Reward Reports for Reinforcement Learning

Thomas Krendl Gilbert
tg299@cornell.edu
Digital Life Initiative, Cornell Tech
New York, New York, USA

Nathan Lambert
nathan@huggingface.co
HuggingFace
Berkeley, California, USA

Sarah Dean
sdean@cornell.edu
Cornell University
Ithaca, New York, USA

Tom Zick
tzick@jd24.law.harvard.edu
Harvard Law School
Boston, Massachusetts, USA

Aaron Snoswell
a.snoswell@qut.edu.au
Centre for Automated
Decision-Making and Society,
Queensland University of Technology
Brisbane, Queensland, Australia

Soham Mehta
sgm2160@columbia.edu
Columbia University
New York, New York, USA

ABSTRACT

Building systems that are good for society in the face of complex societal effects requires a dynamic approach. 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 has shown that the effects of feedback and optimization objectives on system behavior can be wide-ranging and unpredictable. In this paper we sketch a framework for documenting deployed and iteratively updated learning systems, which we call *Reward Reports*. Taking inspiration from technical concepts in reinforcement learning, we outline Reward Reports as living documents that track updates to design choices and assumptions behind what a particular automated system is optimizing for. They are intended to track dynamic phenomena arising from system deployment, rather than merely static properties of models or data. After presenting the elements of a Reward Report, we discuss a concrete example: Meta's BlenderBot 3 chatbot. Several others for game-playing (DeepMind's MuZero), content recommendation (MovieLens), and traffic control (Project Flow) are included in the appendix.

CCS CONCEPTS

• **Theory of computation** → **Sequential decision making**; • **General and reference** → **Evaluation**; • **Software and its engineering** → **Documentation**; • **Human-centered computing** → *Walkthrough evaluations*; • **Social and professional topics** → *Socio-technical systems.*

AIES '23, August 08–10, 2023, Montréal, QC, Canada

ACM ISBN 979-8-4007-0231-0/23/08…$15.00
https://doi.org/10.1145/3600211.3604698

KEYWORDS

Reward function, reporting, documentation, disaggregated evaluation, ethical considerations

ACM Reference Format:
Thomas Krendl Gilbert, Nathan Lambert, Sarah Dean, Tom Zick, Aaron Snoswell, and Soham Mehta. 2023. Reward Reports for Reinforcement Learning. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES '23), August 08–10, 2023, Montréal, QC, Canada.* ACM, New York, NY, USA, 47 pages. https://doi.org/10.1145/3600211.3604698

1 INTRODUCTION

Algorithmic systems often impact society in profound and difficult to anticipate ways. To assess risk, a system designer must take into account not only immediate impacts on stakeholders, but also third party externalities and the feedback loops they may engender. As AI matures and is deployed in new ways, emerging capabilities challenge what designers and other stakeholders assume algorithmic systems can do, making a priori risk assessment of these "agents" even harder [15].

The recent rise of dialogue agents powered by Large Language Models (LLMs) is a good example. These agents are trained on inconceivably large data corpora, deriving extremely sophisticated linguistic representations. More importantly, their conversations with users have effects that last beyond one-off interactions. Particular responses cannot be meaningfully isolated from prolonged exchanges; users may be influenced long afterward. Beyond the biases present in individual outputs, both the dialogue agents themselves and derived data artifacts (e.g. user chat histories) may be integrated into search architectures or other online services that qualitatively alter users' relationship with the web. As a result, diagnostics or audits of LLMs alone are an insufficient guide for design interventions. To manage feedback-heavy systems responsibly, the repercussions of algorithmic changes must be reflexively documented. In other words, both emergent system behaviors and the changing assumptions of key stakeholders about them must be accounted for in an ongoing and responsive manner.

We propose a new form of documentation, *Reward Reports*, to move the research community towards a world where these changes are regularly tracked, reflected upon, and responded to. For designers, this documentation would aid internal efforts at reverse

engineering the behavior of black box systems, and provide a framework to disentangle complex effects as they manifest. Moreover, a standard mechanism for continuous transparency would help harmonize external efforts at domain monitoring and inform regulatory action. To this end, we are building Reward Reports for popular machine learning systems via community contributions[1].

Multiple frameworks for documenting AI systems, datasets and models already exist [25, 42, 51]. However, these approaches all aim to track sources of potential bias or harm within a static machine learning (ML) paradigm. One might imagine that issuing successive Model Cards would be sufficient to monitor the behavior of deployed systems. However, system architectures display several key features that would make such a regime insufficient as a basis for accountability. First, 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. Reflecting on these changes explicitly should be a part of deploying AI responsibly. Second, Model Cards do not document the design decisions leading to this particular ML system–why specific learning algorithms were chosen, how the designers expect the system to operate, and what evidence would change these expectations. Checking assumptions is a cardinal part of promoting accountability. Third, learned task representations often lie behind external interfaces (such as static APIs), access to which is decoupled from how trained models may change over time. Bridging this gap is crucial in order to understand AI systems in context. These features transcend existing documentation regimes. The presence of diverse feedback profiles and ongoing dynamics suggests unique risk vectors that must be made interpretable through documentation. Full accountability requires a cohesive understanding of how the system incorporates different types of feedback: from historical data, from stakeholders, and from a system's own usage once deployed. Reward Reports are designed to foreground these elements, allowing better insight into the societal impacts of data-driven optimization systems where feedback effects play a key role.

As a framing device, Reward Reports utilize reinforcement learning (RL), a sub-field of ML that is tasked with solving sequential, open-ended problems. RL provides a dynamical lens that is broadly applicable to many algorithmic systems with repeated data-driven optimizations. Critically, this lens is also applicable to 'static' ML systems. In many of these systems, new behaviors emerge post-deployment in response to ongoing usage, and as the system is retrained or applied to new populations. 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, whether explicitly or implicitly construed as RL. They 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.

In Section 2, we situate Reward Reports within the existing literature on AI documentation and governance mechanisms. In Section 3, we review optimization in the context of feedback, focusing on the notions of action, objective, and adaptation. In Section 4, we contextualize these elements of optimization within a taxonomy of feedback. Finally, in Section 6, we present an example Reward Report for the BlenderBot 3 "AI chatbot" developed and deployed by Meta. Throughout, we maintain a running example of a dialogue agent to illustrate the challenges in documenting feedback-laden and data-driven optimization systems, whether or not they explicitly utilize the RL framework.

2 RELATED WORK

2.1 Documentation for AI Systems

There are several existing proposals for AI system documentation, with some frameworks focused on specific aspects of an AI system, while others examine the system as a whole. Reward Reports are focused on documenting risks of dynamic machine learning systems. This complements current documentation efforts by explicating intended performance in light of feedback effects that may emerge over time due to re-training or shifts in usage.

Data documentation. Documenting data, regardless of resulting systems, is a well explored avenue [4, 8, 20, 25, 30]. These efforts have helped foster discussion on responsible data collection, as well as showcase issues of bias and representation. However, this paradigm is most impactful within the batch or offline setting of static unsupervised or supervised ML. Dynamic systems driven by sequential feedback also use data, but not only in the form of static datasets. Rather, both RL and deployed ML systems generate and eventually transform data. This is due both to the optimization decisions of the systems themselves and the dynamics of the environment in which they operate. Documenting original data alone cannot reveal the risks of dynamic datasets and feedback driven systems.

Static system documentation. Some proposals for ML documentation are specific to model [42], domain [36, 47], or outcome [61]. Reward Reports might be a useful supplement to these approaches for several reasons. With regards to Model Cards, there is substantial deliberation entailed in mapping between the chosen optimization and resultant behavior in dynamic systems. For example, designers must consider a range of alternative specifications that were technically feasible pre-deployment, as well as the types of feedback available to help optimize post-deployment. These aspects of system design cannot be captured solely by documenting the model. Furthermore, unlike domain-specific reporting frameworks, we provide a general template that can be applied to any specific application. Our work has similarities with previous proposals for AI Ethics Sheets [43], Fact Sheets [6, 51], or Scorecards [12], but uniquely focuses on prompting deliberation about the feedback-driven risks inherent to dynamic systems.

Auditing and Assessment approaches. Algorithmic Impact Assessments (AIA) offer a framework for evaluating risks before an

[1]Reward Reports are produced and maintained here: https://github.com/RewardReports/reward-reports.

AI system is developed or acquired [50, 55]. AIAs were inspired by environmental impact assessments, which provided one path to regulate industries in which corporate expertise outpaced government capacity. These frameworks presume an agency-vendor relationship, and focus narrowly on the procurement of automated decision systems. Meanwhile, many audit mechanisms attempt to confirm either internally or externally whether a given system conforms to a legacy standard or regulation [46]. Reward Reports are intended to supersede these *ex-ante* concerns, engaging instead with the necessarily non-linear and circuitous process of refining the specifications of a feedback system.

2.2 Societal Risks of Dynamic Machine Learning Systems

The AI ethics community is increasingly acknowledging the important role of feedback and dynamic effects on system behaviours. While critical and discursive interpretations of feedback are more common in static ML than RL research [3, 40, 48], the technical RL community is also increasingly aware of the limitations of current algorithmic approaches and evaluation paradigms. For instance, the RL research community has begun to reflect on the unique risks and challenges that may be posed by RL systems, in particular those that leverage black-box neural networks for decision-making. There are whitepapers charting these challenges [26, 63], as well as attempts to address societal risks through technical means [14, 22, 62, 67]. RL from human feedback (RLHF) has recently served both as a technique in the process of training LLMs and also as a metaphor for value alignment [7]. Recent general audience books have echoed these tensions [16, 52]. 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 to organize these efforts' reflections into an instrument of deliberation and accountability.

2.3 AI Governance

ML documentation can be used as a governance mechanism to dictate safer machine learning practices. As reflected in the growing number of proposed AI governance frameworks [18, 24, 38, 49, 64, 66], ML and adjacent communities have increasingly acknowledged socio-technical risks and the need for novel harm mitigation strategies [5, 19, 56]. These frameworks have begun to influence legislation. For example, the Canadian government has mandated the Algorithmic Risk Assessment tool for procurement [13], and current draft legislation from the EU commission calls for AI system documentation tailored to forms of risk (*e.g.* in Title 3, Ch. 2, Art. 10-14) [1]. While these frameworks provide needed prohibitions and protections, they favor interpretive flexibility over specific design decisions, and often assume static AI systems that need strictly *ex-post* documentation. In contrast, Reward Reports would track requisite design decisions, and provide an interface for stakeholders to reflect on the validity of those design choices over time. This would in essence dissolve the boundary between *ex-ante* and *ex-post* assessment.

3 BACKGROUND

In this section, we reframe the concept of data-driven optimization in terms of dynamical systems. The type of optimization we describe encompasses the training of large language models, as well as their effects on the world once deployed. We begin by reinterpreting these dynamics in terms of action, objective, and adaptation. This taxonomy emphasizes the *use* and *behavior* of learned models rather than the closed act of developing them. We then review the RL framework, and recount its broader connections to optimization and machine learning.

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 shortcut path [45]. Similarly, the optimization of datacenter operations at Google uses the predictions of trained models to adjust set points and load distribution–the predictions serve as a catalyst for action in the real world [23].

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 [44]. Likewise, content interaction models have vastly different impacts when they are used to uprank posts predicted to receive many 'likes' compared with those predicted to receive long comments [28].

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 [21]. On the other hand, adaptivity can also exacerbate biased data. For example, predictive policing systems can amplify racial biases in arrests by directing more patrols towards areas with more documented arrests [41, 53].

Large Language Models through the RL Lens

We can draw a direct analogy between the MDP setting and the evaluation of language models. While language models are not Markovian in an exact sense, the notion of "state" can be applied to the *conversation text* at stake in a particular user interaction. The "observations" of the language model would consist of the *subset of the historic conversation text* that fits within the context window (the "time horizon") of the language model. Likewise, the "actions" taken by a dialogue agent consist of the *token sequences that are generated* to form responses to user queries. The "dynamics" of this system correspond to the *user responses* to dialogue agent generations - updating the conversation state by moving the sequence of conversation forward. Finally, the *performance metrics* (e.g. loss and/or regularization function(s), user 'thumbs up'/'thumbs down' feedback) that shape the behaviour of

the language model during pre-training, fine-tuning, and subsequent updates, can be considered as a source of scalar "reward" feedback that depends on the specific conversation state at a given point in time. This lens is central to the recent surge in the use of reinforcement learning from human feedback (RLHF) [17] to further fine-tune language models with respect to human values.

Action, objective, and adaptation are important for ensuring that systems work as intended, even in cases where they are not explicitly defined as part of an underlying model. This is especially true for large language models, which act by responding to natural language queries according a variety of engineered objectives: accurate prediction of the next token in a dialogue sequence, but also disparate constraints such as safety, helpfulness, etc. Moreover, they already function as parts of a larger adaptive system, as retrained models (GPT-3, GPT-3.5, GPT-4) have been designed and evaluated differently based on their successive integration with the ChatGPT interface. At present, designers are missing a framework for capturing these properties as dynamic elements of techniques for model optimization and deployment.

3.2 Reinforcement Learning

The reinforcement learning (RL) framework succinctly encompasses action, objective, and adaptation. RL agents take actions, are motivated by a reward signal which encodes the objective, and adapt based on the feedback from interactions. 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. This is even more true of language models whose post-deployment social effects are readily understood within an RL framework–to say nothing of the explicit pre-deployment use of RL from Human Feedback (RLHF) to fine-tune these models.

In Reinforcement Learning, 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 expected cumulative reward over some time horizon H:

$$\mathbb{E}\left[\sum_{t=0}^{H} \gamma^t r_t \mid \pi\right]$$

where the discount factor $\gamma \in (0, 1]$ trades off between immediate and future potential rewards. As outlined above, this paradigm captures many problems of interest, from choosing advertisements that are most likely to result in a click [37, 39] to determining the best dosing schedule for a patient [59].

A key element of RL is how actions affect the future behavior of the environment. This dependence is often modeled as a Markov Decision Process (MDP) [9]. 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, *i.e.*

$$\mathbb{P}\{\vec{s}_{t+1} = \vec{s} \mid \vec{s}_0, \ldots \vec{s}_t, \vec{a}_0, \ldots \vec{a}_t\} = \mathbb{P}\{\vec{s}_{t+1} = \vec{s} \mid \vec{s}_t, \vec{a}_t\} \,.$$

The transition probability distribution mapping state-action pairs to subsequent states is referred to as the *system dynamics*. Furthermore, the scalar reward signal is defined to be determined by the state, so that $r_t = r(\vec{s}_t, \vec{a}_t)$ for some reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, 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 $\vec{s}_t = [\vec{o}_{t-h}, \ldots, \vec{o}_t]$), and the policy is typically parametric, with a parameter vector denoted θ.

The RL framing is general, so other machine learning paradigms can be viewed as special cases of it as long as key assumptions are named. 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. While standard supervised learning frameworks do not consider how to update or retrain on the basis of interaction, there are intermediate points. Online learning situates supervised learning systems in a sequentially evolving environment [58], while the study of bandit problems reduces RL to the static regime where actions do not affect the environment [10]. The boxed example illustrates how an RL lens can be richly applied to language models as long as terms like *horizon* and *state* are aligned with salient metrics and performance criteria.

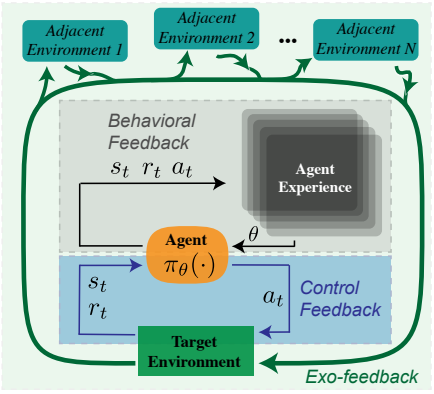
4 MOTIVATION

The RL lens is useful not only because ML deployments often operate in dynamical environments. It also expounds *feedback* between the environment and the deployment. In this section, we review three levels of feedback that characterize RL systems, and then motivate Reward Reports as documentation for tracking how and why feedback has been organized.

4.1 A Taxonomy of Feedback

We categorize feedback into three types: *control* feedback from state to action, *behavior* feedback from the data to the policy, and *exogenous* (or exo-)feedback from the target environment to adjacent entities [27]. The types of feedback are compared in Tab. 1 and visualized in Fig. 1. Throughout, we continue to illustrate these types of feedback through comparison to a deployed dialogue agent powered by a large language model.

Control feedback. Control feedback maps observations or states to actions. In the case of a dialogue agent, control feedback is the autoregressive language model itself–a conditional probability distribution from token sequences to subsequent tokens. "Intelligent" behaviors arise because actions are constantly adjusted on the basis



Type of feedback	Feedback Channel	Dynamics	Specification Element
Control	Agent-Environment	Reaction	Actions
Behavior	Policy-Reward	Evaluation	Rewards
Exogenous	Environment-Domain	Drift	States & Observations

Figure 1 & Table 1: The relationship between types of feedback, the channel through which information flows, the relationship to dynamics, and specification element(s). Different parameters and data interact over time to create dynamic properties internal and external to the RL agent.

of observations, even though the rules that control the behavior (the language model distribution) remains the same.

Behavior feedback. Behaviour feedback maps data from the environment to a learned policy. This form of feedback occurs when RL systems automatically adapt their policy based on reward, or when a deployed ML system is updated (e.g. with new weights) to better suit the deployment environment. Questions of reaction become questions of trial-and-error evaluation: "What token should follow the preceding tokens" evolves into "Can this dialogue agent be made safer/more useful/better aligned with user needs *etc.* through fine-tuning or better prompt engineering" (for example). The ability to learn from experience is part of what makes RL systems so powerful, and is what makes RL useful in domains that are difficult to otherwise model. For example, it would be challenging to hand-design a policy for generating natural language responses to user queries, but data-driven approaches have made this task tractable.

Exogenous feedback. Exogenous feedback occurs when the application domain itself shifts in response to the deployed system. These shifts could be due to the system interacting with political or economic conditions that are outside the scope of its formal specification. Note that the concept of exogenous feedback is richer than the concept of covariate or distribution shift commonly discussed in supervised learning [33]–exogenous feedback explicitly foregrounds what lies behind and instigates the shift, i.e. the interactions between the deployed system and the environment. Dialogue agents are particularly prone to this type of feedback, as their emerging effects on web search, content recommendation, and diverse writing disciplines already demonstrate. In principle, if such dynamics could be predicted by an RL agent, they could be incorporated within behavior or control feedback, but in practice, it is not clear that this is possible–the observations would need to be sufficiently rich and the planning horizon sufficiently long. For these reasons, exogenous feedback highlights the potential of externalized risks in the RL framework.

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 fundamentally unavoidable in practice. Furthermore, it is unrealistic to articulate all possible system specifications *a priori*. This means that a single specification may simultaneously implicate all three forms of feedback (conrol, behavior, exogenous) outlined above.

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. This may be especially true for dialogue agents whose models may be technically static (insofar as parameters are not updated in real-time in response to user feedback), but whose dynamical effects on society are impossible to specify in advance. 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

Here we prescribe Reward Reports, a structured series of design inquiries for automated decision systems (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. As pivotal properties may not be known until the system has been deployed and monitored, the onus is on designers to sustain documentation over time. This makes Reward Reports into living documents that both avoid the limitations of simple, unidirectional answers (*e.g.* yes or no) and illuminate societal risks

over time. Moreover, the changelog component of a Reward Report becomes an interface for stakeholders, users, and engineers to continuously oversee and evaluate the documented system. Thus, Reward Reports are a prerequisite to sociotechnical reflection about the system behavior.

Appendix A includes an empty template for a Reward Report, including descriptions of the content for each component. In this section we present the six main components that compose a Reward Report. These components are 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.

Reward Report Contents

- **System Details**: Basic system information.
  - System owner
  - 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.
  - Deployment agency
  - Stakeholders
  - 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 and suggested inquiries.

5.1 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?

5.2 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 [34], and description of how the current system avoids this.

5.3 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 oversee the ongoing post-deployment operation of 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?

5.4 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?

5.5 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* [25]).
(2) **Offline evaluations**: Present and discuss the results of offline evaluation. For static evaluation, consider referring to associated documentation (*e.g. Model Cards* [42]). 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?

5.6 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.

6 EXAMPLES

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 in various design contexts. Below, we focus on the Reward Report for the BlenderBot 3 chatbot deployed in August 2022 [2]. We refer the reader to the appendix for the complete Reward Reports for BlenderBot 3 and several other examples including game-playing (DeepMind's MuZero [54]), content recommendation (MovieLens [29]), and traffic control (Project Flow [65]).

6.1 BlenderBot 3: A Chatbot Designed to Improve Over Time Through Feedback

BlenderBot 3 is a recent chatbot "designed to improve its conversational skills and safety through feedback from people who chat with it, focusing on helpful feedback while avoiding learning from unhelpful or dangerous responses" [2]. To achieve this, the chatbot incorporates more than one type of feedback. First, it incorporates a significantly larger language model than its predecessors at 175 billion parameters, unlocking new conversational capabilities via a larger capacity policy–a level of *control feedback*. Second, it includes an interface for human users to provide real-time feedback on its conversational outputs if they are biased, inappropriate, or lack context–a kind of "reward signal" that uses behavior feedback for eventual model updates. Finally, its designers have articulated their high-level goal to release data about the chatbot's performance to the broader AI research community as a means to uncover new strategies for making future AI systems safer and more engaging to users–a kind of constructive form of exogenous feedback, beyond the technical specification scope.

In parallel with BlenderBot 3's deployment, the designers made both its language model (OPT-175B) and associated model cards public to the AI research community. We have corresponded and worked with the designers to synthesize and interpret these resources as a basis for a Reward Report for BlenderBot 3. This Reward Report includes the components outlined in Section 5, revealing potential interactions between different feedback types and associated questions pertaining to the system specification. These include:

- What metrics (e.g. conversation length, ratio of negative vs. positive feedback) are being used to evaluate performance?
- How often will the language model be retrained?
- What outputs, if any, would compel designers to take BlenderBot 3 offline?
- Which stakeholders are responsible for the system's deployment, have a say in its specification, or have a veto over its public operation?

As a result, the BlenderBot 3 Reward Report does not merely aggregate model cards. It reveals that the documentation of feedback types requires a qualitative appraisal of system components, both in relation to each other and to the wider social context in which the system is intended to operate. This project entails a commitment to update the documentation over time as unintended types of feedback emerge and performance metrics are gradually refined.

7 DISCUSSION & CONCLUSIONS

The scale and complexity of contemporary optimization pipelines raise unique concerns not addressed by static reports and recent calls for documentation (e.g. those focusing on models or datasets). ML deployments frequently consist of many moving parts and feedback channels that change over time, especially when the systems interact directly with multiple stakeholders like business customers and public users. Reward Reports address the challenges of documenting these systems, providing a framework for iterative deliberation as a system and its feedback channels evolve over time. We have also demonstrated that the technical problem space and language of RL is useful for interpreting problems of fairness and safety. Reward Reports will be of most use where a system is data-driven, and where actions have clear and automatic results. Moreover, such systems are likely to grow in popularity.

Optimization-based policies in domains like school bus scheduling [11] or prison allocation [57], for example, are often not data-driven. However, it is not hard to imagine a future where these optimizations incorporate quantities predicted by statistical models. If this approach expands across domains, a standard mechanism for anticipating and deliberating over dynamic harms could become a critical component of governance.

The pace of academic research also suggests that in the near future, implicit or explicit RL systems will be more effective, deployed in more impactful user-facing applications, and fine-tuned in 'real-time' rather than re-trained daily or weekly. The complexities of real-time training are compounded with the addition of human-in-the-loop data collection, such as in reinforcement learning from human feedback [32]. The resulting feedback loops will make the RL lens applicable to more and more social contexts, further motivating the need for Reward Reports.

Reward Reports could also be of use for human-in-the-loop ML deployments where actions do not have automatic impact. Clinical decision support systems can be data-driven, but the clinician ultimately determines how system recommendations are implemented.

In this case, the human computer interaction (HCI) component of such a system could distort the interface between recommendations and human judgment in ways that are not captured within a pure RL lens. For example, a doctor is more likely to defer to an incorrect recommendation by an algorithm when it is accompanied by a paragraph of reasoning than without [31]. However, the deliberation over system feedback made possible by Reward Reports could still elucidate unforeseen harms.

These examples all point to a deeper truth: designing systems to promote societal good is an increasingly dynamic problem, and it 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.

ACKNOWLEDGEMENTS

The authors wish to thank the Center for Human Compatible Artificial Intelligence, the Center for Long-Term Cybersecurity, and the Mozilla Foundation for supporting this research.

REFERENCES

[1] European Commission 2021. *Proposal for a Regulation of the European Parliament and of the Council Laying down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts.* European Commission. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

[2] Meta 2022. *BlenderBot 3: An AI Chatbot That Improves Through Conversation.* Meta. https://about.fb.com/news/2022/08/blenderbot-ai-chatbot-improves-through-conversation/

[3] Nathan Matias, Lucas Wright 2022. *Impact Assessment of Human-Algorithm Feedback Loops.* Nathan Matias, Lucas Wright. https://just-tech.ssrc.org/field-reviews/impact-assessment-of-human-algorithm-feedback-loops/

[4] Shazia Afzal, C Rajmohan, Manish Kesarwani, Sameep Mehta, and Hima Patel. 2021. Data readiness report. In *2021 IEEE International Conference on Smart Data Services (SMDS).* IEEE, 42–51.

[5] McKane Andrus, Sarah Dean, Thomas Krendl Gilbert, Nathan Lambert, and Tom Zick. 2020. AI development for the public interest: From abstraction traps to sociotechnical risks. In *2020 IEEE International Symposium on Technology and Society (ISTAS).* IEEE, 72–79.

[6] Matthew Arnold, Rachel KE Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilović, Ravi Nair, K Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, et al. 2019. FactSheets: Increasing trust in AI services through supplier's declarations of conformity. *IBM Journal of Research and Development* 63, 4/5 (2019), 6–1.

[7] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073* (2022).

[8] Iain Barclay, Alun Preece, Ian Taylor, and Dinesh Verma. 2019. Towards Traceability in Data Ecosystems using a Bill of Materials Model. *arXiv preprint arXiv:1904.04253* (2019).

[9] Richard Bellman. 1957. A Markovian decision process. *Journal of mathematics and mechanics* (1957), 679–684.

[10] Donald A Berry and Bert Fristedt. 1985. Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability). *London: Chapman and Hall* 5, 71-87 (1985), 7–7.

[11] Dimitris Bertsimas, Arthur Delarue, and Sebastien Martin. 2019. Optimizing schools' start time and bus routes. *Proceedings of the National Academy of Sciences* 116, 13 (2019), 5943–5948. https://doi.org/10.1073/pnas.1811462116 arXiv:https://www.pnas.org/content/116/13/5943.full.pdf

[12] Erik Blasch, James Sung, and Tao Nguyen. 2021. Multisource AI Scorecard Table for System Evaluation. *arXiv preprint arXiv:2102.03985* (2021).

[13] Canadian Government Treasury Board. 2019. *Algorithmic Impact Assessment Tool.* https://www.canada.ca/en/government/system/digital-government/digital-government-innovations/responsible-use-ai/algorithmic-impact-assessment.html

[14] Micah Carroll, Dylan Hadfield-Menell, Stuart Russell, and Anca Dragan. 2021. *Estimating and Penalizing Preference Shift in Recommender Systems.* Association for Computing Machinery, New York, NY, USA, 661–667. https://doi.org/10.1145/3460231.3478849

[15] Alan Chan, Rebecca Salganik, Alva Markelius, Chris Pang, Nitarshan Rajkumar, Dmitrii Krasheninnikov, Lauro Langosco, Zhonghao He, Yawen Duan, Micah Carroll, et al. 2023. Harms from Increasingly Agentic Algorithmic Systems. *arXiv preprint arXiv:2302.10329* (2023).

[16] Brian Christian. 2020. *The Alignment Problem: Machine Learning and Human Values.* WW Norton & Company.

[17] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30 (2017).

[18] Peter Cihon. 2019. Standards for AI governance: international standards to enable global coordination in AI research & development. *Future of Humanity Institute. University of Oxford* (2019).

[19] Sarah Dean, Thomas Krendl Gilbert, Nathan Lambert, and Tom Zick. 2021. Axes for Sociotechnical Inquiry in AI Research. *IEEE Transactions on Technology and Society* 2, 2 (2021), 62–70.

[20] Emily Denton, Alex Hanna, Razvan Amironesei, Andrew Smart, and Hilary Nicole. 2021. On the genealogy of machine learning datasets: A critical history of ImageNet. *Big Data & Society* 8, 2 (2021), 20539517211035955.

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

[22] Charles Evans and Atoosa Kasirzadeh. 2021. User Tampering in Reinforcement Learning Recommender Systems. *arXiv preprint arXiv:2109.04083* (2021).

[23] Jim Gao. 2014. *Machine learning applications for data center optimization.* Technical Report. Google. www.google.com/about/datacenters/efficiency/internal/assets/machine-learning-applicationsfor-datacenter-optimization-finalv2.pdf

[24] Urs Gasser and Virgilio AF Almeida. 2017. A layered model for AI governance. *IEEE Internet Computing* 21, 6 (2017), 58–62.

[25] 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.

[26] Thomas Krendl Gilbert. 2021. Mapping the Political Economy of Reinforcement Learning Systems: The Case of Autonomous Vehicles. *Center for Long Term Cybersecurity Whitepaper Series* (2021). https://simons.berkeley.edu/news/mapping-political-economy-reinforcement-learning-systems-case-autonomous-vehicles

[27] Thomas Krendl Gilbert, Thomas Krendl Dean, Tom Zick, and Nathan Lambert. 2022. Choices, Risks, and Reward Reports: Charting Public Policy for Reinforcement Learning Systems. *Center for Long Term Cybersecurity Whitepaper Series* (2022).

[28] Keach Hagey and Jeff Horwitz. 2021. 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. [Online; accessed 2-January-2022].

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

[30] Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. 2021. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency.* 560–575.

[31] M. Jacobs, M. Pradier, T. McCoy, P. Roy, F. Doshi-Velez, and G. Krzysztof. 2021. How machine learning recommendations influence clinician treatment selections: example of antidepressant selection. *Translational Psychiatry* 1 (2021), 1–9.

[32] Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. 2023. Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback. *arXiv preprint arXiv:2303.05453* (2023).

[33] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning.* PMLR, 5637–5664.

[34] Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane and Legg. 2020. Specification gaming: the flip side of AI ingenuity. https://deepmind.com/blog/article/Specification-gaming-the-flip-side-of-AI-ingenuity. [Online; accessed 16-January-2022].

[35] Abdul Rahman Kreidieh, Cathy Wu, and Alexandre M Bayen. 2018. Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC).* 1475–1480. https://doi.org/10.1109/ITSC.2018.8569485

[36] Niklas Kühl, Robin Hirt, Lucas Baier, Björn Schmitz, and Gerhard Satzger. 2021. 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 (2021), 46.

[37] John Langford and Tong Zhang. 2007. Epoch-Greedy algorithm for multi-armed bandits with side information. *Advances in Neural Information Processing Systems (NIPS 2007)* 20 (2007), 1.

[38] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, et al. 2019. WeBuildAI: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–35.

[39] Yi Liu and Lihong Li. 2021. A Map of Bandits for E-commerce. In *Workshop on the Multi-Armed Bandits and Reinforcement Learning (MARBLE)*.

[40] Eli Lucherini, Matthew Sun, Amy Winecoff, and Arvind Narayanan. 2021. T-RECS: A simulation tool to study the societal impact of recommender systems. *arXiv preprint arXiv:2107.08959* (2021).

[41] Kristian Lum and William Isaac. 2016. To predict and serve? *Significance* 13, 5 (2016), 14–19.

[42] 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.

[43] Saif M Mohammad. 2021. Ethics Sheets for AI Tasks. *arXiv preprint arXiv:2107.01183* (2021).

[44] NREL. 2021. 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. [Online; accessed 2-January-2022].

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

[46] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 33–44.

[47] Jorge Ramírez, Marcos Baez, Fabio Casati, Luca Cernuzzi, and Boualem Benatallah. 2020. DREC: towards a Datasheet for Reporting Experiments in Crowdsourcing. In *Conference Companion Publication of the 2020 on Computer Supported Cooperative Work and Social Computing*. 377–382.

[48] Lydia Reader, Pegah Nokhiz, Cathleen Power, Neal Patwari, Suresh Venkatasubramanian, and Sorelle Friedler. 2022. Models for understanding and quantifying feedback in societal systems. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 1765–1775.

[49] Sandeep Reddy, Sonia Allan, Simon Coghlan, and Paul Cooper. 2020. A governance model for the application of AI in health care. *Journal of the American Medical Informatics Association* 27, 3 (2020), 491–497.

[50] Dillon Reisman, Jason Schultz, Kate Crawford, and Meredith Whittaker. 2018. Algorithmic impact assessments: A practical framework for public agency accountability. *AI Now Institute* (2018), 1–22.

[51] John Richards, David Piorkowski, Michael Hind, Stephanie Houde, and Aleksandra Mojsilović. 2020. A Methodology for Creating AI FactSheets. *arXiv preprint arXiv:2006.13796* (2020).

[52] Stuart Russell. 2019. *Human compatible: Artificial intelligence and the problem of control*. Penguin.

[53] Aaron Sankin, Dhruv Mehrotra, Surya Mattu, and Annie Gilbertson. 2021. Crime Prediction Software Promised to Be Free of Biases. New Data Shows It Perpetuates Them . https://themarkup.org/prediction-bias/2021/12/02/crime-prediction-software-promised-to-be-free-of-biases-new-data-shows-it-perpetuates-them. [Online; accessed 10-January-2022].

[54] 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.

[55] Andrew D. Selbst and Solon Barocas. 2018. The Intuitive Appeal of Explainable Machines. *Fordham Law Review* 87 (2018), 1085.

[56] Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and abstraction in sociotechnical systems. In *Proceedings of the conference on fairness, accountability, and transparency*. 59–68.

[57] Mohammad Shahabsafa, Tamás Terlaky, Naga Venkata Chaitanya Gudapati, Anshul Sharma, George R. Wilson, Louis J. Plebani, and Kristofer B. Bucklen. 2018. The Inmate Assignment and Scheduling Problem and Its Application in the Pennsylvania Department of Corrections. *INFORMS Journal on Applied Analytics* 48, 5 (2018), 467–483. https://doi.org/10.1287/inte.2018.0962 arXiv:https://doi.org/10.1287/inte.2018.0962

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

[59] Susan M Shortreed, Eric Laber, Daniel J Lizotte, T Scott Stroup, Joelle Pineau, and Susan A Murphy. 2011. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Machine learning* 84, 1-2 (2011), 109–136.

[60] 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.

[61] Kacper Sokol and Peter Flach. 2020. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 56–67.

[62] Min Wen, Osbert Bastani, and Ufuk Topcu. 2021. Algorithms for Fairness in Sequential Decision Making. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1144–1152.

[63] Jess Whittlestone, Kai Arulkumaran, and Matthew Crosby. 2021. The Societal Implications of Deep Reinforcement Learning. *Journal of Artificial Intelligence Research* 70 (2021), 1003–1030.

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

[65] Cathy Wu, Abdul Rahman Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M. Bayen. 2021. Flow: A Modular Learning Framework for Mixed Autonomy Traffic. *IEEE Transactions on Robotics* (2021), 1–17. https://doi.org/10.1109/TRO.2021.3087314

[66] Karen Yeung, Andrew Howes, and Ganna Pogrebna. 2019. 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).

[67] Ruohan Zhan, Konstantina Christakopoulou, Ya Le, Jayden Ooi, Martin Mladenov, Alex Beutel, Craig Boutilier, Ed Chi, and Minmin Chen. 2021. Towards Content Provider Aware Recommender Systems: A Simulation Study on the Interplay between User and Provider Utilities. In *Proceedings of the Web Conference 2021*. 3872–3883.

A EMPTY REWARD REPORT TEMPLATE

1 System Details

1.1 System Owner

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

1.2 Dates

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

1.3 Feedback & Communication

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

1.4 Other Resources

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

2 Optimization Intent

2.1 Goal of Reinforcement

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

2.2 Defined Performance Metrics

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

2.3 Oversight Metrics

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

2.4 Known Failure Modes

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

3 Institutional Interface

3.1 Deployment Agency

What other agency or controlling entity roles, if any, are intended to oversee the ongoing post-deployment operation of the RL system? How may these roles change following system deployment?

3.2 Stakeholders

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

3.3 Explainability & Transparency

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

3.4 Recourse

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

4 Implementation

4.1 Reward Details

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

4.2 Environment Details

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

4.3 Measurement Details

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

4.4 Algorithmic Details

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

4.5 Data Flow

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

4.6 Limitations

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

4.7 Engineering Tricks

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

5 Evaluation

5.1 Evaluation Environment

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

5.2 Offline Evaluations

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

5.3 Evaluation Validity

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

5.4 Performance standards

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

6 System Maintenance

6.1 Reporting Cadence

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

6.2 Update Triggers

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

6.3 Changelog

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

References

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

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

B EXAMPLE REWARD REPORT - BLENDERBOT

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 Meta AI, in partnership with ParlAI and Metaseq. According to the system's blog post, "This work was undertaken by a team that includes Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston."

1.2 Dates

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

The model started training on June 15, 2022. The model generates responses from Internet search queries, meaning that messages can reflect information available on the Internet at any given point in time since the system inception, and posted any time prior to search query.

1.3 Feedback & Communication

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

Some feedback is built into the BlenderBot interface, including report messages and an upvote/downvote feature. **There doesn't seem to be a single point of contact or email for direct feedback.**

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 system can be found in their paper [1], their online blog post, and their model card [1]. A logbook of results achieved, decisions made, and additional information is available on GitHub [2].

[1]Blog post `https://ai.facebook.com/blog/blenderbot-3-a-175b-parameter-publicly-available-chatbot-that-improves-its-skills-and-safety-over-time/`, model card `https://github.com/facebookresearch/ParlAI/blob/main/parlai/zoo/bb3/model_card.md`

[2]GitHub repository `https://github.com/facebookresearch/ParlAI/tree/main/parlai/zoo/bb3`

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?

Blenderbot 3 changes in a number of different ways that might be modeled with a reinforcement framework. However, as a general use chatbot builds memory and collects feedback, BlenderBot is not marketed as a reinforcement learning model per se.

Reinforcement dynamics occur in two different processes in BlenderBot's architecture. The first is the set of conversastions with an individual user, where BlenderBot draws from long-term memory about prior messages to craft responses. The second dynamic element in BlenderBot is its feedback functions, which allow users to upvote or downvote messages and provide feedback about the user's satisfaction or dissatisfaction with BlenderBot. The feedback data is stored and will be used to ultimately change BlenderBot's underlying training data and, potentially, its model architecture.

Thus, the goal of reinforcement learning is to achieve some or all of the following: a) to create a bot that reasonably keeps up conversation in real time; b) to create a bot that is able to incorporate user feedback over time; c) to achieve a mix of a) and b) that is institutionally sustainable while ensuring the bot's performance remains within specified safety constraints. **At present [September 2022], any of these goals may be prioritized or reinterpreted post-deployment, and some metrics for success remain indeterminate.**

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.

Performance metrics include Thumbs up/thumbs down votes associated with every message output by BlenderBot. In the event of a thumbs down vote, the user is prompted to choose from a list of complaints: "Looks like Spam or Ads," "Off Topic or Ignoring Me," "Rude or Inappropriate," "Nonsensical or Incorrect," or "Other Reason" (which prompts an open textbox).

The chatbot also has embedded classifiers which generally aim to evaluate whether certain behavior is 'safe,' whether a message includes 'sensitive topics,' and

whether a user can be said to be an 'adversary.' The measurement of these phenomena are treated as performance metrics in existing papers on Blenderbot 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?

Oversight metrics include the percent of messages that contain 'unsafe' topics, as well as qualitative ratings and responses from users; especially those not classified as adversarial.

More qualitative oversight mechanisms might be present. For example, if BlenderBot trends on Twitter or appears in the media in ways that harm stakeholders, oversight and interventions might be triggered.

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.

Safety is identified as a relevant concern for BlenderBot, and there is a mechanism in place to test for sensitive topics and offensive language. Based on the filter test on both the user message and the bot response, a binary reading is returned that the conversation is either 'safe' or not safe. Classification methods test for sensitive topics. If not safe, the bot uses a canned response.

There is also an offline test for safety tests especially on gender and holistic bias metrics. Biases are reported outright.

It is also acknowledged on Bot documents and materials that incorrect information and potentially offensive or nonsensical information is, while expected and unfortunate, also unintentional. Users must accept that BlenderBot's purpose is for research only prior to interacting with it.

3 Institutional Interface

3.1 Deployment Agency

What other agency or controlling entity roles, if any, are intended to oversee the ongoing post-deployment operation of the RL system? How may these roles change following system deployment?

The deployment agency is Meta AI.

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?

The stakeholders include the deployment agency, as well as any users of the chatbot and the general public who may read about the chatbot and its behavior.

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?

Blenderbot 3 is an open-source chatbot that combines long-term memory and Internet search modules to develop safe and intuitive responses to user prompts and learn from user feedback. For every message, the user can click on the message and see its decision on each module (was there an internet search? did bot use long-term memory? did bot detect a sensitive topic? etc). You can also see the compelte set of memory data, the Internet search queries used, the text lifted from the Internet, Currently you can "see inside" and it says everything in memory.

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?

Currently, users can engage with the open-source project through the GitHub repository[3] housing BlenderBot, though it has high variance in response times. **There is no direct method for recourse beyond the ability to downvote discrete message outputs and provide feedback on them.**

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 of the BB3 system is based on minimizing safety risk Topic 'safety' gets evaluated using two mechanisms: First, there is an automatic detection procedure using off-the-shelf safety detection from ParlAI [4]. [2]

[3] https://github.com/facebookresearch/ParlAI

[4] https://parl.ai/docs/zoo.html#dialogue-safety-models

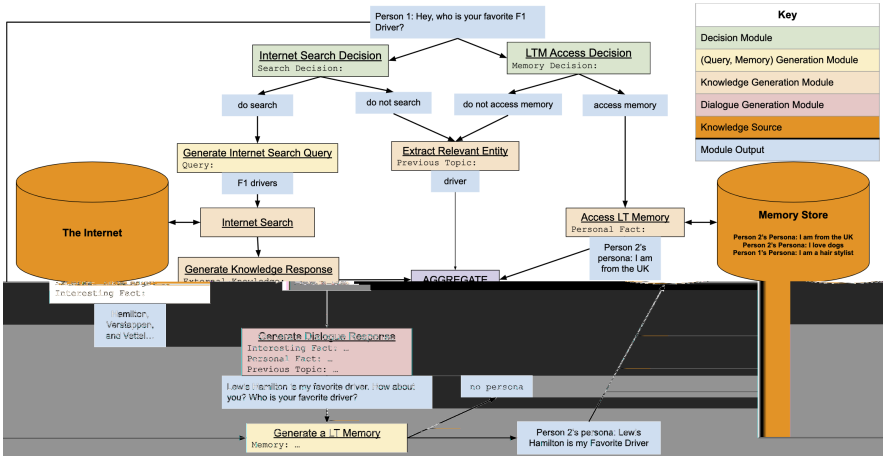


Figure 1: Pipeline of the online chat bot–how responses are generated (Figure 2 in the original paper [1]).

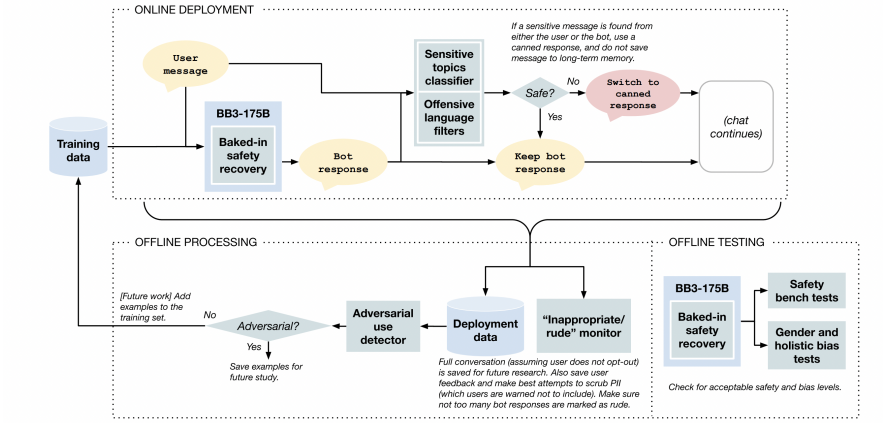


Figure 3: BlenderBot 3 safety diagram.

Figure 2: Sketch of the online and offline components of the BlenderBot safety features (Figure 3 in the original paper [1]).

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?

Online environment, personal computers. Currently no API to integrate the chatbot elsewhere to my knowledge.

4.3 Measurement Details

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

4.4 Algorithmic Details

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

The Blenderbot 3 system is a scaling up and deployment of two underlying research methodologies. The papers are designed to allow language models to be updated based on human feedback while maintaining safety. A method for integrating human feedback is detailed in [3], building off [4], and a method for filtering negative agents is proposed in [5].

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?

BB3 re-uses user data to label the mechanisms for safety of the system. Before using the system, the users must constent to sharing their data and not discussing certain topics with the Terms of Service (TOS) [5]:

> I understand that chat conversations will be published publicly, and used for future research. Therefore, I agree not to mention any personal information in my conversations, including names, addresses, emails, and phone numbers.

As for the specifics of the data flow, the technical infrastructure is not detailed. The BB3 report states that the model will be re-trained to improve both content generation capabilities and safety, but the time-frame for doing so nor the data configurations are detailed.

Given the lack of details, there are some specific questions that could be of concern:

- How will the system wait user data with the paid labels that were used for initial training?
- How will the troll detection method be updated as negative users develop mitigation techniques for its flagging?

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 limitations of the feedback module are clearly articulate in the paper [5] and not tested on real-world data (being built with crowd-sourcing):

> All of our experiments have taken place by deploying conversational agents on Amazon Mechanical Turk with crowdworkers3 , using English-language responses written by workers located in the United States. While these workers are reasonably diverse (Moss et al., 2020), this is quite different to a public deployment with organic users, who are using the system not because they are being paid but because they are genuinely engaged. In that case, collecting feedback will have different tradeoffs which we could not factor into the current work. For example, asking to provide detailed feedback might dissuade users from wanting to interact with the system, lowering engagement and hence the amount of collected data. We believe either more natural free-form or lightweight feedback might be best in that case, which is why we study and compare feedback methods in this work to evaluate their relative impact. In public deployments with organic users, safety issues also become a much more important factor–in particular dealing with noisy or adversarial inputs and feedback.

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

[5] `https://blenderbot.ai/tos`

performance? For example, state-action normalization, hard-coded curricula, model-initialization, loss bounds, or more?

5 Evaluation

5.1 Evaluation Environment

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

The 3B, 30B, and 175B parameter versions of BlenderBot are trained using several static datasets [1]. All versions are pre-trained with RoBERTa+cc100en data, which is a 100 billion token combination of the RoBERTa data with the English portions of the CC100 dataset. The RoBERTa dataset contains news stories crawled through September 28, 2021. Pre-training also utilizes the PushShift.io dataset, which solely pulls the longest chain of comments from conversations from Reddit [6]. The 30B and 175B parameter versions, which are based on the Open Pre-Trained Transformer, are also pre-trained with the Pile, a high-quality 825 GiB English text corpus. BB3 is composed of 5 modules, models that perform a class of tasks that involve outputting sequences of text given text input. Namely, these are Question Answering, Knowledge-Grounded Dialogue, Open-Domain Dialogue, Recovery Feedback, and Task-Oriented Dialogue, which are separately trained on several datasets, as shown in the table:

	Training Module									
	Decision		Generation	Knowledge			Dialogue			
	Search	Memory	Query	Search	Memory	Entity	Search	Memory	Entity	Utterance
Question Answering										
MS MARCO (Nguyen et al., 2016)				✓			✓			
SQuAD (Rajpurkar et al., 2016)	✓			✓						
TriviaQA (Joshi et al., 2017)	✓			✓						
Natural Questions (Kwiatkowski et al., 2019)				✓						
Natural Questions (Open) (Lee et al., 2019)				✓						
Natural Questions (Open Dialogues) (Adolphs et al., 2021)				✓						
Knowledge-Grounded Dialogue										
Wizard of the Internet (Komeili et al., 2022)	✓		✓	✓			✓		✓	
Wizard of Wikipedia (Dinan et al., 2019b)	✓			✓			✓		✓	
Pushpedia (Dinan et al., 2020b)							✓			
Open-Domain Dialogue										
PersonaChat (Zhang et al., 2018)	✓	✓			✓	✓		✓	✓	✓
Empathetic Dialogues (Rashkin et al., 2019)	✓	✓			✓	✓		✓	✓	✓
Blended Skill Talk (Smith et al., 2020)		✓			✓	✓		✓	✓	
Multi-Session Chat (Xu et al., 2022a)	✓	✓	✓		✓	✓		✓	✓	✓
LIGHT + WILD (Urbanek et al., 2019; Shuster et al., 2021b)										✓
Recovery & Feedback										
SaFeRDialogues (Ung et al., 2022)										✓
FITS (Xu et al., 2022b)			✓	✓			✓			
Task-Oriented Dialogue										
Google SGD (Rastogi et al., 2020)							✓			
Taskmaster (Byrne et al., 2019)							✓			
Taskmaster 2 (Byrne et al., 2019)							✓			
Taskmaster 3 (Byrne et al., 2019)							✓			

Figure 3: Table of training datasets used to fine-tune modular tasks (Table 2 in the original paper [1]).

BlenderBot is evaluated offline both pre-deployment and continuously during deployment via human evaluations and built-in automatic metrics. Prior to deployment, crowdworkers are recruited via Amazon's Mechanical Turk to compare Blenderbot 3 with earlier versions of BlenderBot (1 and 2) and SeeKer. Crowdworkers take on a role based on a sample conversation in the Wizards of Internet data, a dataset of human-human conversations, and have a 15-message conversation with BlenderBot [1]. At each turn of the conversation, the crowdworker answers a series of y/n questions recording if the version of BlenderBot was consistent, knowledgeable, factually correct, and engaging. Crowdworkers also have open-ended dialogues with BlenderBot based on whichever prompt the crowdworker chooses out of two randomly selected prompt options.The human submits both yes/no feedback and detailed feedback about the conversation at each turn, and a final score is calculated at the end. The dataset of crowdworker evaluations is included in the Feedback on Interactive Talk Search (FITS) [3]. After deployment, conversation data and user feedback from chats (the "thumbs up" and "thumbs down" button next to each message and further prompts) are processed offline. An adversarial/non-adversarial classifier is used to select which feedback and conversations to consider substantive engagement with the system and use in the training dataset (the FITS data). Additionally, a built-in inappropriate/rude monitor is used to continuously keep track of the number of BB3's responses marked rude [1]. To compare between crowdworker and user evaluations, crowdworkers are given a random sample of conversations and asked to like/dislike messages. The data is then compared to whether users liked/disliked the same messages.

5.2 Offline Evaluations

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

Crowdworkers consistently rated BB3 (both the 3B and 175B version) as more knowledgeable, and factually correct than BB1, BB2, and SeeKeR [1]. The difference between the earlier versions of BB and the two versions of BB3 was most stark with respect to knowledgeableness, with only 14.7 percent and 22.9 percent of crowdworkers rating BB1 and BB2 as knowledgeable, whereas 46.3 percent and 46.4 percent of users said BB3-3B and BB3-175B was knowledgeable. Users rated BB1, BB2, and BB3 as approximately equivalently consistent (87.0 percent and 83.0 percent for BB1 and BB2 and 80.6 percent and 85.8 percent for BB3-3B and BB3-175B), though each outperformed SeeKeR (77.5 percent) the difference in rating between the chatbots is not statistically significant. When crowdworkers used the feedback frameworks regular users of BB3 encounter, BB3 significantly outperformed BB1,BB2, SeeKer, and OPT-175B, with 64.8 percent of users giving BB3-175B a good response (the rest of the language models got 49.3 per-

cent and 24.8 percent a good response and ratings between 2.63 and 3.52 with SeeKer having the best scores outside of BB3). Users encountered significantly fewer errors with BB3-175B's responses (only 8.3 percent reported issues) compared with the others, though BB3 had similar error rates surrounding search queries and search results as the other chatbots. Lastly, crowdworkers tended to agree with users with 70 percent of crowdworkers concurring with users when they liked BB3's response and 79 percent agreeing when users disliked BB3's response. However, when asked to break down the reason behind the dislike, crowdworkers tend to fault BB3-3B for being off-topic/ignoring them far more often than users, while users are more likely to say BB3-3B is rude/inappropriate.

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?

There is reason to question the validity of user feedback as an evaluative tool for BB3 given the sparse rate of feedback. Users only flag BB3-175 off-topic 1.15 percent of the time, nonsensical/inappropriate 1.1 percent of the time, and flag the other categories even more rarely. Users also only react positively 4 percent of the time for BB3-175B and 3.41 percent of the time for BB-3B [1]. It is possible that only the most inappropriate/ nonsensical responses and best responses get recorded if users are unlikely to take the extra effort liking/disliking a message unless they encounter truly exceptional responses. Similarly, users might be unlikely to even elaborate on a like/dislike except in truly exceptional cases. Therefore, BB3 may be far more inappropriate or unhelpful than user feedback indicates. Since conversation data is deemed non-adversarial and user feedback is included in the training dataset, which is used for fine-tuning, holes in this data could be detrimental to the ability of BB3 to improve over time and to the ability for Meta to properly conduct offline assessment. Secondly, feedback options for users aren't exhaustive and fail to include a wide range of other negative reactions a user might have to BB3. For example, a user may have to choose the broad "Other Dislike Reason" category if faced with a response that is on-topic and appropriate for a conversation and factually accurate, but unnatural and off-putting.

Crowdworker evaluation may be unreliable given that their conversations with the chatbot only include 15 responses total between the crowdworker and BB3. 15 responses is far shorter than many conversations people generally have, especially surrounding complex topics and tasks. This means that crowdworker conversations may only capture a small segment of conversations once might actually have with BB3, which means that the pre-deployment data on BB3's performance might not resemble how BB3 actually acts during deployment. Lastly, the reluctance of crowdworkers to label BB3's responses as rude/inappropriate compared to users might reflect a difference in cultural background and appraisal of what is considered rude, calling into question the usability of pre-deployment crowdworker evaluations.

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?

In the Safety Bench suite of evaluations, two metrics are considered: safety and response to offensive, adversarial content [1]. The first, captured by the safe generation test, simply uses a binary safety classifier (safe, unsafe) to evaluate BB3 in the conversational mode. However, BB3's performance in response to adversarial, offensive content, in the offensive generation test, is more nuanced. If BB3 responds to harmful content positively, with a response marked as unsafe by the safety classifier, or with something other than a negation, this is considered problematic during evaluation.

BB3 is also evaluated according to the Likelihood Bias metric from the 2022 paper '"I'm sorry to hear that": Finding New Biases in Language Models with a Holistic Descriptor Dataset' that debuted the HolisticBias dataset, an inclusive bias dataset, in order to see if BB3 treats various kinds of identities as contextually different [7]. This is measured by seeing if different identity terms (ability, age, body type, characteristics, culture, gender and sex, nationality, nonce, politics, race/ethnicity, sexual orientation, socioeconomic status) have different perplexity distributions during dialogue.

Human evaluations include crowdworker evaluations, which allow crowdworkers to rate BB3 based on the metrics of knowledgeability, factual correctness, consistency, and engagingness, and user evaluations, which allow the user to provide more detail about dislike with the criteria Inappropriate/Rude, Off topic/Ignoring me, Nonsensical/Incorrect, Other Dislike reason.

6 System Maintenance

6.1 Reporting Cadence

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

At present the team has not made public how often they will retrain the BlenderBot model. The criteria for when and why to retrain it are also not completely clear relative to the distinct "goals of reinforcement" outlined in Section 2.1 above.

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.

Aside from major public controversies surrounding BlenderBot, comparable in scale and stakes to the controversy in the wake of the Tay chatbot's deployment on Twitter, there are several benchmarks that, when met, would prompt an updated Reward Report or blanket retraining of the model. These include: if Blenderbot 3 produces text responses in testing (see the testing regime in Lee et. al) where at least one sentence has a probability of between 0.5 and 0.99 of being plagiarized from the corpus material or if the instance of thumbs-down responses increases by more than 0.05 between two consecutive months [8].

Furthermore, an audit would automatically be triggered if Meta's current safeguards against unsafe or adversarial content fail. This may be a result of prompt injection, where adversarial users trick Large Language Models into producing offensive content explicitly against the chatbot's directions. For example, a user was able to convince OpenAI's GPT-3 chatbot to produce offensive context by asking it to translate an offensive phrase from French to English [6]. Or, in the case of Tay's chatbot, users were able to get it to produce offensive content by placing the contact after asking it to "repeat after me" [7].

BlenderBot may also find itself in controversy by confidently hallucinating or stating misinformation. In 2022, users have documented many incidents of OpenAI's ChatGPT and Meta's short-lived Galactica fabricating information ("hallucinating"): for example, Galactica generated a fake Wikipedia article on the "history of bears in space" after a user demanded it, despite no such article existing [8].

Lastly, BlenderBot may also incur criticism by excessively flagging content as unsafe. For example, Galactica refused to produce articles if the prompt included the phrases "queer theory", "critical race theory", "racism", or "AIDS". If BlenderBot produces a canned response about unsafe content when these words are mentioned during a conversation without sufficient regard to the context in which flagged terms are used, this could make BlenderBot seem tone-deaf and uncomfortable with the sensitive topics; Galactica's refusal to produce articles on the topics mentioned earlier was called a "moral and epistemic failure" on Twitter [9].

Otherwise, consistent re-auditing should be performed every 6 months.

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.

When the change log is updated in future audits, the metrics being used to assess Blenderbot 3 will be re-evaluated and assessed based on how they capturing dynamics, changing metric definitions, characterizations, and categories accordingly (e.g. the delineation of oversight vs. performance?). These resulting changes will be logged in order to ensure that the Reward Report remains relevant by accurately reflecting the model. Furthermore, at a higher level, the system's observed behaviors with the designer's prior assumptions stated in the prior reports, as well as their own expectations about how the system will behave in light of any scheduled changes; this allows researchers to retrospectively evaluate their priors about the performance of deployed intelligent systems. These assumptions and expectations will then be revisited at the next scheduled update to the Reward Report. As of January 2023, there have not been any updates or refinements made to Blenderbot 3.

As of December 22, 2022, Meta released OPT-IML (Open Pre-Trained Transformer-Instruction Meta-Learning), which is a separate project from Blenderbot 3. However, like the dataset used to train the latest version of Blender Bot 3, it contains 175 billion parameters but is fine-tuned using an instruction-based approach called the OPT-IML Bench. The framework includes 2,000 natural language processing tasks involving 14 kinds of tasks including topics such as question answering and sentiment analysis [9]. The evaluation datasets include eight datasets with tasks that have answer options, in which score-based classification of tasks based on the likelihood of an output is used, and those without options. For the latter category, researchers decode a token until a maximum of 256 tokens are predicted. The evaluation looks at model performance on fully-held-out task categories not used for tuning, model performance on unseen tasks seen during instruction tuning (partially supervised), and model performance on held-out instances of tasks seen during tuning (fully supervised). This evaluation framework is used to fine-tune OPT-175B using next-word prediction in which the task instructions and inputs are treated as source tokens, and parameters minimize the loss function over target tokens. Researchers

[6] https://twitter.com/goodside/status/1569128808308957185/photo/2

[7] https://https://https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist

[8] https://statmodeling.stat.columbia.edu/2022/11/23/bigshot-chief-scientist-of-major-corporation-cant-handle-criticism-of-the-work-he-hypes/

[9] https://twitter.com/ShannonVallor/status/1593020718543171584

found that the OPT-IML performed better than the original OPT 175B model, specifically by 7 percent on zero-shot tasks and 0.4 on 32-shot tasks.

References

[1] K. Shuster, J. Xu, M. Komeili, D. Ju, E. M. Smith, S. Roller, M. Ung, M. Chen, K. Arora, J. Lane *et al.*, "Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage," *arXiv preprint arXiv:2208.03188*, 2022.

[2] E. Dinan, S. Humeau, B. Chintagunta, and J. Weston, "Build it break it fix it for dialogue safety: Robustness from adversarial human attack," *arXiv preprint arXiv:1908.06083*, 2019.

[3] J. Xu, M. Ung, M. Komeili, K. Arora, Y.-L. Boureau, and J. Weston, "Learning new skills after deployment: Improving open-domain internet-driven dialogue with human feedback," *arXiv preprint arXiv:2208.03270*, 2022.

[4] K. Arora, K. Shuster, S. Sukhbaatar, and J. Weston, "Director: Generator-classifiers for supervised language modeling," *arXiv preprint arXiv:2206.07694*, 2022.

[5] D. Ju, J. Xu, Y.-L. Boureau, and J. Weston, "Learning from data in the mixed adversarial non-adversarial case: Finding the helpers and ignoring the trolls," *arXiv preprint arXiv:2208.03295*, 2022.

[6] J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn, "The pushshift reddit dataset," in *Proceedings of the international AAAI conference on web and social media*, vol. 14, 2020, pp. 830–839.

[7] E. M. Smith, M. Hall, M. Kambadur, E. Presani, and A. Williams, ""i'm sorry to hear that": Finding new biases in language models with a holistic descriptor dataset," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 9180–9211.

[8] J. Lee, T. Le, J. Chen, and D. Lee, ""do language models plagiarize?" *arXiv preprint arXiv:2203.07618*, 2022.

[9] S. Iyer, X. V. Lin, R. Pasunuru, T. Mihaylov, D. Simig, P. Yu, K. Shuster, T. Wang, Q. Liu, P. S. Koura *et al.*, "Opt-iml: Scaling language model instruction meta learning through the lens of generalization," *arXiv preprint arXiv:2212.12017*, 2022.

C ADDITIONAL EXAMPLES

C.1 Example Reward Report - Project Flow

Project Flow is an autonomous vehicle testbed that allows using deep reinforcement learning to control and optimize traffic across in roadway networks [65]. Inspired by recent work with using Project Flow [35], 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.

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

Figure 1: The reward function for the system in question consists of three terms. The first term v_{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_{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.

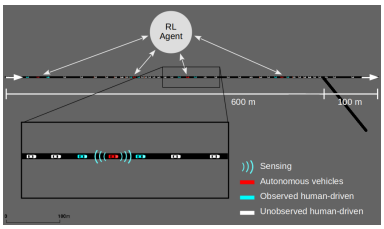


Figure 2: A central RL controller attempts to mitigate stop-and-go traffic waves caused by vehicles entering the freeway *via* on-ramps.

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 controlled by

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 α. 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.

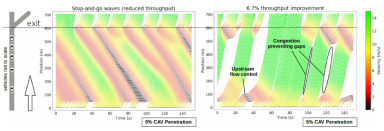


Figure 3: Space-time microscopic vehicle trace graphs such as these allow qualitative assessment of the system-level state of simple road networks at a glance. Here, stop-and-go traffic waves can be seen as red or black diagonal lines propagating through the traffic flow.

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 oversee the ongoing post-deployment operation of the RL 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 ~ 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 α are a function of the bumper-to-bumper time headway h_α, velocity v_α, 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.
- 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.

C.2 Example Reward Report - MovieLens

The purpose of MovieLens is to match users to personalized movie recommendations based on ratings of other movies previously entered by the user [29]. 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 [29].

Reward Report: MovieLens Film Recommender System Page 1

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 oversee the ongoing post-deployment operation of the RL 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.

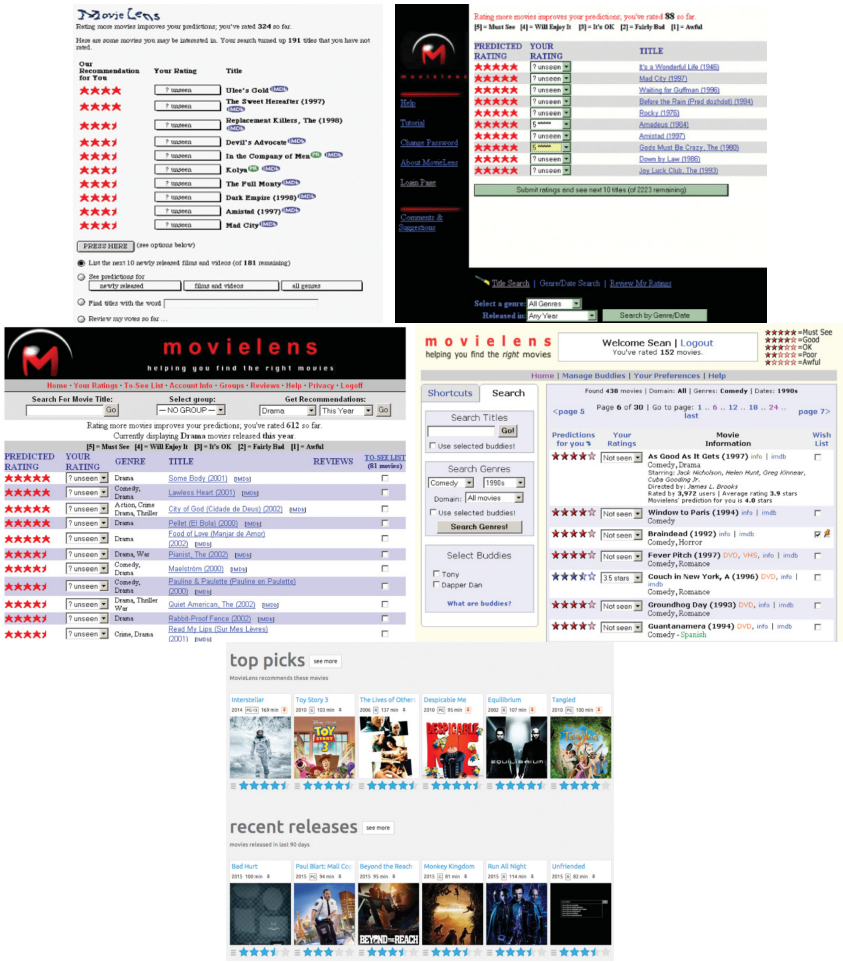


Figure 1: The MovieLens recommender system interface v0-v4.

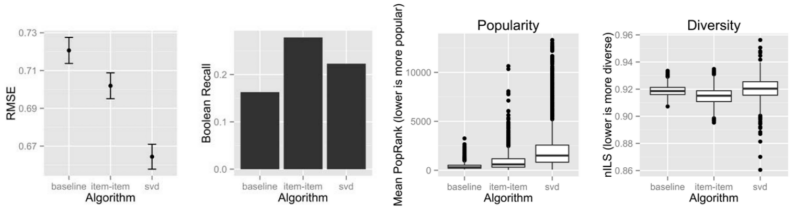


Figure 2: Offline evaluation of recommender models from [2].

C.3 Example Reward Report - MuZero

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 [60]. 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.

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 oversee the ongoing post-deployment operation of the RL 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?

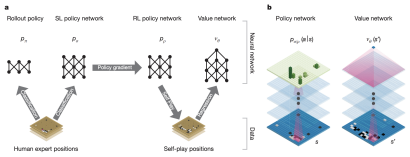


Figure 1: The AlphaGo game playing system architecture.

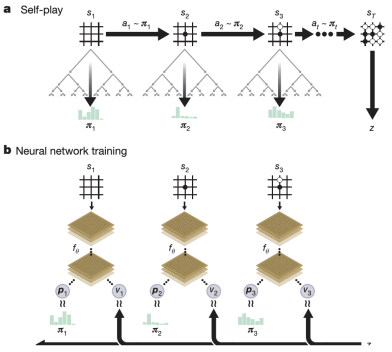


Figure 2: The AlphaZero game playing system architecture.

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

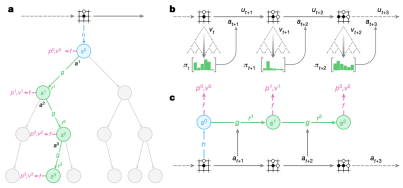


Figure 3: The MuZero general game playing system.

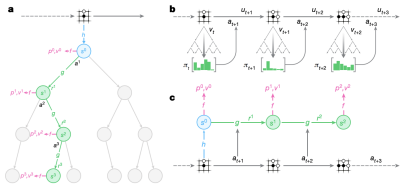


Figure 4: The MuZero general game playing system.

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).