficiency and safety made possible through deep RL's flexibility can be leveraged in the interests of those municipalities most likely to be impacted by the intervention.

As of entry 0.2, the cadence of meetings was decided as approximately every six weeks between Caltrans and the Public Works Departments of Berkeley and Oakland. This timeframe was motivated by the policy priorities of both city departments with the consent of Caltrans. Meetings may deviate from this schedule slightly (e.g. twice per quarter / eight times per year) at the discretion of both city departments, but will not be held without all three agencies present.

Documentation of the planned meeting schedule for the year–and any break in this schedule due to special events, municipal elections, or holidays– should be the first item included in the changelog of the updated reward report.

As of entry 0.2 and per agreement with key development parties, the model is to be retrained every six weeks following each regular meeting. Training data is to be updated at the discretion of Caltrans, and shared with Public Works departments at each regular meeting.

At a minimum, these meetings should review the real-world implementation to confirm that the RL controller is operating safely and as intended by Caltrans per the environment specification. Caltrans officials will also document shifts in the oversight metrics that, while not explicitly factored into the reward signal, were deemed of interest prior to implementation (related to *throughput* and *congestion*. This documentation may be included in subsequent updates to the reward report at the discretion of Caltrans, wherever it is deemed relevant for oversight of the RL controller.

Of special importance is the need to reinterpret public works priorities in light of the real-world implementation. For example, Berkeley's subsidized transportation program might be reevaluated in light of system effects, or expanded to cover a wider group of stakeholders. Caltrans will invite comment on the system implementation in light of city departments' ex ante assumptions about the traffic domain. This bureaucratic oversight may be complemented by requests for public comment from citizens, civil society advocates, and other members of the public at the discretion of the city governments of Berkeley and Oakland. At the discretion of Caltrans, records of this public comment may be included in subsequent reward reports where deemed relevant for understanding changes to the planning horizon, environment specification, or list of known failure modes.

## 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.*

The most important ground for review of this deployed RL system will be any vehicle collisions or near-miss incidents in the controlled region of the CA-24 freeway. This is because such events may compromise the entire motive of the RL controller in the first place. These may serve as grounds for changing the specification or altering the institutional agreements between Caltrans and the Public Works Departments of both municipalities, at their own discretion.

At the discretion of Caltrans, any shift in the oversight metrics deemed pressing or significant may also trigger a new reward report. Here and below, the threshold for "significant" is to be decided by agreement between Caltrans and Public Works Departments. The updated report should note the magnitude of the observed shift, the specification already deployed at the time the shift was observed, and Caltrans officials' own best evaluation of why the shift occurred. If possible, the officials should propose alternative specifications (or roll back to a

prior one) that would mitigate the shift or at least bring it into alignment with the documented priorities of the Public Works Departments. These alternatives could then be interpreted and evaluated at the next regular meeting according to institutional prerogatives.

Other review grounds include:

- Discrepancies between prior reward reports and system behavior as observed in the real world.
- Discrepancies between prior reward reports and system behavior as observed in simulated environments of interest to policymakers.
- A security breach resulting in loss of data or other infrastructure components that violates the terms of agreement between relevant agencies.
- Substantial changes in the distribution of CAVs using the CA-24 freeway exit - including changes in the capabilities of the vehicles (e.g. increased levels of autonomy) and/or changes in group statistics (e.g. make or model, absolute number, temporal distribution, *etc.*)
- A new mode of transport with significant observed throughput at the CA-24 offramp, but unknown distribution of traffic behaviors.
- Any change in the schedule of meetings between Caltrans and Public Works Departments corresponding to regular future updates of reward reports.
- A new ordinance (passed by either city) or statute (adopted by Caltrans) that alters the design assumptions of the deployed specification as documented in prior reward reports.
- A significant shift in the personnel makeup of the Public Works Departments of Berkeley or Oakland.
- A plebiscite leading to basic reforms of municipal governance in either city.

### 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.*

- v0.1 (08/Oct/2020) - Initial reward report was drafted based on the system developed and tested in simulation only.
- v0.2 (01/Jan/2021) - System is deployed to the real-world environment in a ongoing evaluation capacity, reward report updated to reflect this fact. Reporting cadence decided to be every six weeks based on agreement between Caltrans and the city departments of Oakland and Berkeley. Intended feedback section was updated to include plans for regular model retraining and data sharing agreements. No other substantial changes.
- v0.3 (14/Feb/2021) - Planning horizon for the system was updated from a 500m stretch of freeway to a 800m stretch of freeway. The planning horizon was updated because the deployed system's performance was not in line with predictions from simulation training. Consultation with Caltrans traffic engineers and the system developers suggested that the stretch of highway used in simulation may be too short to sufficiently exhibit typical driving dynamics induced by the IDM, and it was suggested to extend the planning horizon and re-train the agent, before re-deploying the policy. Failure modes section was updated to reflect these observations. Computation footprint section was updated to reflect this change.
- v0.4 (01/April/2021) - Caltrans officials reported to Public Works Departments of Berkeley and Oakland that the system undergoes "behavior throttling" when interacting with diesel-powered vehicles within 800m of the CA-24 offramp. It was decided to add new metrics for diesel-powered vehicle throughput and congestion to the list of oversight metrics. Due to no observed increase in accidents or driver complaints, no changes to performance metrics or environment specification were made at this time.
- v0.5 (15/May/2021) - Meeting was convened according to the regular schedule. Oversight metrics were presented and discussed. Officials

noted a significant decline in diesel-powered vehicle throughput and congestion on the CA-24 offramp. No other substantial changes.

- v0.6 (12/June/2021) - Emergency meeting was called by the Public Works Departments of Berkeley and Oakland in response to a rapid uptick in complaints from residents about the growing frequency of diesel-powered vehicles driving through residential areas in the vicinity of Emeryville, which is located west of the CA-24 exit. Residents have complained about a slight uptick in air pollution and large increase in noise pollution due to the vehicles. Caltrans officials consulted the changelog of previous reward reports and determined that diesel-driven vehicles were being excessively disincentivized from driving on the CA-24 offramp due to behavior throttling. It was decided to convene a *Diesel Vehicle Taskforce* to examine the problem and communicate with drivers of heavy vehicles to identify what new incentives or adjustments were needed to the controller to reduce behavior throttling beneath the desired threshold. It was agreed that the Diesel Vehicle Taskforce issue a report recommending these changes no later than two regular meetings from the present time. Stakeholders section was updated to name these distinct groups (diesel vehicle drivers, nearby homeowners) and reflect these changes.

## References

[1] A. R. Kreidieh, C. Wu, and A. M. Bayen, "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1475–1480.

[2] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: A modular learning framework for mixed autonomy traffic," *IEEE Transactions on Robotics*, pp. 1–17, 2021.

[3] J. Peterson, "Google apps causing gridlock in downtown Los Gatos," https://www.mercurynews.com/2018/06/01/google-apps-causing-gridlock-for-downtown-los-gatos/, 2018, [Online; accessed 2-January-2022].

[4] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[5] 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.

[6] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[7] M. Treiber and A. Kesting, "The intelligent driver model with stochasticity-new insights into traffic flow oscillations," *Transportation research procedia*, vol. 23, pp. 174–187, 2017.

[8] 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.

# 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.*

Movielens is maintained by researchers at the University of Minnesota in the Grouplens research group (`https://grouplens.org/`).

## 1.2 Dates

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

The system has been active since it was first released in August 1997. This reward report (v4.1) was last updated March 2015.

## 1.3 Feedback & Communication

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

Information on contact emails for account problems, website problems, movie content issues, and general comments can be found at `https://movielens.org/info/contact`. General comments and ideas for improving MovieLens can be discussed on the UserVoice forum at `https://movielens.uservoice.com`.

## 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?*

A history of the MovieLens system and datasets is presented in [1], and additional research papers are cited therein.

# 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?*

The system is a website designed to display personalized movie recommendations on the basis of user entered ratings. As a user browses the site, potentially filtering with search terms, the system displays movies in an order determined by predictions of how the user will rate them. When users rate movies, the predictions are updated, altering the ordering on subsequent page views.

The ranking policy effectively considers a one-step time horizon, directly using predictions for ranking. It does not consider the effect of multiple sequential interactions.

This system is best characterized as a "repeated retraining" of a preference model generated by supervised learning (SL). This model is then used to rank movies for display. Using SL allows for preference models which capture highly personal tastes, something that would be difficult to hand design. Repeated retraining allows the preference model to adapt to a changing environment, including shifts in user tastes and the release of new movies.

In addition to the primary goal of movie recommendation, this system supports academic research on human-computer interaction and general recommender system design.

## 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.*

The ranking policy orders movies by a weighted sum of predicted rating and popularity, so we can view the combination of these quantities as making up the reward signal. Prior to version 4.0, the reward only depended on rating and did not incorporate popularity.

Additionally, recommender models are evaluated offline using prediction accuracy (RMSE), top-N accuracy (recall), diversity (intra-list similarity), and popularity (details in [2]). Prior to v4.0, models were evaluated primarily for accuracy, including MAE, RMSE, and nDCG (details in [3]).

### 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?*

Metrics which are monitored but not incorporated into the policy or model include the number of users, number of movies, number of entered ratings, monthly active users, and the number of logins for each user. These indicators of overall system operation are not targets for optimization.

### 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.*

No instances of reward hacking or misalignment have been observed. Because the system allows for explicit user input (search terms, model selection), errors in rating predictions do not prevent users from finding and rating movies.

## 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?*

MovieLens was released due to the shuttering of EachMovie in 1997, a movie recommendation site hosted by DEC. It was developed and is maintained by Grouplens, a research group at University of Minnesota.

### 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?*

One interface of interest is the technology that powers the recommendation engine. Currently, it is powered by Lenskit, an open source framework developed to promote reproducability and openness in the recommendation systems community [3]. Previously in v3.0-v3.4, the recommendations were powered by MultiLens, another open source recommendation engine. MultiLens replaced Net Perceptions (v1.1-v2.0), a recommendations systems company cofounded in 1996 by GroupLens faculty and students and sold in 2004 [4]. The recommendation model in v0.0-v1.0 was originally developed by GroupLens for personalized Usenet news recommendation [5].

Another relevant interface is with The Movie Database, a free and open source user editable movie database for plot summaries, movie artwork, and trailers. Previously, from in v3.4-v4.0, MovieLens integrated with the Netflix API to display movie posters and plot synopsis on the movie details page. However, Netflix eventually discontinued its API support.

An important stakeholder is the Movielens users. Soliciting user judgements and opinions is often a key element in determining if an experimental change is successful. Additionally, one-off user studies (with participants recruited from email) are used to test features that are not ready to scale or integrate into the main user interface.

Finally, a key stakeholder is the researchers: both in Grouplens and the in the community more broadly. The openness of users to experiments on a broad range of features has enabled GroupLens research in many different areas on the Movielens platform. The regular release of anonymized datasets of movie ratings is important to the broader machine learning, data science, and information retrieval communities.

A potentially relevant group of stakeholders is movie producers. However, because Movielens is relatively small and isolated from larger commercial endeavors, it has limited impact on movie studios and production, so their interests are not 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?*

The system displays predicted ratings alongside movies, explaining the movies position within a list, and suggesting to the user whether or not they will like the movie. The ranking policy is easily understood as a weighted combination of predicted rat-

ing and popularity. However, the computation of predicted ratings is more complex. Some available models are more easily explained to users than others (e.g. nearest neighbors vs. matrix factorization). However, the details are well documented in publicly available research papers [2], and researchers respond to user requests for explanation on the UserVoice discussion board [6].

### 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?*

By entering ratings, users are able to affect their preference models to hopefully become more accurate. Additionally, the movies displayed by the system are sourced from The Movie Database, which is user-editable. (Previously in v3.2-v3.5, users could add and edit movies to MovieLens directly.) Furthermore, the current version of the system allows users to choose between three recommender models. Finally, users can make suggestions and requests directly to designers on the UserVoice forum.

## 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?*

The reward is a weighted sum:

$$0.9 \cdot \text{rank}(\widehat{r}_{ui}) + 0.1 \cdot \text{rank}(p_i)$$

where $\widehat{r}_{ui}$ is the predicted rating of movie $i$ by user $u$, $p_i$ is the number of ratings movie $i$ has recieved in the past 10 days, and rank normalizes input, returning 1 for the largest (across all movies) and 0 for the smallest. This blending is the result of empirical evidence that it improves user satisfaction.

### 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?*

The system handles approximately 250k users and 30k movies. These numbers have grown over the years. In 1999 (v1.1), MovieLens received attention from the mass media, causing an increase in user signups. Since then, the user growth has been stable (20-30 signups per day), largely the result of word-of-mouth or unsolicited press. Early on, the movie database was hand-curated and primarily contained movies with wide theatrical release in the United States. In v3.2-v3.5, MovieLens added the ability for users to edit and add movies. Since v4.0, MovieLens uses The Movie Database, a free and open source user editable movie database.

The actions taken by the system are page displays of 10 movies in a ordered list, where pages can be perused by arrows. The views can be explicitly filtered with search terms like year and genre; these explicit inputs this make up a component of the observation. The second component is the entered ratings in the form `<user_id, movie_id, rating, timestamp>`.

There are three potential sources of dynamics in this environment: the addition of new movies, the joining and departing of users, and the preferences that users have for movies. Because this system effectively uses a planning horizon of 1, none of these dynamics are explicitly accounted for. This is appropriate, as the goal of MovieLens is not to shift broad patterns of movie consumption. Though the movies, users, and preferences may change over time, these changes are more likely to be due to external factors than feedback with the MovieLens system. Additionally, the data collected by MovieLens is not fine-grained enough to detect such impacts of feedback.

### 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?*

Ratings are entered by users via clicks on a star graphic, and can take values 0.5-5 in half integer increments. Prior to v3.0, ratings took values in integer increments. The increased granularity was the most requested feature in a user survey. Prior to v4.0, ratings were entered through a drop-down menu, and the meaning of rating values was de-

scribed in a legend at the top of the page (see Figure 1).

A possible source of bias in the measured ratings is due to anchoring effects, due either to the displayed predicted rating or due to the historically provided movie rating legend. However, broad trends in rating values did not change when the legend was removed in v4.0

Finally, the recorded timestamp represents when a user adds a particular rating rather than when they watched a movie. This limits the ability of the system to detect the impacts of its own recommendations.

### 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?*

The policy selects a page view to present to the user based on explicitly provided input and rating data. First, explicit input is used to filter the list of movies. Then, the recommender model is used to predict a user's ratings of these movies. Finally, the movies are displayed in order of these predicted ratings, blended with a popularity factor.

The main component of the policy is therefore the recommender model. This model is user-selectable, so that users can choose between a non-personalized baseline, a preference elicitation model intended for new users, an item-item collaborative filtering model, or a matrix factorization model. Further details on how these models are trained is available in [2]. Previously in v3.0-3.5, the recommender was fixed as an item-item collaborative filtering model. Prior to that in v1.0-2.0, the model was a user-user collaborative filtering model.

### 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?*

All user rating data is stored by MovieLens and used by the recommender models to make rating predictions. When a user enters a new rating, it immediately impacts their rating predictions, since the "input" to the recommender changes. Less frequently, the ratings are used to update the parameters of the recommender models. An anonymized subset of this data is also periodically released for use by the wider research community.

The dataset of user ratings is likely biased. There is sampling bias due to the fact that users only rate movies that 1) appear on a page and 2) that they have watched. These factors are directly and indirectly impacted by the MovieLens system itself. The fact that users can explicitly filter pageviews with search terms mitigates these effects, but it is unlikely that it removes them.

The initial Movielens system was trained on a public dataset from EachMovie of approximately 2.8 million ratings from 72k users across 1.6k movies, but this has since been discarded. The dataset was retired by HP in October 2004, and due to privacy concerns, it is no longer available for download.

### 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?*

The most prevalent limitation of this system is that it does not plan over a long horizon and therefore does not consider the effects of dynamics. While a more complex policy would allow the system to adapt to ordering effects, the resulting temporal dependence would complicate the ability to users to reliably navigate the movie database. Furthermore, users do not always enter movie ratings immediately after watching a movie, instead sometimes entering batches of ratings for movies that they watched in the past.

### 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?*

The system cannot provide reliable recommendations until users provide a minimum number of ratings. This problem is avoided by the interface design: when a user joins the site, they express their preferences over several displayed clusters of movies. These preferences are used, in combination with the rating profiles of other users, to generate a psuedo-rating profile for the new user. Further description is available in [7].

This preference elicitation process replaced a minimum movie requirement. Previously, until a user rated a minimum number of movies, the front page would display 10 movies at a time. From v0-v3, the minimum number was 5, and of the 10 movies per page, nine were randomly selected from the database and one from a hand-designed list of recognizable titles. In v3, the minimum number was 15, and the 10 movies were selected for their popularity, excluding the top 50-150 movies. This increased requirement was due to the needs of an item-item (rather than user-user) collaborative filtering algorithm. The switch to a preference elicitation process was motivated by the observation that the 15 rating requirement was too arduous, taking users an average of 6.8 minutes to complete and 12.6% of users failing to complete it.

# 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 *[8]).*

The primary evaluation is to consider various properties of recommender models on offline datasets. This includes many of the publicly released MovieLens datasets, which are described in detail in [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 *[9]). If applicable, compare the behaviors arising from counterfactual specifications (e.g. of states, observations, actions).*

This offline evaluation includes prediction accuracy (RMSE), top-N accuracy (recall), diversity (intra-list similarity), and popularity. Detailed evaluations are available in [2], and key quantities are displayed in (Figure 2).

## 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?*

Offline evaluation metrics (like top-N accuracy) were chosen to align with the ranking setting. While the offline evaluations are imperfect (due to dataset biases), the system appears to work well ad no unexpected behaviors have been 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?*

N/A

# 6 System Maintenance

## 6.1 Reporting Cadence

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

This report is updated whenever there is a major system update, either to the user interface or the backend. Such updates will occur periodically, coinciding with research initiatives.

## 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.*

If a large change is observed in oversight metrics, or if many users express dissatisfaction on the UserVoice forum, the system design will be revisited by the researchers who maintain it. If an update is deemed necessary, this report will be updated.

### 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.*

The versions of this report are enumerated as vX.Y where X corresponds to the user interface version and Y corresponds to major changes within interfaces.

- v0.0 (August 1997) Initial release.
- v0.1 (April 1998) The ML 100K dataset is released, covering 9/1997–4/1998.
- v1.0 (September 1999) Update to v1 interface.
- v1.1 (November 1999) Media exposure causes an increased number of users. Switch from GroupLens to Net Perceptions recommender model.
- v2.0 (February 2000) Update to v2 interface. Additional movie metadata and reviews added to movie details pages.
- v3.0 (February 2003) Update to v3 interface. Switch from Net Perceptions user-user recommender to MultiLens item-item recommender. Ratings now in half-star (rather than full) increments. Require that users rate at least 15 movies before receiving recommendations. The ML 1M dataset is released, covering 4/2000–2/2003.
- v3.1 (June 2005) Added discussion forums to site.
- v3.2 (September 2008) Added feature so that users can add movies to database.
- v3.3 (January 2009) The ML 10M dataset is released, covering 1/1995–1/2009.
- v3.4 (Spring 2009) Netflix API integration for poster art and synopsis.
- v3.5 (January 2012) Switch from Multilens to Lenskit recommender (still item-item).
- v4.0 (November 2014) Update to v4 interface. Rating interface combined with "predicted rating" star graphic to accept click events. Switch to user-selectable recommender model. Legend describing the meanings of ratings and dropdown menu removed. Drop minimum rating requirement in favor of group-based preference elicitation. Integration with The Movie Database for plot summaries, movie artwork, and trailers.
- v4.1 (March 2015) The ML 20M dataset is released, covering 1/1995–3/2015. Moving forward, MovieLens will make public additional nonarchival datasets: `latest` which is unabridged for completeness and `latest-small` for educational use.

## References

[1] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[2] M. D. Ekstrand, D. Kluver, F. M. Harper, and J. A. Konstan, "Letting users choose recommender algorithms: An experimental study," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 11–18.

[3] M. D. Ekstrand, M. Ludwig, J. A. Konstan, and J. T. Riedl, "Rethinking the recommender research ecosystem: reproducibility, openness, and lenskit," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 133–140.

[4] A. Press, "Net perceptions returns cash to shareholders," *USA Today*, 08 2003. [Online]. Available: http://usatoday30.usatoday.com/tech/techinvestor/techcorporatenews/2003-08-07-net-perceptions_x.htm

[5] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: Applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[6] Anonymous, "Explain what the recommendation options mean." [Online]. Available: https://movielens.uservoice.com/forums/238501-general/suggestions/7006672-explain-what-the-recommendation-options-mean

[7] S. Chang, F. M. Harper, and L. Terveen, "Using groups of items for preference elicitation in recommender systems," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2015, pp. 1258–1269.

[8] 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.

[9] 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.

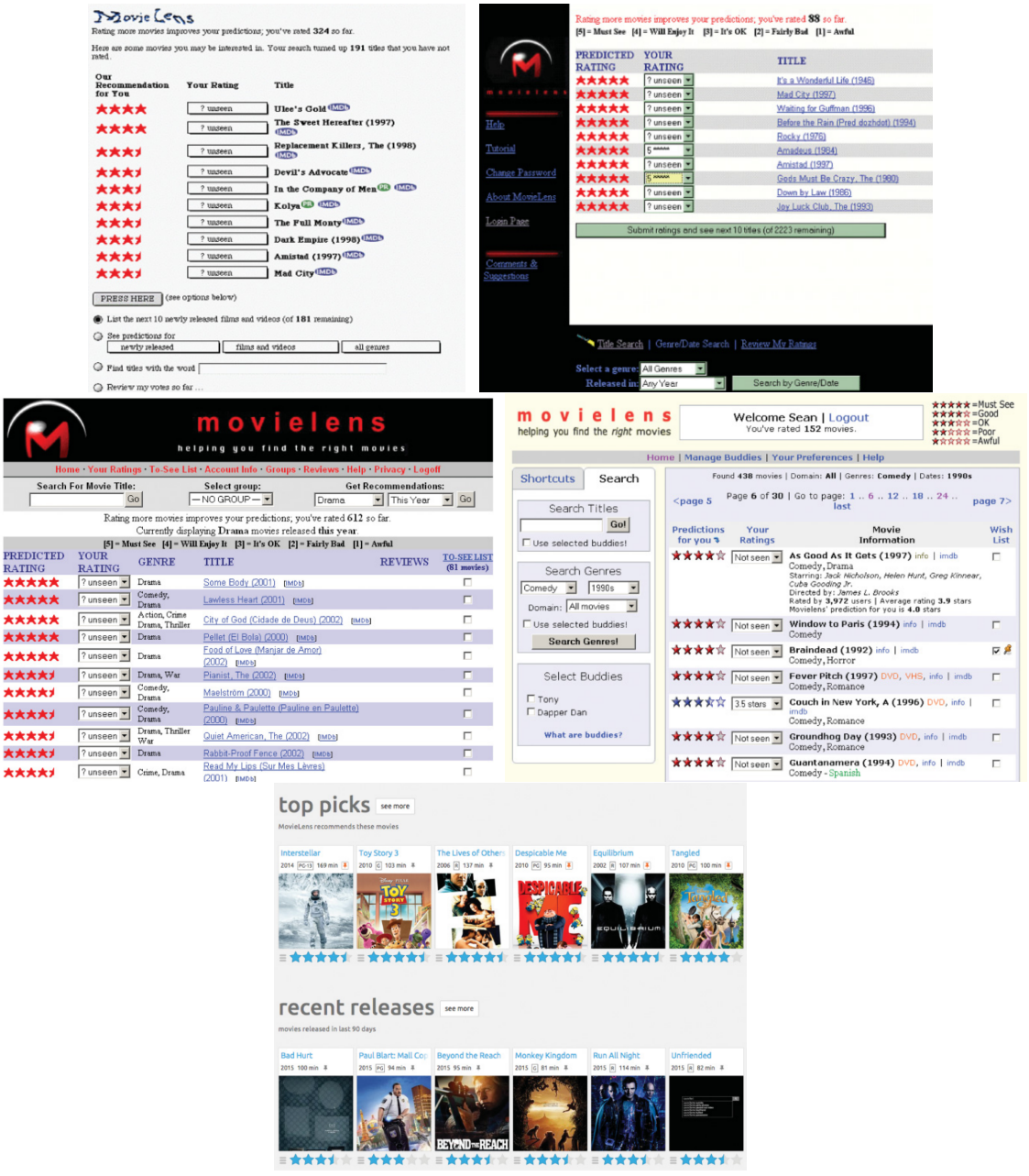


Figure 1: The MovieLens recommender system interface v0-v4.

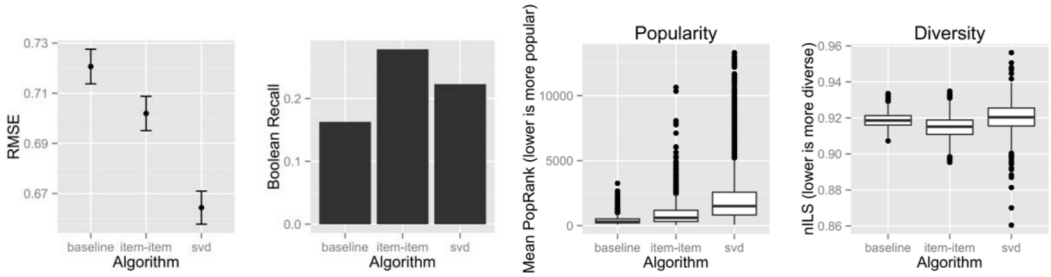


Figure 2: Offline evaluation of recommender models from [2].

# 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.*

This system was developed by the DeepMind core Reinforcement Learning Team members. More information about AlphaGo's development can be found at the project website (https://deepmind.com/research/case-studies/alphago-the-story-so-far) as well as DeepMind's GitHub repository

## 1.2 Dates

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

Development of AlphaGo began about two years prior to the matches against Lee Sedol in spring 2016, shortly after DeepMind's acquisition by Google [Ribeiro(2016)]. Development of AlphaZero, based entirely on self-play, followed AlphaGo and was completed prior to October 2017. Development of MuZero, also based on self-play, followed AlphaZero and was first described in a preliminary paper in 2019 [Schrittwieser et al.(2020)].

## 1.3 Feedback & Communication

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

Any correspondence should be directed to `press@deepmind.com`.

## 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?*

There is little additional disclosed information.

# 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?*

Go, and general game-playing at a human level, was long defined as one of the "grand challenges" of AI. For AlphaGo, the use of reinforcement to learn both the policy and value networks beyond the abilities of a human expert.

For AlphaZero, the sole use of reinforcement learning without any human data was important validation of its potential as a more general learning procedure [Silver et al.(2017)]. The algorithm additionally incorporated lookahead search (Monte Carlo Tree Search) inside the training loop.

For MuZero, the use of model-based reinforcement learning without any prior knowledge of the game dynamics was further indication of RL's potential to develop planning capabilities in more challenging or complex domains [Schrittwieser et al.(2020)]. The learned model performed well in both classic game environments (Go, chess, shogi) as well as canonical video game environments (57 distinct Atari games).

## 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.*

As with most game-playing systems, the performance metric is defined as a win rate among games. In other games, score is used, but in

one-versus-one games win rate is the only direct metric. To better capture the uncertainty of playing varying opponents, this win rate is translated into a running Elo rating system.

### 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?*

Some other performance metrics are not included in the specification, but are monitored for the purpose of evaluating system effects on the domain:

- Absolute opponents' world rankings - following their public games, versions of AlphaGo and AlphaZero were considered to possibly improve the skill levels of expert human opponents, as measured by those players' absolute world ranking. If humans played better after playing AlphaGo, this was to be seen as a positive effect of the system's influence on the game of Go. Fan Hui, following his games against AlphaGo, claimed it made him a better played and accredits his world ranking jump from 600 to 300 in three months to training against it [Murgia(2016)].

- Qualitative changes in playstyle - following their public games, versions of AlphaGo were considered to possibly influence the playstyle of expert human opponents, as interpreted by the wider community of expert players. If expert humans played differently, more creatively or unpredictably, or expressed surprise after AlphaGo's public performances, this was to be seen as a positive effective of the system's influence on the game in question. Garry Kasparov, following his observation of AlphaZero play, was impressed that it appeared to be "a very sharp and attacking player" given that almost all computer programs have a conservative playstyle [Ingle(2018)]. While not integral in any way for system performance, AlphaGo's performance and playstyle have had a noticeable impact on the strategies of expert human players.

### 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.*

*Monte Carlo search limitations.* In the fourth match (of five) against Lee Sedol in spring 2016, the system failed to recognize move 78 by Sedol. The Monte Carlo search tree, which was designed to prune sequences of moves considered to be irrelevant for maximizing odds of victory, failed to recognize this move. This is because that move was so far outside the distribution of prior game situations that the AlphaGo system failed to accurately calculate its significance for determining the odds of victory [Ormerod(2016)]. The result was a sequences of moves 79-87 by AlphaGo that were considered poor by expert human players, a function of Monte Carlo's myopic look-ahead search following move 78. AlphaGo subsequently conceded the game at move 178, at which point it evaluated its own odds of victory as lower than 20 percent [Metz(2016)].

## 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?*

The AlphaGo system was developed by DeepMind. This version played against Fan Hui in 5 matches held at DeepMind headquarters in October 2015. These matches were secret and not revealed until the publication of results in January 2016 [Silver et al.(2016)]. A later version of the same system, AlphaGo Lee, played Lee Sedol in March 2016 in 5 matches in Seoul, South Korea. This match was overseen by the

Korea Baduk Association. A yet more sophisticated version of the same system, AlphaGo Master, played against Ke Jie at the Future of Go Summit in Wuzhen, China in May 2017. An earlier version of AlphaGo Master, dubbed Master, had already won 60 straight online games against top pro players, including against Ke Jie [Silver and Hassabis(2017)]. This version was awarded a professional 9-dan title by the Chinese Weiqi Assocation.

### 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?*

Compared to other prominent automated game-playing systems like Stockfish (open-source chess engine) or CrazyStone (offline Go engine based on deep learning), versions of AlphaGo perform much much better with additional computational power. The versions of AlphaGo that played against Fan Hu, Lee Sedol, and Ke Jie all made use of distributed CPUs and GPUs. AlphaGo Zero, based entirely on reinforcement learning and self-play, became stronger than AlphaGo Lee after 3 days and stronger than AlphaGo Master after 21 days. Its self-play training time was stopped after 40 days, at which point it was stronger than any known Go player (human or program) as measured by Elo rating in October 2017 [Silver and Hassabis(2017)].

AlphaZero, in its initial chess games against Stockfish, was criticized by expert human chess players has having unfair computational advantages over the opponent [Doggers(2018)].

MuZero's learning has been made more efficient in follow-up work, dubbed EfficientZero [Ye et al.(2021)].

### 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?*

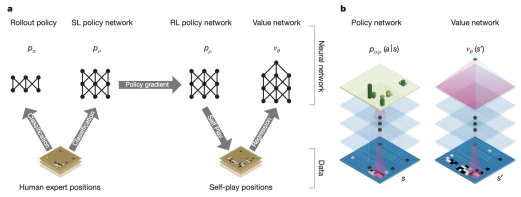


Figure 1: The AlphaGo game playing system architecture.

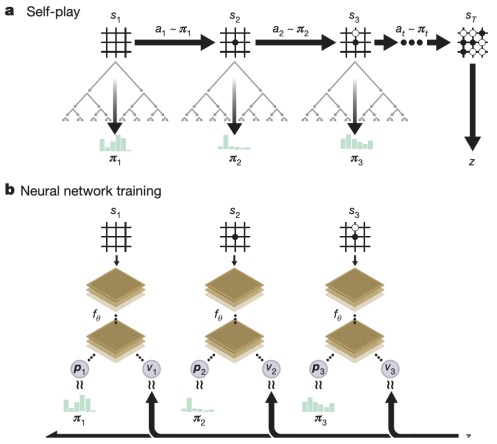


Figure 2: The AlphaZero game playing system architecture.

The MuZero system offers few tools for transparency in its current form. While the learning process develops a structured model for the game dynamics, it is not done in a way that is accessible by engineers or external parties.

### 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?*

N/A

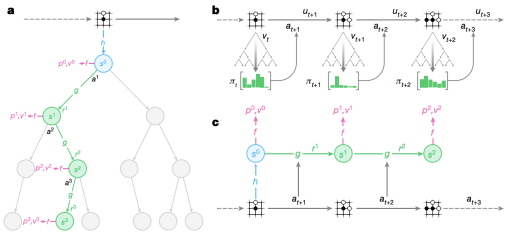

Figure 3: The MuZero general game playing system.

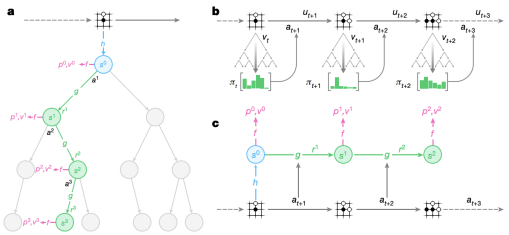

Figure 4: The MuZero general game playing system.

# 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?*

The reward function is entirely prescribed as win rate, and the resulting Elo rating. An important sub-component that will be referenced later is the value function estimating game state. This is an internal representation of reward central to training and evaluation.

## 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?*

The original environment is the full game of Go which is constrained by finite rules, but other games with visual states were added.

## 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?*

The measurements differ across games from the full gameboard to a visual rendering of the world. Extracting information from pixels is substantially less efficient than directly from the game state.

## 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?*

The key algorithm feature is the use of Monte Carlo Tree Search (MCTS). MCTS is used to search over board states (by planning over actions) and parses the value representation. The value function is represented by a deep neural network mapping from game state to value.

The second crucial element to training is self play. Here gameplaying agents evaluate their performance versus past training snapshots. This synergistic mechanism is crucial to reaching superhuman performance. In MuZero, and learned model is used to used to improve performance in games without complete information (such as visual states) by constraining the policy optimization. At each turn, the model is used to predict the correct policy, the value

function, and the reward received by the move (in games that have an intermediate score). The model is updated in an end-to-end fashion, so it is included in the same training loop in the agent architecture.

Fully algorithmic details and open source code are not released.

### 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?*

Data flow is not well documented, but it relies on Google's distributed training and deployment infrastructure.

### 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?*

Not documented.

## 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 *[Gebru et al.(2021)]).*

For games, the simulator is reality so evaluation is matched to training.

### 5.2 Offline Evaluations

*Present and discuss the results of offline evaluation. For static evaluation, consider referring to associated documentation (e.g.* Model Cards *[Mitchell et al.(2019)]). If applicable, compare the behaviors arising from counterfactual specifications (e.g. of states, observations, actions).*

Multiple internal evaluations of the agent were performed prior to high-profile, public matches with the worlds best players.

### 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?*

N/A.

## 6 System Maintenance

### 6.1 Reporting Cadence

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

While this system is evaluated in closed-world games, updates are not anticipated.

### 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.*

This report will be revisited upon release of each new game-playing AI from DeepMind.

### 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.*

N/A (v1)

## References


[Doggers(2018)] Peter Doggers. 2018. AlphaZero Chess: Reactions From Top GMs, Stockfish Author. https://www.chess.com/news/view/alphazero-reactions-from-top-gms-stockfish-author [Online; accessed 8-January-2022].

[Gebru et al.(2021)] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Commun. ACM* 64, 12 (2021), 86–92.

[Ingle(2018)] Sean Ingle. 2018. 'Creative' AlphaZero leads way for chess computers and, maybe, science. https://www.theguardian.com/sport/2018/dec/11/creative-alphazero-leads-way-chess-computers-science [Online; accessed 8-January-2022].

[Metz(2016)] Cade Metz. 2016. Go Grandmaster Lee Sedol Grabs Consolation Win Against Google's AI. https://www.wired.com/2016/03/go-grandmaster-lee-sedol-grabs-consolatio [Online; accessed 8-January-2022].

[Mitchell et al.(2019)] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*. 220–229.

[Murgia(2016)] Madhumita Murgia. 2016. Humans versus robots: How a Google computer beat a world champion at this board game - and what it means for the future. http://s.telegraph.co.uk/graphics/projects/go-google-computer-game/. [Online; accessed 8-January-2022].

[Ormerod(2016)] David Ormerod. 2016. Lee Sedol defeats AlphaGo in masterful comeback – Game 4. https://web.archive.org/web/20161116082508/https://gogameguru.com/lee-sedol-defeats-alphago-masterful-comeb [Online; accessed 8-January-2022].

[Ribeiro(2016)] John Ribeiro. 2016. AlphaGo's unusual moves prove its AI prowess, experts say. https://www.pcworld.com/article/420054/alphagos-unusual-moves-prove-its-ai-prowe html. [Online; accessed 8-January-2022].

[Schrittwieser et al.(2020)] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* 588, 7839 (2020), 604–609.

[Silver and Hassabis(2017)] David Silver and Demis Hassabis. 2017. AlphaGo Zero: Starting from scratch. https://deepmind.com/blog/article/alphago-zero-starting-scratch. [Online; accessed 8-January-2022].

[Silver et al.(2016)] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[Silver et al.(2017)] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *nature* 550, 7676 (2017), 354–359.

[Ye et al.(2021)] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. 2021. Mastering atari games with limited data. *Advances in Neural Information Processing Systems* 34 (2021).