SHARPIE: A Modular Framework for Reinforcement Learning and Human-AI Interaction Experiments

Anonymous authors
Paper under double-blind review

Keywords: Human-in-the-loop, Interactive RL, Hybrid Intelligence, Experimental Platforms

Summary

Reinforcement Learning (RL) offers a general approach for modeling and training AI agents, including human-AI interaction scenarios. In this paper, we propose SHARPIE (**S**hared **H**uman-**A**I **R**einforcement Learning **P**latform for **I**nteractive **E**xperiments) to address the need for a generic framework to support experiments with RL agents and humans. Its modular design consists of a versatile wrapper for RL environments and algorithm libraries, a participant-facing web interface, logging utilities, deployment on popular cloud and participant recruitment platforms. The platform is based on a generic interface for human-RL interactions that aims to standardize the field of study on RL in human contexts.

Contribution(s)

1. We present a first-of-its-kind, general-purpose, standardized open-source experimentation platform for research on human-centered reinforcement learning, bridging the gap in existing work by supporting simultaneous multi-agent and multi-human experimentation with configurable communication channels in various modalities.
   Context: Prior frameworks provide task-specific end solutions rather than configurable modules, and mainly support either multi-agent or multi-human scenarios, and restrict the type of possible interactions among them.
2. The framework is generic, modular and applicable to a wide range of RL algorithms, environments, and interaction paradigms at a limited additional implementation cost using only minimal wrapper extensions to Gymnasium environments. Moreover, it seamlessly integrates with crowdworker platforms.
   Context: The modularity allows using the framework for research on a wide range of research questions. Crowdworker platform integration has proven invaluable in non-RL machine learning domains but has so far seen lacking support in RL platforms involving humans.
3. We evaluate the flexibility and functional capabilities of the platform using four representative case studies that illustrate different ways humans can be integrated into the RL process: through evaluative feedback, action intervention, environment modification, and with natural language instructions.
   Context: These representative case studies validates the high compatibility of the versatile wrapper that our platform provides with diverse RL algorithms and environments.
4. We conduct several performance and scalability tests to demonstrate that our framework supports seamless distributed interaction across geographically distant human participants, with end-to-end latency within 0.21 seconds.
   Context: Low-latency communication is paramount to several real-time synchronized collaborative human-RL tasks, but challenging due to fundamental networking limitations.

SHARPIE: A Modular Framework for Reinforcement Learning and Human-AI Interaction Experiments

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement Learning (RL) offers a general approach for modeling and training AI agents, potentially including human-AI interaction scenarios. In this paper, we propose SHARPIE (**S**hared **H**uman-**A**I **R**einforcement Learning **P**latform for **I**nteractive **E**xperiments) to address the need for a generic framework to support experiments with RL agents and humans. Its modular design consists of a versatile wrapper for RL environments and algorithm libraries, a participant-facing web interface, logging utilities, deployment on popular cloud and participant recruitment platforms. It empowers researchers to study a wide variety of research questions related to the interaction between humans and RL agents, including those related to interactive reward specification and learning, learning from human feedback, action delegation, preference elicitation, user-modeling, and human-AI teaming. The platform is based on a generic interface for human-RL interactions that aims to standardize the field of study on RL in human contexts. To demonstrate ease-of-use and versatility, we replicate four illustrative experiments from the multi-agent RL literature and report on the platform's performance under stress testing to showcase its robustness.

1 Introduction

Reinforcement learning (RL) agents learn through environmental interaction to maximise cumulative long-term reward (Sutton & Barto, 1998). In recent years, this paradigm has expanded beyond pure autonomous discovery to include scenarios involving humans, under the umbrella term of Reinforcement Learning from Human Feedback (Kaufmann et al., 2025). As available compute increases, RL methods are being deployed in increasingly complex domains where humans are central to the loop. RL agents interact with humans in a wide variety of ways: in some problem settings, RL agents observe humans to achieve a common goal or act on their behalf (Natarajan et al., 2010); in other settings, RL agents communicate with humans to provide services or support their decision making Zhao et al. (2021); in other settings, human feedback can also be used to evaluate the value of agent behavior (Knox & Stone, 2009; Christiano et al., 2017), to communicate preferences (Bıyık et al., 2022), to correct behavior (Saunders et al., 2018), and to guide exploration (Guan et al., 2021; Torne et al., 2023).

These different types of interaction highlight the need to incorporate humans in the training and evaluation of RL agents for various aims including efficiency, safety, personalization and alignment purposes. However, implementing these experiments in a standardized manner presents specific infrastructural challenges. Connecting local RL training loops to human participants requires building interfaces, managing state synchronization, and implementing data logging services. Consequently, current research on RL involving humans frequently relies on custom, ad-hoc point-solutions engineered for individual studies. While popular RL libraries such as Gymnasium (Towers et al., 2024), PettingZoo (Terry et al., 2021), JaxMARL (Rutherford et al., 2024), and MO-Gymnasium (Alegre et al., 2022) provide established benchmarks in their related domains, experimentation involving

humans in RL has thus far seen a lack of standardization and has relied on study-specific platforms and tools. While some efforts have been made, these have been lacking in their integrative scope, and have suffered from support challenges (Taylor et al., 2023).

To address this infrastructural gap, we propose SHARPIE (**S**hared **H**uman-**A**I **R**einforcement-Learning **P**latform for **I**nteractive **E**xperiments). This platform provides a reusable toolset designed to standardize the design, deployment, and analysis of human-RL interactions. By moving away from rigid point-solutions and adopting a modular platform, researchers can more systematically develop more experimental setups. What is more, by first-class support for multi-agent and multi-human interaction scenarios, we hope to facilitate the study of interactions beyond the basic and unidirectional patterns we largely see today (Baraka et al., 2026) in order to study effectively address problems which humans nor agent can solve indepedently (Akata et al., 2020; Carroll et al., 2019).

The proposed platform provides versatile wrappers for popular single-agent RL and multi-agent RL environments and algorithms, supports configurable communication channels between humans and RL agents in various modalities, and offering logging services, deployment utilities, and participant recruitment platforms integration. The platform aims to empower researchers to address research questions on the interaction of RL agents and humans, focusing on interactive reward specification and learning, learning from human feedback, action delegation, preference elicitation, user-modeling, and human-AI teaming. By providing a technical platform in which these paradigms can be easily studied, compared, and combined, we hope to increase rigor and contribute to setting the standard for human-RL experiments.

Our main contributions in this work are outlined below:

- SHARPIE introduces a potential standard for RL-based human-agent interactions in a multi-human/multi-agent setting in the same way that the Gymnasium API has become this standard for fully simulated RL environments.
- SHARPIE supports experiments with diverse RL applications via minimal Gymnasium extensions and native integration of crowdworker platforms.
- four representative case studies are benchmarked to illustrate different modalities human participants can be integrated into the RL learning process.
- Several performance and scalability tests to demonstrate that our framework supports seamless distributed interaction across different numbers and geographically distant human participants, with end-to-end latency within 0.21 seconds.

The rest of the manuscript is organized as follows. Section 2 summarizes the related work for the study. Design principles of SHARPIE framework are explained in Section 3. While Section 4 provides the qualitative evaluation of SHARPIE with 4 use cases, quantitative analysis for its performance is presented in Section 5. Finally, the implications of the study along with possible future research directions are discussed in Section 6.

2 Related Work

There are several software platforms for conducting behavioral experiments (Peirce, 2007; De Leeuw, 2015). None of these offers any infrastructure for any type of artificially intelligent entities, whether learning or not. Various platforms and software packages have been developed for the deployment of RL (Gauci et al., 2019; Albers et al., 2022; Zhu et al., 2024). However, these are not focused on controlled experiments on the interaction between RL agents and humans.

Several successful platforms for the development of human-in-the-loop RL algorithms and evaluation environments have been made over the past decades, emphasizing a different number of objectives, different numbers of agents, different kinds of tasks and different programming languages. We do not intend to review all software packages in RL and here we focus only on the related work that

particularly deals with RL platforms that explicitly target the study of RL in a context that involves humans.

With recent advances in Reinforcement Learning from Human Feedback (RLHF), platforms such as (Christiano et al., 2017; Yuan et al., 2024) have been developed. However, the interfaces for human interaction provided by these platforms are only capable of handling restricted feedback, and treat humans mostly as a source of information rather than an interaction partner. Along similar lines, Qian et al. (2025) proposed UserRL, an approach for training agentic large language models based on human input with RL in various settings. This work is limited to a *one-to-one* text-based interaction with LLMs whereas we focus on the multi-human multi-agent experiments involving various modalities, including reward annotation, action guidance, as well as textual communication.

With a specific focus on user studies, Knierim et al. (2024a) presents a framework in human teaching RL agents setting with audio communication. Their codebase also facilitates a multi-participant remote Wizard-of-Oz approach (Green & Wei-Haas, 1985), where one participant can play the role of a teacher and another plays the role of a learning agent, addressing additional requirements for early iterative testing of interactive algorithmic settings, where usability and human behavior are used as a baseline, target, or data collection / calibration step. From an algorithmic perspective, although an alternative modality is proposed in this study (namely prosodic voice cues), the interactions are designed to be unidirectional, with human teachers guiding RL-agents with a restricted communication modality.

Closer to the scope of SHARPIE, the HIPPO-GYM platform (Taylor et al., 2023) provides a framework that involves human interactions in a RL setting. While similar in its main objectives as well as the needs it addresses, this framework is limited to single-agent tasks and provides only specific interactions in those tasks (i.e. humans teaching RL agents through various forms of feedback).

On the multi-agent front, previous studies in multi-agent learning such as MAgent (Zheng et al., 2018) and MOMAland (Felten et al., 2024) are capable of handling multi-agent experiments. MAgent supports a large population of agents, ranging from hundreds to millions, whereas it is possible to train the agents for multiple objectives in MOMAland. However, these packages do not yet support human interaction in any way. SHARPIE specifically targets the gap of easily setting up experiments with (multiple) human participants and RL agents by interfacing with existing environment and algorithm libraries.

Finally, the closest work ot SHARPIE that we have identified in the existing literature is Interactive Gym (McDonald, 2024) which is a framework in early stage of development that enables human-AI interaction in multi-agent RL settings. This framework currently provides two different setups: single human with multiple agents in a browser based execution and any number of human and AI agents in a client-server based execution of experiments. With its current structure, Interactive Gym seems to be specific to the Gymnasium environments where human(s) can take the control of RL agents in a game setup which can require more effort to deal with a diverse set of scenarios including various type of human-RL agent interactions. SHARPIE, on the other hand, aims to provide a more generic and extendable design that is able to efficiently facilitate such scenarios.

All in all, there are several existing approaches and subsequently experimental platforms that have looked at specific restricted human-RL scenarios, but do not focus on offering the platform as a versatile tool for research in this area. Additionally, we identified a few similar platforms that partially overlap with the aims and scope of SHARPIE, and offer a solid ground to build a unified platform. We envision SHARPIE will empower human-AI researchers to deploy interactive agents and collect data on a wide variety of human-in-the-loop RL scenarios in *rich, real-time and potentially multi-agent* settings.

3 SHARPIE Framework

SHARPIE is a Python-based web framework that aims to provide a versatile wrapper around popular Reinforcement Learning a) environments, b) algorithms, and c) methodologies. For the first

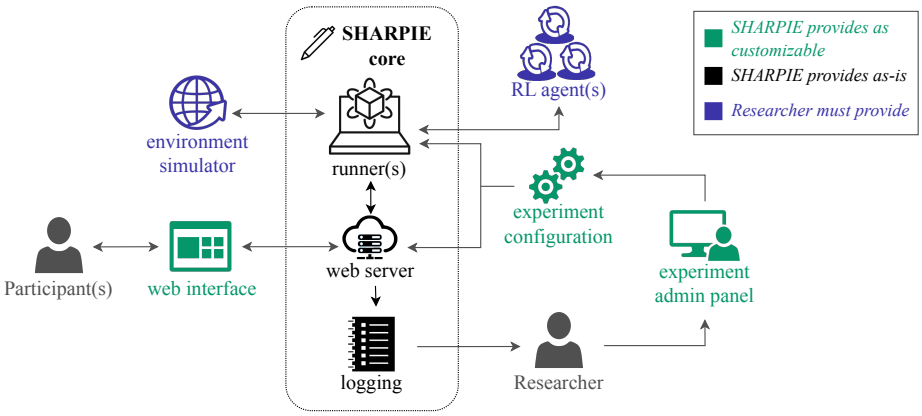

Figure 1: High-level SHARPIE architecture.

part, it can encapsulate any existing environment that follows the conventional Gymnasium API (reset, step, render, etc.) which encompasses most of the existing RL platforms (see Appendix A). Most importantly, the ambition of SHARPIE is not to be tightly integrated with any one particular environment or library, but rather to be compatible with many.

As illustrated in Figure 1, the library can be decomposed into three main families of modules. In black are the built-in components, responsible for scheduling episodes and communication between agent(s), human(s), the learning algorithm(s) and the environment. These provide the backbone of the library and are largely transparent to any but the most curious of users. In green are components that may be subjected to modification to suit a particular experiment, for instance, with arrow keys and dynamic performance feedback or text-based instructions followed by colour-coded answers. Finally, the parts of SHARPIE that the experimenters will be most manipulating are depicted in blue. These correspond mainly to the agents and their environment, although in most cases a simple wrapper around a gym- or pettingzoo-compatible environment is enough.

In terms of customizable components, the front-end User Interface (UI) is web-based and multi-modal. This allows human users to interact with the environment and other (RL or human) agents. Additionally, RL agents may be able to request action delegation, explicitly or as prompted by their learning algorithm, to further increase the range of interaction scenarios that SHARPIE can help streamline.

The front-end also provides various complementary utilities to further smoothen out the experimental processes: from (a)synchronous evaluations of a learning agent to scheduling and management on long-term data storage. In the first case, this takes many forms, such as a preference elicitation module to visualize and rank trajectories, actions, and policies. In the second case, this includes all the logging facilities required in a large-scale RL experiment, especially one involving human participants. Indeed, it is of paramount importance to be able to securely and robustly store any relevant data, as restarting such an experiment from scratch might be, at best, long and costly and, at worse, practically impossible. To this end, we devised a data model capable of handling the minutiae of multiple human studies by linking experiments, agents and time steps in a single SQL database (see Figure 3).

In addition, SHARPIE is designed to support multi-modal communication channels to allow communication between agents (again, RL or human). Depending on said agents, these channels may be used for e.g. coordination, teaching, or overriding. At its core, SHARPIE is providing json-like communication channels allowing experiments to transmit any type of information between agents, environments and the learning algorithm. Considering the most common types of data being used in RL research some types are handled natively by the library. Heavy types such as large byte ar-

Table 1: Use cases included in functional validation. The last columns indicate the numbers of lines of code (LoC) and database (DB) entries to integrate with SHARPIE.

Algorithm	Paradigm	Description	LoC	DB
SayCan (Ichter et al., 2023)	1 messages	Language instructions, learned affordances	17	5
DAgger (Ross et al., 2011)	2 actions	Queries demonstrations for visited states	65	5
SP/BC (Carroll et al., 2019)	3 environment	Plays cooperatively with a human	29	6
TAMER (Knox & Stone, 2009)	4 rewards	Learns from human evaluative feedback	31	5

rays are additionally subjected to compression to reduce the overhead brought about by network communication, in case the participant has limited bandwidth.

Finally, the library is designed with utilities to deploy to a cloud server, a private machine, or a local host. Given the widespread support for Python, this encompasses any desktop operating system (e.g. Linux, Mac, Windows) with Python 3 installed, including remote web-servers or platforms such as Amazon Web Services (AWS). This provides an abstraction between the machines that the researchers and participants are using and where the experiment is actually running. Such a feature is essential in studies involving numerous participants with varying locations and hardware. Delegation is done through the runners, allowing replicates of an experiment to be performed in parallel by instantiating multiple such runners on any machine from local servers to service providers.

4 Functional Validation via Case Studies

We validate the expressiveness and flexibility of the proposed framework by replicating multiple interactive scenarios involving RL agents and human participants. To evaluate whether the framework is sufficiently expressive to implement a wide range of such scenarios, we consider a taxonomy of interaction paradigms. Figure 2 displays the different interaction paradigms based on how humans, agents and the environment can interact with one another.

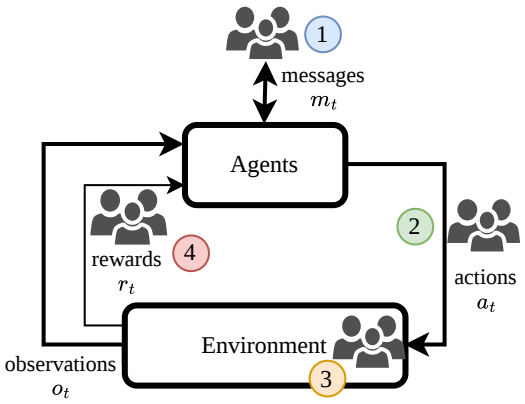

Figure 2: Taxonomy of agent-human-environment interactions.

In the first interaction paradigm, RL agents and humans can send *messages*. These messages can be used to give instructions, ask for help, explain the agent internal state, give action advice and can have various modalities including text, images, etc. In the second paradigm, humans can select and override RL agent actions. This paradigm includes shared control, learning from demonstrations etc. In the third paradigm, humans act as agents in the environment. This includes shared tasks, human-RL-teaming etc. In the fourth paradigm, the human specifies or alters an environment reward to model various evaluative feedback and reward shaping setups. As a final note on these paradigms, we want to highlight that they are not mutually exclusive and are often combined.

Table 1 lists a representative study that involves human participants for each of the above-mentioned paradigms. We evaluate SHARPIE's functional expressiveness by using it to implement a replication for each of these studies. To assess the framework's flexibility, we record the number of lines of code (LoC)[1] and the number of experiment configuration database (DB) entries. In our analysis, we focus on the overhead imposed by creating a human-participant study and therefore only include code and

[1]Excluding docstrings, comments, and empty lines

configurations to correctly route policy decisions, participant inputs, agent outputs, and environment renderings to the appropriate location. That is, we explicitly do not incorporate agent training and environment simulation logic as this is out of scope for our framework. The remainder of this section consists of detailed descriptions of these replications per use case.

SayCan by Ichter et al. (2023) implements the first paradigm, where a human participant and RL agents interact by exchanging messages containing instructions. Specifically, the human user provides high-level, unstructured natural language instructions, which the agent then grounds into executable skills using learned affordances. The primary challenge in this paradigm is message passing: standard RL environments and agent APIs typically expect fixed-dimensional numerical vectors (e.g., standard Gym spaces). Routing variable-length string payloads from a user-facing web interface to the agent's policy typically requires custom serialization and custom API wrappers. SHARPIE natively abstracts this exchange. We successfully connected a participant-facing text-input interface to a SayCan policy interacting with a table top pick-and-place environment using only 17 LoC and 5 DB entries. This demonstrates SHARPIE's capability to support experiments involving messages between humans and RL agents.

DAgger (Ross et al., 2011) represents the second interaction paradigm, where a human expert provides corrective action labels for states visited by the agent's current policy. To evaluate the framework's support for this paradigm, we implemented an early DAgger-based imitation learning experiment using the *Super Mario Bros* environment (Kauten, 2018). A substantial proportion of the 43 LoC implementation deals with migrating the environment from an old version of OpenAI-Gym library and translating participant inputs to the environment's action space. While the environment uses a bitmask-based input system to represent concurrent button presses (e.g., 'right' + 'jump'), we found that mapping these inputs to the agent's action space is straightforward using SHARPIE utilities. At the same time, SHARPIE's policy interface enables the interception of the control loop to implement DAgger's "query" phase for the expert policy. Here, our implementation routes the current observation to the participant, while the update method ensures that the resulting transition is correctly captured and buffered for the training phase. Overall, complete integration is done with 65 LoC and 5 DB entries.

Overcooked exemplifies paradigm three, where humans and RL agents act simultaneously within a shared environment. In the original study, a human participant plays cooperatively with an agent trained via either Self-Play or Behavioral Cloning (SP/BC) in the Overcooked environment (Carroll et al., 2019). The primary infrastructural challenge in this paradigm is real-time synchronization: the system must reliably stream environment renderings to the participant-facing web interface, capture participant keystrokes, query the RL policy, merge these separate actions, and transition the environment. Using SHARPIE, we replicated this functionality using 29 LoC and 6 DB entries, which highlights SHARPIE's capacity to support experiments in which users and agents operate in realtime in the same environment. Assuming access to the original agents, which currently rely on deprecated code[2], one could fully replicate the user study performed in the original paper or even scale it up through crowd-sourcing.

TAMER represents the fourth paradigm, where human participants specify an immediate reward via evaluative feedback. In the TAMER framework (Knox & Stone, 2009), the participants observe the agent act and provide an asynchronous positive or negative label to its decisions. These labels are interpreted as immediate rewards and used to train a policy during interactions with the participant and the environment. Implementing this in a standard RL codebase often requires building complex threading or networking logic to inject asynchronous UI inputs into the agent's synchronous training loop. SHARPIE's functionalities reduce this overhead significantly. We successfully captured and routed the human's real-time evaluative feedback for the MountainCar environment to the agent's reward signal using only 31 LoC and 5 DB entries. This demonstrates that SHARPIE can

[2] https://github.com/HumanCompatibleAI/overcooked_ai?tab=readme-ov-file#deprecated-behavior-cloning-and-reinforcement-learning

elegantly handle the timing and synchronization challenges inherent to interactive reward shaping at low implementation cost.

5 Scalability and Performance Profiling

We now evaluate SHARPIE's scalability and performance characteristics to ensure it meets the rigorous demands of interaction studies involving possibly multiple agents and humans. In our benchmarks we include four aspects that might affect reliability of experiment results. First, we asses the scalability wrt the number of human participants interacting with a single environment simultaneously to give an idea on the overhead imposed by the SHARPIE framework. Second, we assess the scalability with respect to the number of RL agents to assess the degree to which SHARPIE can support experiments involving multiple RL agents. Third, we assess performance in geographically distributed experimentation scenarios. Although SHARPIE's architecture supports geographically distributed experiments due to its client-server setup with the environment simulation performed at a single server, this architecture may negatively affect performance in scenarios where participants' connection is of low quality. Finally, we test how the framework is affected by environments with reasonably high computational demands, including environments producing high-resolution renderings. We detail these scenarios below for a target of 30 frames per second (FPS). All tests have been conducted using the same machine with one Intel® Core™ UItra 9 (22 cores) and 96 GB of RAM.

5.1 Human participant scalability

This experiment assesses SHARPIE's capacity to host large-scale participant studies. We simulate $n \in \{1, 10, 50, 100, 250\}$ simultaneous participants interacting simultaneously with a single real-time environment instance, in a single experiment, hosted at the same machine. To isolate SHARPIE overhead from environment-specific computation, we use a no-op environment which discards the N received actions, and returns a 64×64 RGB observation containing random noise at each time step to simulate a visual payload per timestep. Having all participants interact with the same environment instance is challenging since all inputs need to be processed simultaneously before transitioning the environment to the next state. Moreover, this single-experiment setting is the most relevant since scalability in settings with multiple environment instances can be easily achieved by hosting separate experiments on separate SHARPIE web server instances, i.e. by scaling horizontally with n.

We report system performance via several metrics in Table 2. First, we consider actual FPS as an indicator of performance for real-time interactions, averaged across participants throughout an episode. Second, we report the median and 95th percentile (P95) round-trip times (RTT) as the latency between a participant sending an action and receiving its associated observation. Third, we report throughput as a measure of capacity of the system independent of the individual participant experience. We also recorded the frequency of socket-level errors or dropped packets which was 0 in all experiments.

From Table 2 we see that the system maintains nominal performance up to $n = 100$ concurrent participants, with average FPS remaining close to the target of 30 and median RTT showing negligible variance across number of participants ($\Delta \approx 0.1$ms), although P95 latency increases substantially.At $n = 250$, however, the system has reached a performance bottleneck and average FPS degrades by 67% relative to the single-participant baseline, and Median RTT increases to 68.66 ms. This non-linear degradation indicates the saturation point of the single-instance server architecture under the tested hardware configuration. Notably, across all tested scales, the system recorded zero communication errors, indicating that while throughput limits affect fluidity and will affect experimental results, the underlying state synchronization remains intact.

5.2 Multi-agent scalability

Similarly, we evaluate SHARPIE's scalability in the number of RL agents. We simulate $n \in \{1, 10, 50, 100, 250\}$ concurrent random RL agents interacting in a single no-op environment in-

Table 2: Performance metrics by participant count, results in parenthesis are relative to $N = 1$ baseline.

n participants	FPS	median RTT (ms)	p95 RTT (ms)	Throughput (msg/s)
1	32.49 (100%)	20.12 (1.00×)	22.54 (1.00×)	50.78 (1.0×)
10	33.11 (102%)	19.82 (0.99×)	22.33 (0.99×)	480.67 (9.5×)
50	29.69 (91%)	19.90 (0.99×)	59.30 (2.63×)	2244.50 (44.2×)
100	22.98 (71%)	27.88 (1.39×)	71.25 (3.16×)	3056.07 (60.2×)
250	10.64 (33%)	68.66 (3.41×)	130.06 (5.77×)	3750.71 (73.9×)

stance. This environment discards all actions and samples a random 64×64 noise image at each timestep. The metrics reported in 3 show that the system maintains nominal performance in all cases. We hypothesize that this due to SHARPIEs architecture, in which all RL policies run in the same process as the environment simulation loop to minimize data transfer and the overhead imposed by SHARPIE consists mostly of routing and logging of messages.

Table 3: Performance metrics for a variable number of RL agents, results in parenthesis are relative to $N = 1$ baseline.

n RL agents	FPS	Throughput (msg/s)
1	32.33 (100%)	51.48 (1.00×)
10	32.40 (100.2%)	54.40 (1.05×)
50	32.39 (100.2%)	48.84 (0.94×)
100	32.43 (100.3%)	49.53 (0.96×)
250	33.08 (102.3%)	50.85 (0.98×)

5.3 Network Quality and Geographic Distribution

To assess the suitability of our server-client architecure in supporting experiments involving geographically distant participants and RL agents, we simulate the impact of network-induced latency on system performance. Specifically, we emulate various network latencies to assess FPS and RTT in the following settings: a SHARPIE webserver instance running on the participants' localhost (<0.1 ms), on the participants network (<10 ms), in the participants country (<20 ms), on the participants continent (50 ms) and world-wide (200 ms). The results reported in 4 show that, with a target of 30 FPS, SHARPIE does not suffer loss when running on the same LAN. This is perfect for hosting interactive real-time experiments. in the same university as the server running SHARPIE. For an experiment requiring participants from a cloud source platform, the system will be able to run at least 26 FPS and provide a smooth interaction. However, for regional or world-wide interactions, it is more suited for experiments with less real-time requirements.

Table 4: Performance stability analysis under variable network conditions, results in parenthesis are relative to localhost baseline.

Trial / Condition	Avg FPS (rel.)	Median RTT (rel.)	P95 RTT (rel.)	Throughput (rel.)
Localhost	32.39 (100%)	20.29 (1.00×)	21.91 (1.00×)	52.43 (1.0×)
LAN	33.08 (102%)	20.13 (0.99×)	22.00 (1.00×)	53.37 (1.0×)
National	26.77 (83%)	26.90 (1.33×)	30.34 (1.38×)	44.32 (0.85×)
Regional	14.62 (45%)	57.88 (2.85×)	60.58 (2.76×)	26.14 (0.5×)
Global	4.59 (14%)	208.72 (10.29×)	211.13 (9.64×)	8.83 (0.2×)

5.4 Environment and Rendering Complexity

Finally, we test the throughput of the framework when handling high definition rendering or highly demanding payloads. To illustrate that, we evaluate rendering different image sizes with random noise as well as testing a Starcraft II environment. The results in Table 5 show that SHARPIE can handle up to 256x256 payloads at 30FPS which will comfortably serve most use cases. Scalability beyond this size remains challenging and will require additional measures such as compression currently not implemented in SHARPIE.

Table 5: Performance metrics for a single participant under varying image size, results in parenthesis are relative to the smallest (64x64).

Image size	Avg FPS (rel.)	Median RTT (rel.)	P95 RTT (rel.)	Throughput (rel.)
64x64	33.12 (100%)	20.28 (100%)	22.22 (100%)	64.34 (100%)
128x128	33.17 (100.2%)	20.13 (99.4%)	22.67 (102.1%)	53.03 (82.5%)
256x256	30.37 (91.7%)	22.30 (110.0%)	27.89 (125.6%)	49.59 (77.1%)
512x512	12.10 (36.6%)	72.80 (359%)	77.17 (347%)	22.12 (34.4%)
1024x1024	3.43 (10.4%)	282.63 (1394%)	326.91 (1470%)	6.64 (10.3%)

6 Discussion & Future Work

We have presented and motivated the design and implementation of a framework to accelerate human–AI interaction studies, with a particular focus on studies for multi-agent tasks involving humans. As stated, the scope of this project is not to implement alternative environment and algorithms for such studies, but rather to integrate with existing solutions seamlessly and provide utilities to ease experimentation with participants. This is done by empowering researchers through a modular software interface relying, in part, on the de facto standard set by Gymnasium. With SHARPIE we aim to provide an easily integrable framework that researchers can use to painlessly set up experiments involving both human and artificial agents. Our hope is that in turn such an architecture lays the foundation for a standard for the interaction between human and artificial agents.

This modular approach also allows for numerous directions of improvement in terms of interoperability and features. In the former case, we aim to provide an increasing number of ready-made, supported plugins to handle a large part of the existing work on environments, libraries, and deployment options. In the latter case, we plan to widen the scope of possible human-agent interactions by incorporating additional modalities such as audio or video (Christofi & Baraka, 2024; Knierim et al., 2024b). The resulting real-time and multimodal communication between users and agents would allow the study of rich and fine-grained communication protocols, and support experiments involving pitch and tone, nonverbal cues, etc. Finally, we envision a hosted version of SHARPIE that can be used for outreach, education and user literacy purposes.

References

Zeynep Akata, Dan Balliet, Maarten De Rijke, Frank Dignum, Virginia Dignum, Guszti Eiben, Antske Fokkens, Davide Grossi, Koen Hindriks, Holger Hoos, et al. A research agenda for hybrid intelligence: augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. *Computer*, 53(8):18–28, 2020.

Nele Albers, Mark A Neerincx, and Willem-Paul Brinkman. Addressing people's current and future states in a reinforcement learning algorithm for persuading to quit smoking and to be physically active. *Plos one*, 17(12):e0277295, 2022.

Lucas N Alegre, Florian Felten, El-Ghazali Talbi, Grégoire Danoy, Ann Nowé, Ana LC Bazzan, and Bruno C da Silva. Mo-gym: A library of multi-objective reinforcement learning environments. In

Proceedings of the 34th Benelux Conference on Artificial Intelligence BNAIC/Benelearn, volume 2022, pp. 2, 2022.

Kim Baraka, Ifrah Idrees, Taylor Kessler Faulkner, Erdem Biyik, Serena Booth, Mohamed Chetouani, Daniel H. Grollman, Akanksha Saran, Emmanuel Senft, Silvia Tulli, Anna-Lisa Vollmer, Antonio Andriella, Helen Beierling, Tiffany Horter, Jens Kober, Isaac Sheidlower, Matthew E. Taylor, Sanne van Waveren, and Xuesu Xiao. Human-interactive robot learning: Definition, challenges, and recommendations. *J. Hum.-Robot Interact.*, 15(2), February 2026. DOI: 10.1145/3779297. URL https://doi-org.vu-nl.idm.oclc.org/10.1145/3779297.

Erdem Bıyık, Aditi Talati, and Dorsa Sadigh. Aprel: A library for active preference-based reward learning algorithms. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 613–617. IEEE, 2022.

G Brockman. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. *On the utility of learning about humans for human-AI coordination*. Curran Associates Inc., Red Hook, NY, USA, 2019.

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

Konstantinos Christofi and Kim Baraka. Uncovering patterns in humans that teach robots through demonstrations and feedback. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 332–336, 2024.

Joshua R De Leeuw. jspsych: A javascript library for creating behavioral experiments in a web browser. *Behavior research methods*, 47:1–12, 2015.

DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem, 2020. URL http://github.com/deepmind.

Florian Felten, Lucas N. Alegre, Ann Nowé, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno Castro da Silva. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.

Florian Felten, Umut Ucak, Hicham Azmani, Gao Peng, Willem Röpke, Hendrik Baier, Patrick Mannion, Diederik M Roijers, Jordan K Terry, El-Ghazali Talbi, et al. Momaland: A set of benchmarks for multi-objective multi-agent reinforcement learning. In *Multi-objective Decision Making Workshop at ECAI 2024*, 2024.

Jason Gauci, Edoardo Conti, Yitao Liang, Kittipat Virochsiri, Yuchen He, Zachary Kaden, Vivek Narayanan, Xiaohui Ye, Zhengxing Chen, and Scott Fujimoto. Horizon: Facebook's open source applied reinforcement learning platform. In *ICML 2019 Workshop on RL4RealLife*, 2019. URL https://openreview.net/forum?id=SylQKinLi4.

Kevin Godin-Dubois, Karine Miras, and Anna V Kononova. AMaze: A benchmark generator for sighted maze-navigating agents. *Journal of Open Source Software*, pp. in press, 2024.

Paul Green and Lisa Wei-Haas. The rapid development of user interfaces: Experience with the wizard of oz method. *Proceedings of the Human Factors Society Annual Meeting*, 29(5):470–474, 1985. DOI: 10.1177/154193128502900515. URL https://doi.org/10.1177/154193128502900515.

Lin Guan, Mudit Verma, Suna Sihang Guo, Ruohan Zhang, and Subbarao Kambhampati. Widening the pipeline in human-guided reinforcement learning with explanation and context-aware data augmentation. *Advances in Neural Information Processing Systems*, 34:21885–21897, 2021.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=f7OkIurx4b. Survey Certification.

Christian Kauten. Super Mario Bros for OpenAI Gym. GitHub, 2018. URL https://github.com/Kautenja/gym-super-mario-bros.

Matilda Knierim, Sahil Jain, Murat Han Aydoğan, Kenneth Mitra, Kush Desai, Akanksha Saran, and Kim Baraka. Prosody as a teaching signal for agent learning: Exploratory studies and algorithmic implications. *arXiv preprint arXiv:2410.23554*, 2024a.

Matilda Knierim, Sahil Jain, Murat Han Aydoğan, Kenneth D Mitra, Kush Desai, Akanksha Saran, and Kim Baraka. Leveraging prosody as an informative teaching signal for agent learning: Exploratory studies and algorithmic implications. In *Proceedings of the 26th International Conference on Multimodal Interaction*, pp. 95–123, 2024b.

W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pp. 9–16, 2009.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. OpenSpiel: A framework for reinforcement learning in games. *CoRR*, abs/1908.09453, 2019. URL http://arxiv.org/abs/1908.09453.

Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. RLlib: Abstractions for distributed reinforcement learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3053–3062.

PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/liang18b.html.

Chase McDonald. Interactive gym. https://github.com/chasemcd/interactive-gym, 2024.

Sriraam Natarajan, Gautam Kunapuli, Kshitij Judah, Prasad Tadepalli, Kristian Kersting, and Jude Shavlik. Multi-agent inverse reinforcement learning. In *2010 ninth international conference on machine learning and applications*, pp. 395–400. IEEE, 2010.

Jonathan W Peirce. Psychopy—psychophysics software in python. *Journal of neuroscience methods*, 162(1-2):8–13, 2007.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, et al. Userrl: Training interactive user-centric agent via reinforcement learning. *arXiv preprint arXiv:2509.19736*, 2025.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Garðar Ingvarsson, Timon Willi, Akbir Khan, Christian Schroeder de Witt, Alexandra Souly, Saptarashmi Bandyopadhyay, Mikayel Samvelyan, Minqi Jiang, Robert Lange, Shimon Whiteson, Bruno Lacerda, Nick Hawes, Tim Rocktäschel, Chris Lu, and Jakob Foerster. Jaxmarl: Multi-agent rl environments and algorithms in jax. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, pp. 2444–2446, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

William Saunders, Girish Sastry, Andreas Stuhlmüller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pp. 2067–2069, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Matthew E Taylor, Nicholas Nissen, Yuan Wang, and Neda Navidi. Improving reinforcement learning with human assistance: an argument for human subject studies with hippo gym. *Neural Computing and Applications*, 35(32):23429–23439, 2023.

Jordan Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

Marcel Torne, Max Balsells, Zihan Wang, Samedh Desai, Tao Chen, Pulkit Agrawal, and Abhishek Gupta. Breadcrumbs to the goal: goal-conditioned exploration from human-in-the-loop feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 63222–63258, 2023.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulao, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Yifu Yuan, Jianye Hao, Yi Ma, Zibin Dong, Hebin Liang, Jinyi Liu, Zhixin Feng, Kai Zhao, and Yan Zheng. Uni-RLHF: Universal platform and benchmark suite for reinforcement learning with diverse human feedback. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024. URL https://openreview.net/forum?id=WesY0H9ghM.

Yangyang Zhao, Zhenyu Wang, Changxi Zhu, and Shihan Wang. Efficient dialogue complementary policy learning via deep q-network policy and episodic memory policy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4311–4323, 2021.

Lianmin Zheng, Jiacheng Yang, Han Cai, Ming Zhou, Weinan Zhang, Jun Wang, and Yong Yu. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Zheqing Zhu, Rodrigo de Salvo Braz, Jalaj Bhandari, Daniel Jiang, Yi Wan, Yonathan Efroni, Liyuan Wang, Ruiyang Xu, Hongbo Guo, Alex Nikulkov, et al. Pearl: A production-ready reinforcement learning agent. *Journal of Machine Learning Research*, 25(273):1–30, 2024.

Supplementary Materials

The following content was not necessarily subject to peer review.

Name	Description	Language	Target	API
Amaze[1]	Maze navigation environments	Python	Research	Gymnasium+
CleanRL[2]	Algorithm library	Python/Torch	Research/Production	Gymnasium
Gymnasium[3]	De-facto standard environments	Python	Research	Gymnasium
HIPPO-Gym[4]	Human Input Parsing	Python	Research	Gymnasium+
Interactive-Gym	Interactive web-based experiment platform	Python	Research	Gymnasium+
JaxMARL[5]	Multi-agent environments	JAX	Research	Gymnasium+
Momaland[6]	Multi-agent multi-objective environments	Python	Research	Gymnasium
MO-Gymnasium[7]	Multi-objective environments	Python	Research	Gymnasium+
MORL Baselines[8]	Multi-objective algorithm library	Python	Research	Gymnasium+
OpenSpiel[9]	Game (theory) environments/algorithms	Python	Research	Gymnasium
Pearl[10]	Algorithm library	Python	Production	Gymnasium
PettingZoo[11]	Multi-agent environments	Python	Research	Gymnasium+
Ray RLlib[12]	Algorithm library	Python/Torch	Production	Gymnasium
RLax[13]	Algorithm library	JAX	Research	Gymnasium
Stable-baselines3[14]	Algorithm library	Python/Torch	Research/Production	Gymnasium
Uni-RLHF[15]	Annotation tool	Python	Annotation	Gymnasium+

Table 6: Comparison of RL Packages, APIs, and Applications. + indicates minor alterations to the Gymnasium API such as a vectorial reward for multi-objective, or vectorial action for multi-agent RL. References are: 1- Godin-Dubois et al. (2024); 2- Huang et al. (2022); 3- Brockman (2016); 4- Taylor et al. (2023); 5- Rutherford et al. (2024); 6-Felten et al. (2024); 7- Alegre et al. (2022); 8- Felten et al. (2023); 9- Lanctot et al. (2019); 10- Zhu et al. (2024); 11- Terry et al. (2021); 12- Liang et al. (2018); 13- DeepMind et al. (2020); 14- Raffin et al. (2021); 15-Yuan et al. (2024).

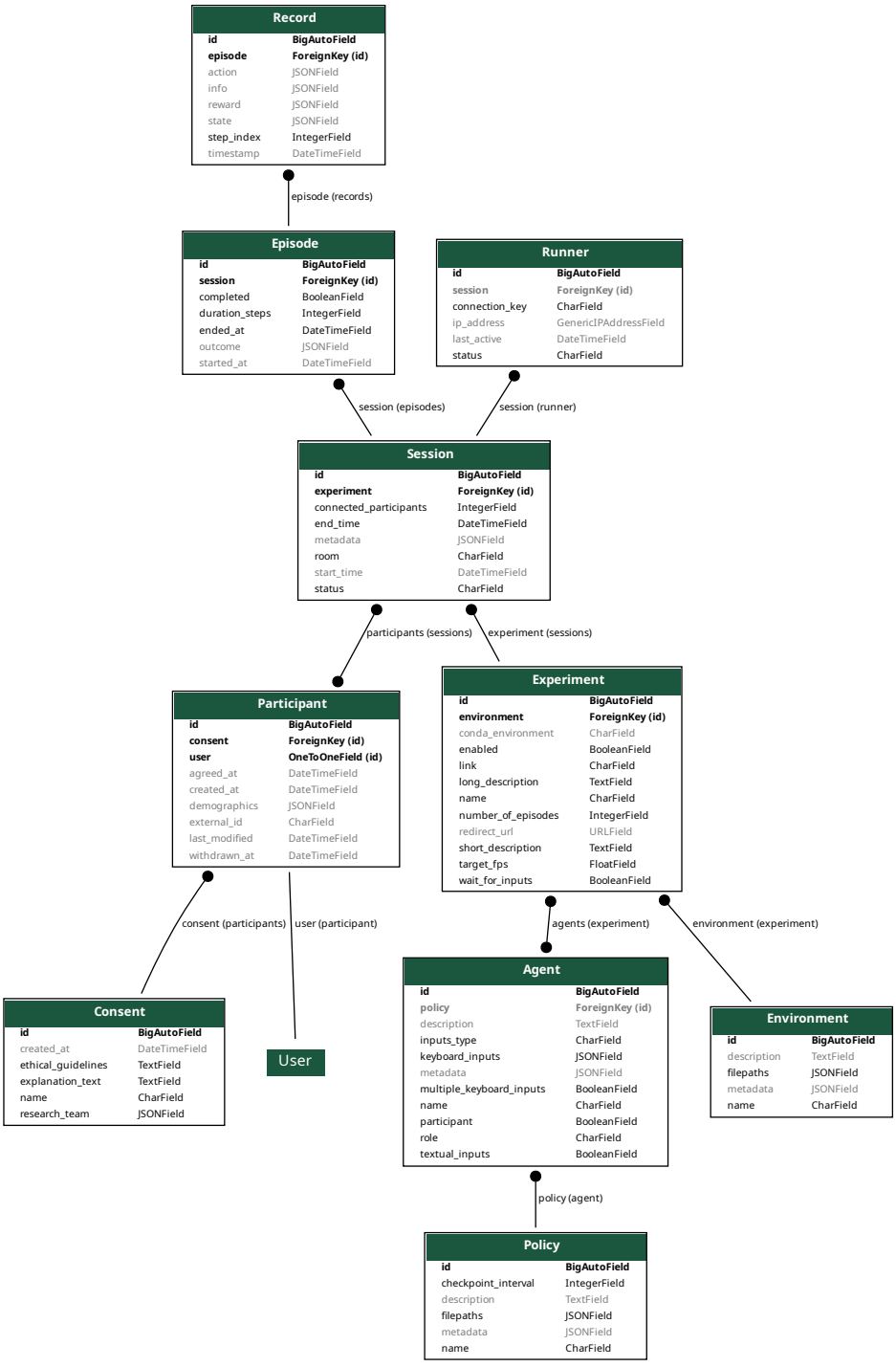

Figure 3: Overview of SHARPIE's data model