Kompatscher, Jan; Shi, Danqing; Varni, Giovanna; Weinkauf, Tino; Oulasvirta, Antti

Interactive Groupwise Comparison for Reinforcement Learning from Human Feedback

Published in:
Computer Graphics Forum

DOI:
10.1111/cgf.70290

E-pub ahead of print: 01/01/2025

Document Version
Publisher's PDF, also known as Version of record

DOI: 10.1111/cgf.70290

COMPUTER GRAPHICS *forum*
Volume 0 (2025), number 0, e70290

Interactive Groupwise Comparison for Reinforcement Learning from Human Feedback

Jan Kompatscher,[1] Danqing Shi,[1,2] Giovanna Varni,[3] Tino Weinkauf[4] and Antti Oulasvirta[1]

[1]Department of Information and Communications Engineering, Aalto University, Espoo, Finland
{jan.kompatscher, danqing.shi, antti.oulasvirta}@aalto.fi
[2]Department of Engineering, University of Cambridge, Cambridge, United Kingdom
[3]Department of Information Engineering and Computer Science, University of Trento, Trento, Italy
giovanna.varni@unitn.it
[4]Division of Computational Science and Technology, KTH Royal Institute of Technology, Stockholm, Sweden
weinkauf@kth.se

Abstract
Reinforcement learning from human feedback (RLHF) has emerged as a key enabling technology for aligning AI behaviour with human preferences. The traditional way to collect data in RLHF is via pairwise comparisons: human raters are asked to indicate which one of two samples they prefer. We present an interactive visualisation that better exploits the human visual ability to compare and explore whole groups of samples. The interface is comprised of two linked views: 1) an exploration view showing a contextual overview of all sampled behaviours organised in a hierarchical clustering structure; and 2) a comparison view displaying two selected groups of behaviours for user queries. Users can efficiently explore large sets of behaviours by iterating between these two views. Additionally, we devised an active learning approach suggesting groups for comparison. As shown by our evaluation in six simulated robotics tasks, our approach increases the final rewards by 69.34%. It leads to lower error rates and better policies. We open-source the code that can be easily integrated into the RLHF training loop, supporting research on human–AI alignment.

Keywords: interaction, human–computer interfaces,visualisation, visual analytics

CCS Concepts: • Human-centred computing → Visual analytics; Computing methodologies; Learning from critiques;

1. Introduction

Over the last decade, success with *reinforcement learning* (RL) has expanded to a wide range of difficult tasks, among them Go [SHM*16], Dota 2 [BBC*19], and Atari games [MKS*15]. The main idea in this method of experience-based training for artificial-intelligence (AI) models [SB18] is to reward preferred behaviours and punish the non-preferred ones. By optimising for reward (i.e., a numerical value from a *reward function* that judges the model's behaviour), the AI model improves its *policy*, a mapping from each state to an action. In this context, *behaviours* are segments of the agent's state–action sequences. For example, one might be part of a written response; a generated image; or a time series of angles, positions, and torques of a robot's joints. The *behaviour space* encompasses the set of all behaviours sampled as of the given time to present for human inspection.

Reward functions pose a problem, however. It has proven highly challenging to link them to human preferences by means of math ematical equations [Mah96, GCJ*24]. Therefore, research has turned to human feedback as a source of guidance for RL. One of the most popular methods of this kind is *reinforcement learning from human feedback* (RLHF) [CLB*17]. It applies the principle that, repeatedly, human evaluators shown two behaviours by an AI model (e.g., distinct images/videos) choose which of the two they prefer, thus generating preference data that function in training a *reward model*, which serves as the reward function [CLB*17, ZSW*19, OWJ*22]. This concept has informed work on such hard-to-formulate objectives as safety [DPS*23], factuality [SSC*23], and aesthetics [WDR*23]; on fine-tuning models to generate images better aligned with human preferences [BJD*23, LLR*23]; and on stronger grounding for textual explanations of images [YYZ*23].

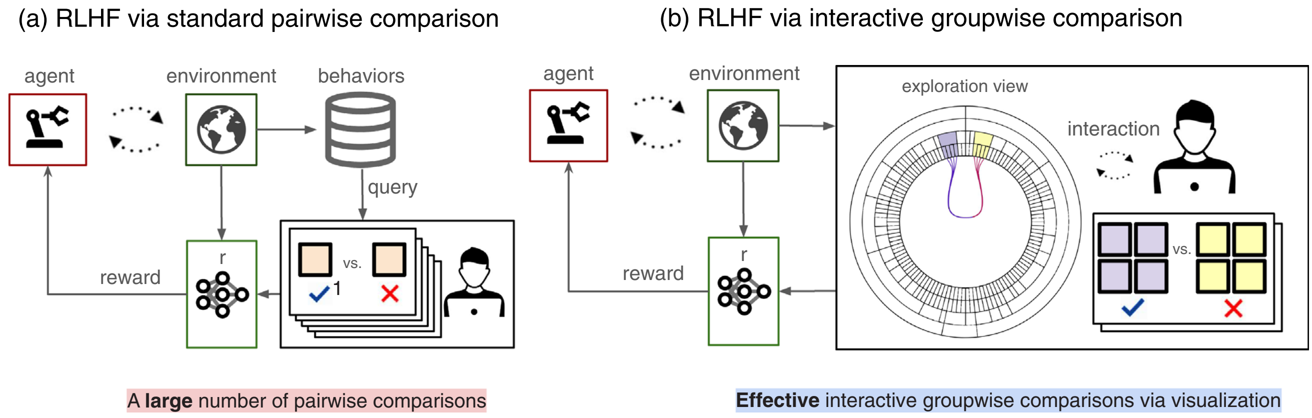


Figure 1: *The standard RLHF uses pairwise comparisons and therefore requires a large number of comparisons leading to a high workload. The comparison pairs are suggested by the system and cannot be chosen by the user. Our RLHF approach provides more agency to the user and demands less work: we leverage the user's visual abilities to effectively explore the behavior space via hierarchical grouping in the 'exploration view' and to select groups for comparison. RLHF, reinforcement learning from human feedback.*

Standard RLHF [CLB*17] relies on *pairwise comparisons*—that is, asking users to judge many pairs of behaviours in succession (see Figure 1a) until a set comparison-count threshold is reached. This is highly laborious work, though, and leaves no agency for the users, who cannot choose which behaviours to compare. It displays major limitations:

- *Time-inefficiency*: Looking at one pair of behaviours at a time is time-consuming. Gathering enough human feedback requires 700-plus comparisons for even simple behaviours such as a robot walking forward [CLB*17]. It is labour-intensive and costly.
- *Lack of user agency*: Users have a clear idea of what the desired agent behaviour should look like, yet standard RLHF does not permit them to explore and select behaviours *interactively*, to provide more effective feedback.
- *Inability to exploit context-related information*: Standard implementations neither show nor utilise valuable context information: there is no abstract overview of all behaviours, list of the comparisons already made by the user, and so forth. Users are left unable to understand the broader behaviour space.

Therefore, standard RLHF can be impractical in applications that require the expertise of a specialist (e.g., a medical doctor) and in settings wherein the aim is to teach a model for a creative purpose (such as game design). Such cases would benefit from more user agency and a lower workload than the standard RLHF approach affords [DKF22].

We set out to expand the reward-elicitation interface substantially by visualising the entire behaviour space in a hierarchical manner and letting users navigate it freely, thereby exercising greater agency. We enabled them to categorise behaviours into separate groups, then compare these groups with each other. Our tests showed that this increases efficiency; more preferences can be recorded in a given time span than with the standard approach, and the final policy returns are improved, thanks to letting the user give more informative feedback. Furthermore, we decisively augmented the behaviour space's depiction by visualising the progress thus far and suggesting sets to compare next. The user gains power to review earlier decisions and adapt the work accordingly. Figure 2 illustrates the interface created.

Prior work tackling user interfaces for RLHF addresses modular environments' role in rapidly developing such interfaces and analysing the feedback [MLB*23, YHM*24]. Groupwise comparison of behaviours has been discussed especially by Zhang et al. [ZCBD22]. Our work goes significantly further, as Section 2.2 details, by advancing how the behaviour space is visualised (via a hierarchical structure with information on the feedback history instead of a scatterplot), the complexity of the RL cases addressed (with movement patterns rather than still poses as goal behaviours), and the level of empirical evaluation (using synthetic studies in six environments, based on those environments' original reward functions, plus a non-synthetic one—as opposed to only synthetic evaluation in three environments with simplified reward functions).

This paper reports on several vital contributions:

- A novel user interface for interactive groupwise preference elicitation for RLHF based on thorough data and task abstractions (see Section 3).
- Three case studies centred on interactive groupwise comparison in training for complex novel behaviours in the robotics domain (see Section 4).
- A simulation study demonstrating that the interactive groupwise approach outperforms the standard pairwise one by 69.34% in its policy return (see Section 5).
- Evaluation with experts that identified a marked increase in efficiency: eliciting 86.7% more preferences with interactive groupwise comparison than with standard pairwise comparison (see Section 6).
- Open-source code to support further research [Kom25].

2. Background and Related Work

For a backdrop to our work, this section lays out the key concepts behind RLHF in overview, then reviews work that informed our

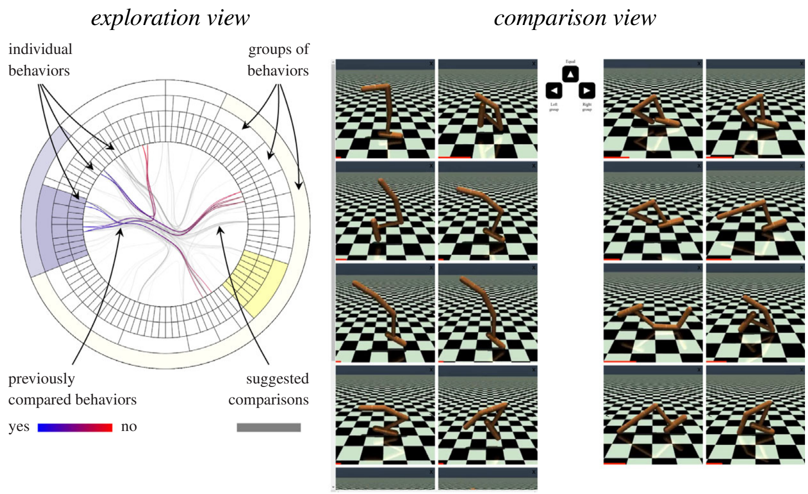


Figure 2: *Our user interface's two connected views: On the left, the* Exploration View *displays the model-supplied sampled behaviours in a hierarchically arranged radial chart. Users can select groups or individual behaviours for comparison, using a mouse. Gray lines denote machine suggestions for comparisons, while the system presents previously made comparisons in colour. This view shows two groups of videos, and the user's task is to specify a preference—that is, to state which group comes closer to the behaviour desired from the agent. Users can edit these groups at any time by adding or deleting videos, or even moving them from one group to the other.*

project: research into visualisations for RL and corresponding behaviour data.

2.1. Prerequisites for RLHF

Reinforcement learning, the branch of machine learning that teaches autonomous agents how to perform tasks by interacting within an environment [SB18], is formally defined as the sequential decision-making problem expressed by the Markov decision process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma \rangle$. At each time step t, the agent receives the state observation $s_t \in \mathcal{S}$ from the environment, where $\mathcal{S}$ is the set of possible states. The agent interacts with the environment by performing action a_t from the action space, $\mathcal{A}$. The environment enters the the next state, s_{t+1}, as defined by the state–action transition function, $\mathcal{T}$. At each step, the agent receives a scalar reward (r) from the reward function (R) that reflects the agent's performance in pursuit of the goal set. An RL agent learns to maximise the accumulated reward through a trial-and-error process by trying out actions and observing the resulting reward, which it tries to maximise. Usually, the policy training employs deep reinforcement learning [FLHI*18], which combines traditional RL algorithms with deep neural networks. This enables their use with potentially complex material, such as images and other high-dimensionality observations.

In formulating an MDP for a real-world application, designing the reward is perhaps the most decisive [SSPS21] but, at the same time, most challenging element [Mah96, GCJ*24]. Empirical evidence demonstrates the effectiveness of incorporating human preferences into RL to enhance robotics [AAC*22, HWP*24] and to tune large language models (LLMs) [ZSW*19, OWJ*22]. Learning from user priorities exhibits greater efficiency when users compare state–action trajectories by expressing their preferences [WAN*17, ASS12, FHCP12], relative to when from users demonstrate their desires [NR*00, AN04]. Research in the RL field has homed in on various methods of incorporating human feedback, with rankings, ratings scales, clustering, and so forth [ASS11, PDD*11, ASS12, DKV*15, EAPG*16, ZRL*18].

Meanwhile, other research has explored using preferences rather than absolute rewards for reinforcement learning [FHCP12, ASSS14]. Christiano and colleagues studied how to elicit human preferences from pairwise comparisons of trajectory segments [CLB*17]. This process of reward–modelling from human feedback involves learning a user's preferences from among several options by collecting feedback from that user. Users are asked to indicate their preferences via relative feedback, such as 'I prefer A over B.' Often, the preference elicitation is based on pairwise comparison, where a user query is defined as $q = \{(\tau_i, \tau_j; \mathrm{o})\}$, with $\mathrm{o} = \{\prec, \succ, \sim\}$ indicating the preference relation between the two trajectories. The preference order is commonly defined on the basis of estimated expected return $\hat{R}$ for the trajectories τ_i and τ_j. The following equation captures how the user decides on the preference order in light of the expected return implicitly given for each trajectory and the level of noise:

$$\mathrm{o}(\tau_i, \tau_j) = \begin{cases} \tau_i \succ \tau_j & \text{if } \hat{R}(\tau_i) + \epsilon_i > \hat{R}(\tau_j) + \epsilon_j \\ \tau_i \prec \tau_j & \text{if } \hat{R}(\tau_i) + \epsilon_i < \hat{R}(\tau_j) + \epsilon_j \\ \tau_i \sim \tau_j & \text{if } \hat{R}(\tau_i) + \epsilon_i = \hat{R}(\tau_j) + \epsilon_j \end{cases} \tag{1}$$

where ϵ is a term specifying the level of random noise that affects the perception of an instance. The average magnitude of ϵ is influenced by factors such as the annotator's cognitive abilities; in presentation of a behaviour for the relevant query, this noise affects how the user internally assesses the expected return for the behaviour, $\hat{R}(\cdot)$, and chooses accordingly. Through training on the order of preference from many pairs, a neural network can be trained to predict the expected return of a trajectory directly, by means of the Bradley–Terry model [BT52]. Loosely characterised, the neural network serves as the inverse of the function 1; that is, the model learns to predict $\hat{R}(\cdot)$ for a trajectory τ from a set of n queries $\mathcal{Q} = \{q_1, \ldots, q_n\}$.

2.2. Visualisations and Graphical Interfaces for Deep RL

Liu et al. demonstrated the power of visual analytics to enhance explainability and aid in implementing explainable AI [LYWY24]. Other researchers created the tool SampleViz, to help visually with analytics for RL and in debugging [LLG*24]. Scholarship is concerned also with visual analytics tools in the setting of multi-Agent RL [SZLS23, ZZL*24]. Visual analytics has shown use not only with RL but also for new machine-learning paradigms such as foundation models [YLWL24].

Along similar lines, DQNViz [WGSY18] offers a multi-level visualisation system for Deep Q-Networks, incorporating training statistics, trajectory displays, and segment-level details. This system enables users to diagnose agent behaviours and refine strategies by means of interactive visual exploration of agent experiences in Atari environments. With focus on recurrent-neural-network-based deep-RL agents, DRLViz [JVW20] and DRLIVE [WZY*21], in turn, visualise their internal memory representations. Recently, the VISITOR framework [MBJ*23] has expanded on these approaches, as a general way of exploring state sequences. Efforts with Interactive Reward Tuning [SZWO24] and RLHF-Blender [MLB*23] likewise emphasise a human-in-the-loop approach for AI alignment, letting users interactively modify reward functions or provide multiple types of feedback.

The closest relative to our technique is CLRVis, by Zhang et al. [ZCBD22], alluded to above. They too considered groupwise comparisons of behaviours. In their system, human labellers explore the behaviour space t-SNE projections. Yet t-SNE is not ideal for such a scenario, as we show in Section 3.3, since this mechanism often fails to group similar behaviours together. Furthermore, the CLRVis system creates the dataset from a comparison of two groups (of sizes m and n) by enumerating all possible pairs and hence obtaining $m \times n$ pairwise comparisons—a method not evaluated by real users. We visualise the behaviour space quite differently, create the training dataset differently, and also show comparison progress and suggestions. In further contrasts, CLRVis focuses on ranking time steps (images), whereas we enable ranking of sequences (videos). Also, we conducted empirical evaluation of our setup and can attest to its utility for more complex tasks.

2.3. Visual Interactive Labelling

Visual interactive labelling (VIL) constitutes another area of related work. Several studies by Bernard et al. [BZL*18, BZSA18, BHS*21] address this technique, in which the users find the samples in need of labels with the aid of data visualisations, whereas active learning (AL) relies on algorithms to find those samples (in both cases, the user does the labelling). From comparing the two especially [BHZ*18], their findings underline that VIL can outperform AL provided that the dimensionality-reduction technique separates the data well. Crucially, VIL can help bridge the 'cold start' problem that plagues AL. Therefore, the team provided a strong foundation on which we could build, even though they did not work on RLHF or examine VIL's utility in the context of providing human feedback for iteratively training an RL agent. In other work, Matt et al. [MSB*25] highlighted the advantage of shifting from class-to-instance assignment to instance-to-class assignment in VIL. Our work differs in focusing on preference ratings for RL rather than classification. Also, our interface-design goals were informed by Bernard et al.'s proposal of an optimal labelling strategy that consists of a 'Discovery, Consolidation, and Fine-Tuning' phase [BHL*18]. Finally, Grossmann et al. [GBSW21] showed that interface layout has little effect on accuracy estimation in VIL. Therefore, rather than seek the single most effective layout, we focus on enabling interactions that support an optimal labelling strategy.

2.4. Visualising Agent Behaviours

One can regard behaviours of RL agents as event sequences of varying length [WYYZ20]. Hierarchy-based visual representations adeptly organise and aggregate these sequences, affording the emergence of valuable insight. For instance, LifeFlow [WGGP*11] exploits a tree structure to visualise common patterns through icicle plots (which depict hierarchical data by using rectangular sectors that cascade from root to leaves [KL83]). Similarly, CoreFlow [LKD*17] illustrated branching patterns via nodes and links to highlight frequently occurring paths. GestureAnalyzer [JER14], in turn, portrayed hierarchical clustering of behaviours in a pose tree, visually representing motion trends. The follow-on system MotionFlow [JER15] emphasised transitions' depiction through flow diagrams and facilitates direct user interaction to refine a 'pose tree'. Other progress has come from ViewFusion [TTD12], which combined hierarchical structures with time-dependent activities by using treemaps, and from ActiviTree [VJC09] which offerd an interactive node–link tree layout for exploring event sequences. Although these methods capture hierarchical relationships well, displaying the data linearly can limit inter-group comparisons. We sought a design that enriches the hierarchical relations' graphical presentation, facilitates clustering, and improves comparative analysis.

3. Interactive Groupwise Comparison of Behaviours

To describe our approach, we begin by characterising the data and the task analysis undertaken, then, from the starting point thus provided, describe the approach to rendering the behaviour space visually accessible to users through data clustering and discuss three design alternatives and the visualisation interface. Integrating these methods into an interactive RLHF system culminated in the novel approach of interactive groupwise comparison for RLHF.

3.1. Characterisation of the Data

The behaviour space comprises the sequences of states and agent actions. What a state represents depends on the agent, but a state generally can be understood as a vector of numbers, with the agent's behaviour being a series of these vectors. In our robotics examples (see Section 4), the joint angles and positions of the robotic skeleton constitute the state, which could easily extend to 20 dimensions or more.

Our work focused on agents whose behaviour can be represented in the form of videos or images. Although the system was not set up to learn from the videos or images themselves, doing so would not fundamentally change the setup, as long as a distance metric between videos can serve to represent each behaviour in a lower-dimensional space. Since the state changes over time, it can be handled as n-dimensional time-series data. Behaviours are likely to differ in their length, so all analysis and visualisation algorithms applied had to be able to cope with multi-variate time-series data of varying lengths. User exploration of the behaviour data (detailed in 3.2) created a need for reducing the dimensionality of the behaviour space.

For our purposes, a sequence's frame rate is 30 images per second. Since the video clips in this setting are one second long, each behaviour is defined as a sequence of 30 frames. Other settings should be equally viable, though. Longer clips can give the model more information but grow harder for a human to judge. This tradeoff needs to be balanced.

3.2. Task Description

In a round of feedback in RLHF, users are asked to *identify desirable and undesired behaviours* when presented with behaviours of an agent (typically, 100–200 [CLB*17, GTR*22]). On this basis, the behaviours identified as better get rewarded such that the system can learn from human feedback.

The standard RLHF approach boils this overarching task down to a beguilingly simple one: *pairwise comparisons* (see Figure 1a). The user is shown two behaviours, τ_i and τ_j, in the form of an image or video. Then the user has to state a preference, for τ_i over τ_j or the other way around. It is possible also to specify no preference between τ_i and τ_j. The pairwise-comparison task is easy for humans to understand but brings with it a massive workload, since updating the underlying model requires quite a large number of such comparisons. Furthermore, without agency over which behaviours to compare, the user ends up acting as a mere tool performing a simple task many times, stoically.

We strove, then, for a system that gives the user more agency in the interaction. For this, we still addressed the overarching task defined above, but we broke it into four sub-tasks.

T.1 : *Explore behaviours from among the set of all behaviours*
T.2 : *Class behaviours into two groups (preferred vs. non-preferred)*
T.3 : *Compare between groups of behaviours*
T.4 : *Track the progress of comparison*

This approach calls for more agency on users' part, since they can see which behaviours have/have not been compared and are empowered to select the behaviours to compare next. Machine support remains available but as an offer that need not be accepted. Furthermore, we capitalise on humans' natural pattern-recognition: people can categorise behaviours and group them together [Zel13]. Lastly, groupwise comparison leads to a much lower workload [ZCBD22], thanks to working with more feedback at a time, than pairwise comparison (see Figure 1b).

3.3. Hierarchical Structuring of the Behaviour Space

Users engaging with RLHF need support for exploring given behaviours, categorising them into preference groups, comparing groups, and tracking progress. To make the behaviour space interpretable, we evaluated three methods: dimensionality reduction via principal component analysis (PCA) [MR93], t-SNE [VDMH08], and agglomerative hierarchical clustering [DE84]. All operated on pairwise distances computed with dynamic time warping [SC07].

Dimensionality-reduction methods yield a 2D embedding that can be visualised as a scatterplot. However, group-selection then requires an additional clustering step performed by the user. In contrast, hierarchical clustering directly produces clusters at multiple levels of granularity [CAKMTM17], enabling immediate selection of groups through the tree structure.

To compare methods, we measured intra-cluster variance with respect to the true reward in 10 runs for an environment for which the true reward is known [TKT*24, ETT12]. Paired t-tests showed that hierarchical clustering (hc) yields significantly lower intra-cluster variance than PCA or t-SNE (hc < PCA: $t = 6.54$, $p = 0.0001$; hc < t-SNE: $t = 4.47$, $p = 0.0016$). We concluded that hc forms more reward-consistent groups.

In summary, we found hierarchical clustering better suited to our purpose: it avoids an additional grouping step and supports multi-level exploration. The supplementary material supplies details of the data-processing pipeline and documents our analysis of the design [Kom25].

3.4. Visualisation of the Behaviour Space

For a solid design, it is crucial that the user not rely solely on trial and error when selecting behaviours. Therefore, the exploration view incorporated cues that the user can take as guidance. In addition to the hierarchical relations between behaviours, the data depiction captures adjacency relations between the leaves in the hierarchy—namely, which behaviours the user has compared with each other and which behaviours the system recommends for comparison. This directly supports handling sub-task 3.2. Following Holten's lead in detailed exploration of graphically presenting hierarchies with adjacency information [Hol06], we experimented with several options for visualising adjacency relations (the most promising are shown in the supplemental materials [Kom25]). Proceeding from these experiments alongside Holten's work and that of Schulz [Sch11], we reached the following conclusions with regard to 3.2, 3.2, and 3.2:

- The layout needs to be *space-efficient*. Treemaps and radial layouts make good use of the space available and can accommodate quite a few nodes on the screen, whereas classical node–link representations are likely to waste space from growing more in one direction than the other.
- Each node in the tree must be *selectable* by the user. Radial layouts and node–link representations simplify this while treemaps produce item overlaps that complicate it.
- The selectable nodes should remain *non-obstructed* by the adjacency information. Although treemaps' space-filling nature precludes that, a few radial layouts can meet this requirement.

Neither classical node–link nor icicle representations excel here [Hol06].

We decided on the radial layout at the left in Figure 2. In such a layout, the innermost segments represent leaf nodes and the outer layers point to parent nodes.

Within the radial chart, curved lines connecting leaf nodes express relations between behaviours. There are two types of lines: gray lines denote comparison recommendations based on the variance in the predictions, while colored ones track the user's feedback history, maintaining guidance and orientation throughout the interaction. The lines are bundled together to avoid visual clutter. The main purpose behind the gray lines is to give a window to which comparisons the reward predictors are most 'unsure' about. Although the coloured lines' core purpose is to record which comparisons have been made, so that these are not repeated, they fill a secondary function of supporting exploration by presenting the preferred versus non-preferred behaviours from each past comparison. Although endpoints in different colours may connect similar behaviours, users can easily track their past decisions, both avoiding repetition and benefiting from the colour gradient as a cue for remembering their preferences.

To prevent undue influence on the user's decision, the interface omits the absolute predictions of the reward and the mean of the predicted rewards. For RLHF, it is pivotal that users focus on their own preference without being swayed by external inputs.

3.5. Interactive Comparison of Behaviour Groups

At the heart of the user-interface design are two views: dubbed the *Exploration View* and the *Comparison View* (see Figure 2). The behaviours visualised in these views are linked. Clicking any item during exploration selects the corresponding behaviour set for comparison. Importantly, clicking a parent node brings in all behaviours falling under it, and individual behaviours can be added to or removed from a group by clicking on the relevant leaf. This combines efficiency with free selection of behaviours. From the view on the right, which displays the groups selected for comparison, the user can judge between two groups of behaviours (per 3.2), deciding which group matches the desired outcome more. Managing each group by removing outliers or transferring behaviours to the other group before the choice enhances flexibility and the preference feedback's quality both. After providing a preference decision, users are free to select the next groups via the Exploration View or request the system-recommended group-comparison query.

Label generation: Given the feedback on the two groups, of sizes m and n, the system samples $\max(m, n)$ pairs of behaviours from these groups as the preference data. Instead of using the Cartesian product of the two groups for the sampling pool (giving us $m \times n$ pairs), we sample $\max(m, n)$ pairs such that each group member is present in a pair at least once. Sampling fewer pairs per group comparison avoids overfitting to a suboptimal reward. When we sampled with the Cartesian product as proposed by Zhang et al. [ZCBD22], the resulting policies often got stuck in local optima. Hence, we turned to $\max(m, n)$ pairs of behaviours.

Active learning for suggestions: Our approach adapts techniques from active-learning heuristics in pairwise comparison [CLB*17] to groupwise comparisons by calculating a *group-variance score*. This score is higher if the reward predictors are uncertain which of the two groups is the better one (finding the correct choice requires querying a human) and is lower if they are less certain about the quality of the behaviours within the groups (i.e., when the groups are not uniform). Also, it applies a slight penalty for size differences between the groups. For details, refer to the Supporting Information [Kom25].

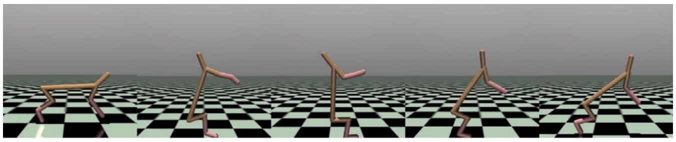
3.6. Visual Mapping of Tasks

In summary, 3.2 is mapped to a hierarchical radial chart (HRC) within the exploration view. Sub-task 3.2 maps to the comparison view. The mapping for 3.2 uses the two views in combination. Finally, 3.2 maps to employing the HRC's edge bundles as part of the exploration view and displaying the number of feedback instances completed.

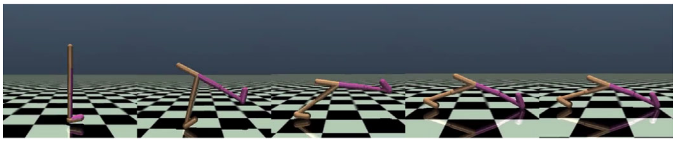
4. Example Cases

Deployments in three MuJoCo [TET12] environments commonly utilised in RL and RLHF research [CLB*17] (HalfCheetah, Walker, and Hopper) demonstrate our approach. MuJoCo is a popular free and open-source physics engine designed to facilitate research and development in robotics, biomechanics, graphics, animation, and other areas necessitating fast yet accurate simulation [TET12]. Although we follow prior work's lead in placing the robotics domain in an illustrative role, our approach holds potential for video games, image- and video-generation, and many other applications. The objective is to arrive at policies to execute behaviours for which no predefined reward functions exist; RLHF can help us train such policies. Using the interface affords purposeful provision of comparisons linked to the final intended behaviour.

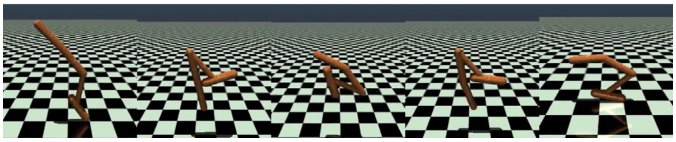
The first of the three interactive-groupwise-RLHF case studies used **HalfCheetah**. Its original objective was to apply torque to the joints for running in the forward direction. After approximately 35 min of exploration, selection of groups, and groupwise comparison via our interface, we had 601 preferences, since each comparison led to multiple preferences. HalfCheetah succeeded in learning how to stand up and sit. Second, we taught **Walker**, a robot originally designed to walk forward by means of torques to the six hinges connecting its seven body parts, to do the splits, from 616 preferences found in 40 min of exploring, selecting groups, and making groupwise comparisons. Finally, the initial goal with **Hopper** was to move the hopper forward in hops by applying torques to the three hinges connecting its four body parts. Drawing inspiration from Christiano et al. [CLB*17], who taught it to do backflips with RLHF from 900 pairwise queries (completed in less than an hour), we used the groupwise interface for exploration, groups' selection, AL, and groupwise comparison to get the robot to perform a double backflip from 1032 preferences, gathered within 34 min. Figure 3 presents sample frames capturing the case-study results.



(a) *HalfCheetah learned to stand up from 601 binary preferences.*



(b) *Walker learned to do the splits from 616 binary preferences.*



(c) *Hopper learned to double-backflip from 1,032 binary preferences.*

Figure 3: *Behaviours not coupled with ground-truth rewards can be effectively learned from interactive groupwise comparisons. For each behaviour, we present a five-frame sequence (the project page links to the video represented [Kom25]).*

5. The Simulation Study

Research evaluating human-feedback settings often utilises models of people's decision-making [BCD*21, ZCBD22, SZWO24, Bı22, AKU*18]. In aims of understanding the *general* conditions in which human experts would be able to benefit from our approach, we conducted a simulation study modelling a human DECISION-MAKER (DM) engaged in RLHF.

Running experiments in six environments with the physics engine MuJoCo, we used five runs, with different seeds, for each setting. We modeled DMs with three distinct approaches: standard *pairwise* comparison (PAIRWISE-DM), *groupwise* comparison (GROUPWISE-DM), and *interactive* groupwise comparison (INTERACTIVE-DM). **PAIRWISE-DM** takes RLHF's traditional approach, wherein one pair of behaviours gets evaluated at a time. It relies completely on the pairs presented to the DM by active learning. In **GROUPWISE-DM**, an approach permitted only through a data-pre-processing step (see Section 3.3), two groups are compared each time. This method too relies entirely on the samples chosen by AL based on the adapted heuristic with group-variance score. In **INTERACTIVE-DM**'s groupwise comparison, the comparisons are found not via active learning but interactively, through exploration of the behavior space, something that is possible only through an exploration interface. In summary, PAIRWISE-DM is the baseline approach, INTERACTIVE-DM is ours, and GROUPWISE-DM is is a no-exploration ablation of the latter.

We chose six popular robotics tasks for execution in MuJoCo: to (a) teach *Hopper* to make hops and move, (b) teach *Cheetah* to run forward, (c) teach *Walker* to run forward, (d) teach *Reacher* to touch a randomly positioned target, (e) teach the agent in *GridWorld* to move to the goal position, and (f) teach *MountainCar* to drive to the finish line. The environments' order of complexity (indicated by the length of the observation and action vectors) from simplest to most complex is MountainCar (3), GridWorld (9), Reacher (12), Hopper (14), HalfCheetah (23), Walker2d (23). The policy was learned via a reward model trained on the feedback from the DM, and the true reward function acted as the utility function to evaluate how well the trained policies reached these behaviours.

5.1. The Decision-Maker

We built DMs by modelling two key steps in PAIRWISE-DM, GROUPWISE-DM, and INTERACTIVE-DM: (1) how users select pairs/groups for comparisons and (2) how they supply their preferences. The variance score across the ensemble of reward predictors informs the active-learning suggestions.

- PAIRWISE-DM: The content with the most variance in predictions between reward predictors gets recommended for comparison.
- GROUPWISE-DM: The group-variance score (see Section 3.5) dictates the groups for comparison. The suggested groups' maximum size is set to eight behaviours to keep any group from covering too many (an excessive number could make comparisons difficult for humans in real-world settings).
- INTERACTIVE-DM: Groups to be compared are found by comparing the real average return values of the groups—something perceptible only to the human who can explore and find comparisons, not through the reward-predictor models. We employed a strategy aimed at an even spread from comparing behaviours with low rewards to comparing ones with high rewards.

We modelled the preferences of the DM based on the noisy user model defined in Equation (1) [CLB*17]. For PAIRWISE-DM, preferences were determined from the true rewards plus noise value ϵ. For GROUPWISE-DM and INTERACTIVE-DM, the group whose behaviours' rewards had a higher mean, including ϵ, was considered the preferred one. In the event of significant interference between the rewards of the two groups, comparison was skipped.

Table 1: *The average final reward and the standard deviation over five policies for each environment tested Active-learning-only groupwise comparison (*Groupwise-DM*) outperformed standard pairwise comparison (*Pairwise-DM*) in four of the six environments, and interactive groupwise comparison (*Interactive-DM*) did the same in every environment.*

Environment	Pairwise-DM		Groupwise-DM		Interactive-DM	
	M	+−	*M*	+−	*M*	+−
Hopper	1221	471	1754	851	**2053**	443
HalfCheetah	579	195	**1372**	799	1269	1583
Walker	92	155	118	268	**603**	642
Reacher (-)	−280	110	−239	78	**−202**	49
MountainCar	18059	389	**18363**	263	18259	261
GridWorld	762	85	712	23	**770**	31

Note: Interactive-DM outperformed Groupwise-DM in four of the environments. There are large differences in rewards from one environment to another, stemming from the environments' true reward functions differing greatly in their scales; however, in every environment a higher reward is better than a lower one.

5.2. The Results

The rewards achieved by the final trained policies are presented in Table 1, and Figure 4 shows the training curves. Simpler RL environments (e.g., Gridworld) let us solve properly every time with relatively simple networks (2 fully-connected layers, with 64 neurons) and fairly few feedback instances (400); harder RL problems (e.g., Walker2d) sometimes do not get solved. For this reason, the average training curve of the Walker2d environment slopes downward after 2–3 million steps and HalfCheetah shows very high variance. Generally, Interactive-DM outperformed Pairwise-DM across all environments.

Both Groupwise-DM and Interactive-DM yielded a slight decrease in errors made in binary comparisons relative to Pairwise-DM. Incorrect comparisons grow less likely when the choice is between uniform groups as opposed to single behaviours. The probability that the perceptions of the rewards are skewed in the wrong direction (such that the wrong result emerges) is lower for two groups with multiple observed rewards than it is for just two behaviours with one reward each. Our user study (reported upon in Section 6) confirmed this finding.

Because the DMs in Interactive-DM often compared more similar groups—ones unlikely to be suggested by the active-learning technique—they did more post-selection discarding of groups. Therefore, they made fewer comparisons than in Groupwise-DM with its pure active learning. Therefore, they made fewer comparisons (expressing roughly the same number of preferences as in Pairwise-DM). The remaining comparisons sufficed for creation of a more robust reward model that had better knowledge of different degrees of 'goodness.' The optimal strategy for DM agents and real-world users alike is themselves finding comparisons that cover the range from the most desired to most undesired behaviours well, to guide the agent with a

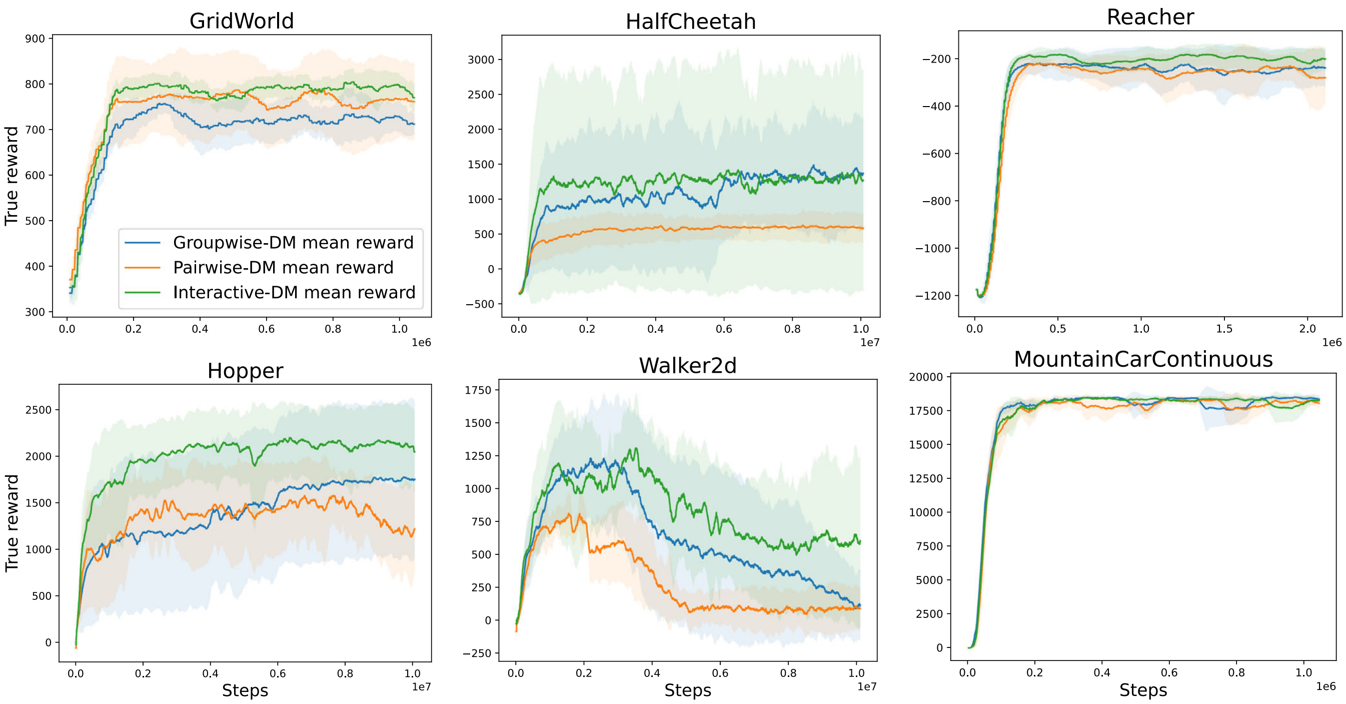


Figure 4: *Simulation-study training logs for six environments, grounded in the true rewards. Each trajectory is the average of five separate runs; the shaded region denotes the confidence interval. In most environments, groupwise active-learning-only comparisons (*Groupwise-DM*) led to higher true rewards in the final policy than did pairwise comparisons (*Pairwise-DM*), on average. Interactive groupwise comparisons (*Interactive-DM*) did the same relative to *Groupwise-DM*. With all environments, *Interactive-DM* outperformed *Pairwise-DM* in terms of the average final reward.*

denser reward signal in all the steps needed for reaching a good policy.

Delving into the results from training the agents with 400 comparisons (see Table 1) reveals that for four of the six environments INTERACTIVE-DM led to higher returns than GROUPWISE-DM on the true reward. In the other two (HalfCheetah and MountainCar-Continuous), GROUPWISE-DM performed better but not by much; furthermore, the training curve for HalfCheetah shows that the average of INTERACTIVE-DM reached the maximum reward before a million steps while it took six million steps before the average of GROUPWISE-DM overtook INTERACTIVE-DM.

In summary, our statistical comparison examining normalised final rewards (scaled such that the interquartile range lies in [0,1]) produced three main conclusions:

1) **Both GROUPWISE-DM and INTERACTIVE-DM outperform PAIRWISE-DM**. The latter reaches a 69.3% higher average reward than PAIRWISE-DM, with statistically significance ($t = 2.684$, $p = 9.456e-03$). GROUPWISE-DM's corresponding advantage over it, 41.3%, does not have statistical significance ($t = 1.636$, $p = 1.072e-01$).
2) **INTERACTIVE-DM and GROUPWISE-DM yield comparable rewards**. Although INTERACTIVE-DM yields a 28.0% higher reward, on average, than GROUPWISE-DM, this difference lacks statistical significance ($t = 1.147$, $p = 2.560e-01$).
3) **GROUPWISE-DM elicits numerous binary preferences while INTERACTIVE-DM is more efficient**. On average, PAIRWISE-DM yields 400 preferences, INTERACTIVE-DM 690, and GROUPWISE-DM 1,209. Groupwise-DM supplies significantly more preferences than PAIRWISE-DM ($t = 2.663$, $p = 1.129e-02$), while INTERACTIVE-DM exceeds the PAIRWISE-DM figure somewhat but not significantly ($t = 1.033$, $p = 3.083e-01$), and it yields significantly fewer preferences than GROUPWISE-DM ($t = -2.107$, $p = 4.175e-02$).

5.3. Implications

Both GROUPWISE-DM and INTERACTIVE-DM surpass PAIRWISE-DM in terms of reward. However, INTERACTIVE-DM does so more efficiently, from substantially fewer preferences.

One could object to this interpretation since there are two dependent variables: the number of preferences and the final reward. Therefore, we conducted additional analysis in which we limited the number of preferences that each comparison approach is allowed to return. In that condition – which is far from realistic, since comparison of groups enables supplying many more preferences—we again used five training runs per method in all six environments. In this testing, GROUPWISE-DM reaped worse normalised rewards, with a mean of -0.003 (*STD* 1.238) than PAIRWISE-DM, with a 0.582 mean (*STD* 0.780) (t = -2.154, p = 3.544e-02). INTERACTIVE-DM reached a higher mean, 0.920 (*STD* 0.650), relative to PAIRWISE-DM ($t = 1.792$, p = 7.834e-02) and to GROUPWISE-DM ($t = 3.555$, p = 7.601e-04). Figure 5 crystallises the approach-specific distributions of the final returns when the number of preferences is fixed. GROUPWISE-DM shows the lowest returns on average if we limit the number of preferences to be the same as with PAIRWISE-DM. That makes sense, in that GROUPWISE-DM's key advantage is that it affords returning more preferences. That said, even with the comparison count capped, INTERACTIVE-DM proved much better than GROUPWISE-DM and slightly (though not significantly in this experiment) better than PAIRWISE-DM.

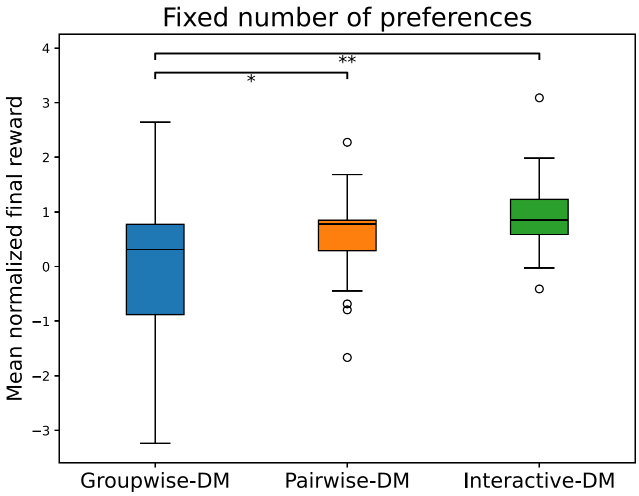


Figure 5: *An unrealistic setting, created for controlled analysis—when the number of preferences is fixed, GROUPWISE-DM does not bring higher average rewards than PAIRWISE-DM. Although groupwise comparison's main benefit is that more preferences can be elicited within the same time, INTERACTIVE-DM outperformed both other approaches even with the preference count held constant. The boxes denote the interquartile range, and the black line in each represents the median.*

In conclusion, mistakes seem less commonplace when two homogeneous groups rather than two atomic behaviours get mutually compared. The general conditions in which users can benefit from the approach arise when they (a) increase the number of pairs judged by using groupwise comparison to their advantage and/or (b) find suitable candidate comparisons by exploring the behaviour space, thereby giving more meaningful preferences to the reward model. People who explore might provide fewer comparisons in all but can assure that their feedback gives more information about the target behaviour they want to teach the RL agent.

6. The User Study

We carried out a user study aimed at evaluating the real-world *efficiency*, *usefulness*, and *ease of use* of interactive groupwise comparison (INTERACTIVE-UI) versus a baseline of standard pairwise comparison (PAIRWISE-UI). The participants were all experts applying RL in their day-to-day work.

6.1. The Study Design

Participants: We recruited 10 expert users (identified as E1–E10) with at least a year's RL experience. The users, three of whom were female, averaged 2.5 years of experience with RL (*SD* = 1.58). Eight were already familiar with RLHF, while E4 and E5 encountered it for the first time during our study.

Figure 6: *A participant performing the task from a Firefox browser on a desktop computer running Ubuntu, with a 27-inch retina display.*

Experiment procedure: We invited the experts to engage in two RLHF sessions, with different tools: one with PAIRWISE-UI and the other with INTERACTIVE-UI. We chose the hopper test environment, thanks to its easy-to-understand goal for participants. The experts were instructed that 'the centre of the robot is the joint closest to the pointy end. The first priority is for the centre of the robot to move to the right (moving to the left is worse than not moving at all). If the two robots are roughly tied by this metric, then the tiebreaker is how high the centre is.' With INTERACTIVE-UI, the participants were instructed to follow the active-learning suggestions two thirds of the time.

Experiment design: The study employed a within-subjects design in which there was one independent variable, with two levels: PAIRWISE-UI and INTERACTIVE-UI. We counterbalanced the order of the two conditions.

The experiment protocol: All sessions were held in a lab setting (shown in Figure 6), that used a Firefox browser on a Ubuntu desktop platform with a 27-inch retina display (2560 × 1440, 60 fps) and a commodity GPU (NVIDIA GeForce RTX 2080). Each expert began with 30 min of training to guarantee an opportunity for familiarity with the usage of the tools. The first 10 min consisted of a tutorial giving an introduction to the tools. Then, participants freely tested the environments and played with the tools. After the training came the two RLHF sessions, each lasting about 35 min. The time for giving feedback in RLHF was fixed, with all participants completing seven 3-min rounds of feedback (for 21 min of user work per tool). After each round, the models were retrained for about 2 min, and new videos were generated for user analysis. Each tool recorded the number of preference choices returned and logged training performance as measured by the true reward. After completing both sessions, which a researcher observed, the experts completed a questionnaire and the researcher conducted a post-experiment interview lasting about 20 min. Each study lasted roughly 2 hours. Participants received a 30-euro gift voucher as compensation for their time.

6.2. Results

Overall ratings: Per the overall ratings from the questionnaire, outlined in Figure 7, INTERACTIVE-UI comparison fared better than PAIRWISE-UI's for *efficiency* and *usefulness*. The participants rated PAIRWISE-UI higher for *ease of use*.

Efficiency and the feedback's accuracy: For each approach, we tabulated the number of preferences that the participants supplied

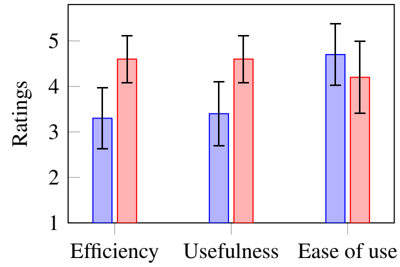


Figure 7: *Participants' ratings on a five-point Likert scale (the vertical lines show the standard deviation). The experts found INTERACTIVE-UI more efficient and useful than PAIRWISE-UI.*

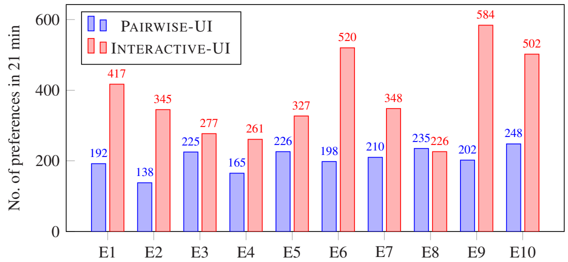


Figure 8: *The number of preferences supplied by the users, by comparison type: PAIRWISE-UI versus INTERACTIVE-UI. The vast majority of the experts (E1–E10) produced more preferences in the time given when using INTERACTIVE-UI rather than PAIRWISE-UI.*

(see Figure 8). All experts but one offered more preferences with INTERACTIVE-UI than PAIRWISE-UI. On average, users supplied 86.7% more in the same span of time when using INTERACTIVE-UI. We also factored in the number of mistakes that the experts made in their preference indications, based on the true reward known for the task. Their error rate was generally lower with INTERACTIVE-UI (10.8% of preferences) as compared to PAIRWISE-UI (12.8%). In summary, INTERACTIVE-UI yields more preferences in total, along with an error rate lower than PAIRWISE-UI's. Adhering to the initial instructions fairly well, participants obtained comparison candidates from the exploration view about a third of the time. They often succeeded in producing lower error rates, more preferences, and better policies.

Quality of the trained policies: The average reward from INTERACTIVE-UI was 1043, which is 60.9% higher overall than the reward value from PAIRWISE-UI, 648. However, the sample in the user study was quite small, not permitting claims of statistical significance. In 7 of the 10 sessions, INTERACTIVE-UI produced better policies than PAIRWISE-UI after the set feedback-provision time. From those seven sessions, the hopper learned to move forward, bringing higher scores than the PAIRWISE-UI condition. INTERACTIVE-UI yielded the four best policies, with the top one having a true reward of over 2500.

Experts' remarks: All but one of the participants found INTERACTIVE-UI efficient to work with (e.g., 'It is efficient because I have a global overview,' 'Yes, more control over similar cases,' and 'Yes, I can control what I wanna compare'). Only E1 deemed

PAIRWISE-UI efficient, thanks to its simplicity ('It's efficient because it only provides three options'), while declaring INTERACTIVE-UI inefficient since 'although I get more detailed visualisation and more options, it is not efficient [when] compared to only three options.' Seven users reacted negatively to the *information* from PAIRWISE-UI, and the others remained neutral; for example, E3 stated, 'Local, limited, does the job but minimally, so it's slightly boring and repetitive.' All of the experts gave favourable assessments of the information supplied by INTERACTIVE-UI: 'More inclusive of the bigger picture, more complete visualisation of the data coverage so far,' 'It was useful to be able to see previous preferences and where in the 'trajectory space' the clips came from,' 'The tool['s] suggestions [were] nice, to select better videos,' and so forth.

Although three experts expressed neutral opinions about PAIRWISE-UI's controllability and the other seven responded negatively ('It is simple but not helpful in exploring the behaviours,' 'It does not [let one] see a lot of behaviours,' 'The pairwise tool did not allow much exploration; I imagine that mistakes in the pairwise tool would be quite costly' and so forth.), all but one stated that they sensed more control with INTERACTIVE-UI.

That said, INTERACTIVE-UI was experienced as harder to use. For example, one participant stressed that the way of comparing clips in this tool 'needs several rounds to get used to.' Even E7, who racked up the highest scores across all experiments, said of INTERACTIVE-UI that 'I could do more comparisons at the same time and choose better videos but it was more cognitively demanding' while PAIRWISE-UI made it 'quick to select the best video and less cognitively demanding.' Indeed, his score in the pairwise experiments was the second highest.

The experts' input in summary: We conclude that INTERACTIVE-UI offers efficient, flexible functionality but requires more cognitive effort. On the other hand, PAIRWISE-UI is simpler, with limited exploration and control capabilities. These tradeoffs emphasise the importance of considering task complexity and user cognitive load in future iterations of comparison tools.

7. Discussion

Notwithstanding the clear advantages ushered in with the exploration view and tools for groupwise comparison, we recognise some limitations.

Granting users the power to provide quality feedback in larger quantities brings a danger of drawing in more meaningless feedback and added noise if they pay little attention or work in very broad strokes only. The user must strike a careful balance so as to be quick yet still provide variety-rich, accurate feedback. Our study with real human users spotlighted this factor. Although giving them a set time for both interfaces, to afford fair comparison between the two, might be a somewhat unrealistic constraint, properly judging the relative advantages of the interface demanded it. These conditions revealed that some experts could mesh INTERACTIVE-UI with their way of working and train much better policies while others gave less valuable feedback—so much so that their policies' rewards were lower than with the standard interface even though they stated more preferences in absolute terms. Skills and effort are required. Cognitive load, stress, and so forth, rear their head here. In the interviews, some users cited the mental burden in the interactive visualisation as greater than with a tool that does not allow for exploration. The questionnaire data echo this sense of higher perceived cognitive load: INTERACTIVE-UI received lower average scores for only ease of use, relative to the pairwise baseline.

The design's scalability imposes further limits. Although our system enables user agency and exploration, its radial design faces restrictions and might not scale well to behaviour spaces where two behaviours can differ on many thousands of dimensions (e.g., LLaMa 3 applies a 4096-dimension token-embedding vector for each token). Although the specific design we deployed serves smaller cases better, enabling its methods for user agency and exploration could still benefit handling other cases too.

This design is limited also in that the visualisation of the behaviour space remains quite abstract until a behaviour or group of them gets selected. Therefore, a feature pointing out representative behaviours within a group could be worthwhile: this should let users see at a glance what characterises the group before they choose it. Another idea for future work came from one of the expert users, who suggested showing a preview of the clips next to the mouse upon hovering over a behaviour in the exploration view. Although the current design does not support these features, they could be added, advancing the tool greatly in future design iterations.

In theory, comparing groups rather than individual behaviours could increase error rates, since outliers within a group may distort the labels generated. Empirically, we observed the opposite. Further research is needed to identify the conditions under which this error-reduction tendency occurs and whether it reverses in some other circumstances.

There are various avenues for developing design features that enhance the approach further. For now, the most obvious safeguards for success in applying it are to be sure the sample from the behaviour space is not overly large (sampling fewer than 200 behaviours per round is best) and work with well-trained, attentive users.

8. Conclusions

Our central contribution is a novel approach for interactive groupwise comparison of behaviours for RLHF. Evaluations of its interactive visualisation, which includes a hierarchical radial chart and edge bundling to aid in exploring and analysing behaviours for comparison, attest to the trained policy returns improving by 69.34% with respect to the baseline. Its efficiency benefits were highly evident from the evaluation study carried out with experts, in which the number of preferences elicited within a given time frame rose by 86.7%. Moreover, the interactive approach reduced error rates (incorrect preference input). Expert interviews support concluding that the new approach empowers users to control their exploration and gain comprehensive understanding of the context.

There are inherent tradeoffs to consider when reflecting on interactive groupwise comparison against the backdrop of standard pairwise comparison. Our approach offers high efficiency but may require longer training, to familiarise users with its visual-analysis features. Also, it demands more cognitive investment during use.

However, users exposed to it emphasised its efficient functioning and flexibility. Through such advances, visualisation research has much to offer processes for training AI models (RLHF and others) by designing interfaces with room for strong user agency and that help users make full use of their cognitive capabilities.

Acknowledgements

J.K., S.D., and A.O. received support from the Research Council of Finland (FCAI: 328400, 345604, 341763; Subjective Functions: 357578) and the ERC (AdG project Artificial User: 101141916). TW was supported by the Swedish e-Science Research Centre, or SeRC. AO acknowledges the research environment provided by ELLIS Institute Finland.

Open access publishing facilitated by Aalto-yliopisto, as part of the Wiley - FinELib agreement.

Conflict of Interest

The authors declare that they have no conflict of interest.

Ethics Statement

Local regulations do not require formal ethics review.

References

[AAC*22] Abramson J., Ahuja A., Carnevale F., Georgiev P., Goldin A., Hung A., Landon J., Lhotka J., Lillicrap T., Muldal A., et al.: Improving multimodal interactive agents with reinforcement learning from human feedback. *arXiv Preprint arXiv:2211.11602* (2022). URL: https://arxiv.org/abs/2211.11602.

[AKU*18] Arakawa R., Kobayashi S., Unno Y., Tsuboi Y., Maeda S.: DQN-TAMER: Human-in-the-loop reinforcement learning with intractable feedback, 2018. URL: https://arxiv.org/abs/1810.11748.

[AN04] Abbeel P., Ng A. Y.: Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning* (2004), p. 1.

[ASS11] Akrour R., Schoenauer M., Sebag M.: Preference-based policy learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011. Proceedings, Part I 11* (2011), Springer, pp. 12–27.

[ASS12] Akrour R., Schoenauer M., Sebag M.: April: Active preference learning-based reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012. Proceedings, Part II 23* (2012), Springer, pp. 116–131.

[ASSS14] Akrour R., Schoenauer M., Sebag M., Souplet J.-C.: Programming by feedback. In *International Conference on Machine Learning* (2014), vol. *32*, JMLR. org, pp. 1503–1511.

[Bı22] Bıyık E.: Learning preferences for interactive autonomy, 2022. URL: https://arxiv.org/abs/2210.10899.

[BBC*19] Berner C., Brockman G., Chan B., Cheung V., Debiak P., Dennison C., Farhi D., Fischer Q., Hashme S., Hesse C., Józefowicz R., Gray S., Olsson C., Pachocki J. W., Petrov M., de Oliveira Pinto H. P., Raiman J., Salimans T., Schlatter J., Schneider J., Sidor S., Sutskever I., Tang J., Wolski F., Zhang S.: Dota 2 with large scale deep reinforcement learning. *arXiv abs/1912.06680* (2019). URL: https://cdn.openai.com/dota-2.pdf.

[BCD*21] Bignold A., Cruz F., Dazeley R., Vamplew P., Foale C.: An evaluation methodology for interactive reinforcement learning with simulated users. *Biomimetics 6*, 1 (2021), 13. URL: https://www.mdpi.com/2313-7673/6/1/13, https://doi.org/10.3390/biomimetics6010013.

[BHL*18] Bernard J., Hutter M., Lehmann M., Müller M., Zeppelzauer M., Sedlmair M.: Learning from the best: visual analysis of a quasi-optimal data labeling strategy. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers* (Goslar, DEU, 2018), EuroVis '18, Eurographics Association, pp. 95–99.

[BHS*21] Bernard J., Hutter M., Sedlmair M., Zeppelzauer M., Munzner T.: A taxonomy of property measures to unify active learning and human-centered approaches to data labeling. *ACM Transactions on Interactive Intelligent Systems 11*, 3–4 (2021), 20. URL: https://doi.org/10.1145/3439333.

[BHZ*18] Bernard J., Hutter M., Zeppelzauer M., Fellner D., Sedlmair M.: Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 298–308. https://doi.org/10.1109/TVCG.2017.2744818.

[BJD*23] Black K., Janner M., Du Y., Kostrikov I., Levine S.: Training diffusion models with reinforcement learning. *arXiv abs/2305.13301* (2023). URL: https://arxiv.org/abs/2305.13301.

[BT52] Bradley R. A., Terry M. E.: Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika 39*, 3/4 (1952), 324–345. URL: http://www.jstor.org/stable/2334029.

[BZL*18] Bernard J., Zeppelzauer M., Lehmann M., Müller M., Sedlmair M.: Towards user-centered active learning algorithms. *Computer Graphics Forum 37*, 3 (2018), 121–132. https://doi.org/10.1111/cgf.13406.

[BZSA18] Bernard J., Zeppelzauer M., Sedlmair M., Aigner W.: Vial: a unified process for visual interactive labeling. *Visual Computer 34* (2018), 1189–1207. https://doi.org/10.1007/s00371-018-1500-3.

[CAKMTM17] Cohen-Addad V., Kanade V., Mallmann-Trenn F., Mathieu C.: Hierarchical clustering: Objective functions and algorithms, 2017. URL: https://arxiv.org/abs/1704.02147.

[CLB*17] Christiano P. F., Leike J., Brown T. B., Martic M., Legg S., Amodei D.: Deep reinforcement learning from human preferences. *arXiv abs/1706.03741* (2017). URL: https://arxiv.org/abs/1706.03741.

[DE84] Day W. H., Edelsbrunner H.: Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification 1*, 1 (1984), 7–24.

[DKF22] Daniels-Koch O., Freedman R.: The expertise problem: Learning from specialized feedback, 2022. URL: https://arxiv.org/abs/2211.06519.

[DKV*15] Daniel C., Kroemer O., Viering M., Metz J., Peters J.: Active reward learning with a novel acquisition function. *Autonomous Robots 39* (2015), 389–405.

[DPS*23] Dai J., Pan X., Sun R., Ji J., Xu X., Liu M., Wang Y., Yang Y.: Safe RLHF: Safe reinforcement learning from human feedback. *arXiv abs/2310.12773* (2023). URL: https://arxiv.org/abs/2310.12773.

[EAPG*16] El Asri L., Piot B., Geist M., Laroche R., Pietquin O.: Score-based inverse reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)* (2016).

[ETT12] Erez T., Tassa Y., Todorov E.: *Infinite-Horizon Model Predictive Control for Periodic Tasks with Contacts*. MIT Press, 2012, pp. 73–80.

[FHCP12] Fürnkranz J., Hüllermeier E., Cheng W., Park S.-H.: Preference-based reinforcement learning: A formal framework and a policy iteration algorithm. *Machine Learning 89* (2012), 123–156.

[FLHI*18] François-Lavet V., Henderson P., Islam R., Bellemare M. G., Pineau J., et al.: An introduction to deep reinforcement learning. *Foundations and Trends in Machine Learning 11*, 3-4 (2018), 219–354.

[GBSW21] Grossmann N., Bernard J., Sedlmair M., Waldner M.: Does the layout really matter? A study on visual model accuracy estimation, 2021, *IEEE Visualization Conference (VIS)*, 61–65. URL: https://arxiv.org/abs/2110.07188.

[GCJ*24] Gupta D., Chandak Y., Jordan S., Thomas P. S., C da Silva B.: Behavior alignment via reward function optimization. *Advances in Neural Information Processing Systems 36* (2024), Spotlight Poster.

[GTR*22] Gleave A., Taufeeque M., Rocamonde J., Jenner E., Wang S. H., Toyer S., Ernestus M., Belrose N., Emmons S., Russell S.: Imitation: Clean imitation learning implementations. arXiv:2211.11972v1 [cs.LG], 2022. URL: https://arxiv.org/abs/2211.11972.

[Hol06] Holten D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics 12* (2006), 741–748. URL: https://ieeexplore.ieee.org/document/4015425.

[HWP*24] Hwang M., Weihs L., Park C., Lee K., Kembhavi A., Ehsani K.: Promptable behaviors: Personalizing multi-objective rewards from human preferences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 16216–16226.

[JER14] Jang S., Elmqvist N., Ramani K.: Gestureanalyzer: Visual analytics for pattern analysis of mid-air hand gestures. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction* (2014), pp. 30–39.

[JER15] Jang S., Elmqvist N., Ramani K.: Motionflow: Visual abstraction and aggregation of sequential patterns in human motion tracking data. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 21–30.

[JVW20] Jaunet T., Vuillemot R., Wolf C.: DRLViz: Understanding decisions and memory in deep reinforcement learning. In *Computer Graphics Forum* (2020), vol. *39*, Wiley Online Library, pp. 49–61.

[KL83] Kruskal J. B., Landwehr J. M.: Icicle plots: Better displays for hierarchical clustering. *American Statistician 37*, 2 (1983), 162–168. URL: http://www.jstor.org/stable/2685881.

[Kom25] Komp J.: Interactive RLHF, 2025. Includes code, supplementary material, main paper, and video figure. URL: https://jankomp.github.io/interactive_rlhf/.

[LKD*17] Liu Z., Kerr B., Dontcheva M., Grover J., Hoffman M., Wilson A.: Coreflow: Extracting and visualizing branching patterns from event sequences. In *Computer Graphics Forum* (2017), vol. *36*, Wiley Online Library, pp. 527–538.

[LLG*24] Liang Z., Li G., Gu R., Wang Y., Shan G.: Sampleviz: Concept based sampling for policy refinement in deep reinforcement learning. In *2024 IEEE 17th Pacific Visualization Conference (PacificVis)* (2024), pp. 359–368. https://doi.org/10.1109/PacificVis60374.2024.00051.

[LLR*23] Lee K., Liu H., Ryu M., Watkins O., Du Y., Boutilier C., Abbeel P., Ghavamzadeh M., Gu S. S.: Aligning text-to-image models using human feedback. *arXiv abs/2302.12192* (2023). URL: https://arxiv.org/abs/2302.12192.

[LYWY24] Liu S., Yang W., Wang J., Yuan J.: Visualization for artificial intelligence, Springer (2025). https://doi.org/10.1007/978-3-031-75340-4.

[Mah96] Mahadevan S.: Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning 22*, 1 (1996), 159–195.

[MBJ*23] Metz Y., Bykovets E., Joos L., Keim D., El-Assady M.: Visitor: Visual interactive state sequence exploration for reinforcement learning. In *Computer Graphics Forum* (2023), vol. *42*, Wiley Online Library, pp. 397–408.

[MKS*15] Mnih V., Kavukcuoglu K., Silver D., Rusu A. A., Veness J., Bellemare M. G., Graves A., Riedmiller M. A., Fidjeland A. K., Ostrovski G., Petersen S., Beattie C., Sadik A., Antonoglou I., King H., Kumaran D., Wierstra D., Legg S., Hassabis D.: Human-level control through deep reinforcement learning. *Nature 518* (2015), 529–533. URL: https://www.nature.com/articles/nature14236.

[MLB*23] Metz Y., Lindner D., Baur R., Keim D. A., El-Assady M.: RLFH-blender: A configurable interactive interface for learning from diverse human feedback. *arXiv abs/2308.04332* (2023). URL: https://sites.google.com/view/rlhfblender.

[MR93] Maćkiewicz A., Ratajczak W.: Principal components analysis (PCA). *Computers & Geosciences 19*, 3 (1993), 303–342.

[MSB*25] Matt M., Sedlakova J., Bernard J., Zeppelzauer M., Waldner M.: Scalable class-centric visual interactive labeling. *Computers & Graphics 129* (2025), 104240. https://doi.org/10.1016/j.cag.2025.104240.

[NR*00] Ng A. Y., Russell S., et al.: Algorithms for inverse reinforcement learning. In *ICML* (2000), vol. *1*, p. 2.

[OWJ*22] Ouyang L., Wu J., Jiang X., Almeida D., Wainwright C., Mishkin P., Zhang C., Agarwal S., Slama K., Ray A., et al.: Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems 35* (2022), 27730–27744.

[PDD*11] Pilarski P. M., Dawson M. R., Degris T., Fahimi F., Carey J. P., Sutton R. S.: Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *2011 IEEE International Conference on Rehabilitation Robotics* (2011), IEEE, pp. 1–7.

[SB18] Sutton R. S., Barto A. G.: *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[SC07] Salvador S., Chan P.: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis 11*, 5 (2007), 561–580.

[Sch11] Schulz H.-J.: Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications 31*, 6 (2011), 11–15. https://doi.org/10.1109/MCG.2011.103.

[SHM*16] Silver D., Huang A., Maddison C. J., Guez A., Sifre L., van den Driessche G., Schrittwieser J., Antonoglou I., Panneershelvam V., Lanctot M., Dieleman S., Grewe D., Nham J., Kalchbrenner N., Sutskever I., Lillicrap T. P., Leach M., Kavukcuoglu K., Graepel T., Hassabis D.: Mastering the game of go with deep neural networks and tree search. *Nature 529* (2016), 484–489. URL: https://www.nature.com/articles/nature16961.

[SSC*23] Sun Z., Shen S., Cao S., Liu H., Li C., Shen Y., Gan C., Gui L., Wang Y.-X., Yang Y., Keutzer K., Darrell T.: Aligning large multimodal models with factually augmented RLHF. *arXiv abs/2309.14525* (2023). URL: https://arxiv.org/abs/2309.14525.

[SSPS21] Silver D., Singh S., Precup D., Sutton R. S.: Reward is enough. *Artificial Intelligence 299* (2021), 103535.

[SZLS23] Shi X., Zhang J., Liang Z., Seng D.: Maddpgviz: a visual analytics approach to understand multi-agent deep reinforcement learning. *Journal of Visualized 26*, 5 (2023), 1189–1205. https://doi.org/10.1007/s12650-023-00928-0.

[SZWO24] Shi D., Zhu S., Weinkauf T., Oulasvirta A.: Interactive reward tuning: Interactive visualization for preference elicitation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2024), IEEE, pp. 9254–9261.

[TET12] Todorov E., Erez T., Tassa Y.: Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), IEEE, pp. 5026–5033.

[TKT*24] Towers M., Kwiatkowski A., Terry J., Balis J. U., De Cola G., Deleu T., Goulao M., Kallinteris A., Krimmel M., KG A., et al.: Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032* (2024). https://arxiv.org/abs/2407.17032.

[TTD12] Trümper J., Telea A. C., Döllner J.: Viewfusion: Correlating structure and activity views for execution traces. In *TPCG* (2012), Citeseer, pp. 45–52.

[VDMH08] Van Der Maaten L., Hinton G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9*, 86 (2008), 2579–2605.

[VJC09] Vrotsou K., Johansson J., Cooper M.: Activitree: Interactive visual exploration of sequences in event-based data using graph similarity. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 945–952.

[WAN*17] Wirth C., Akrour R., Neumann G., Fürnkranz J., et al.: A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research 18*, 136 (2017), 1–46.

[WDR*23] Wallace B., Dang M., Rafailov R., Zhou L., Lou A., Purushwalkam S., Ermon S., Xiong C., Joty S. R., Naik N.: Diffusion model alignment using direct preference optimization. *arXiv abs/2311.12908* (2023). URL: https://arxiv.org/abs/2311.12908.

[WGGP*11] Wongsuphasawat K., Guerra Gómez J. A., Plaisant C., Wang T. D., Taieb-Maimon M., Shneiderman B.: Lifeflow: Visualizing an overview of event sequences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), pp. 1747–1756.

[WGSY18] Wang J., Gou L., Shen H.-W., Yang H.: DQNVis: A visual analytics approach to understand deep q-networks. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2018), 288–298.

[WYYZ20] Wu W., Yan J., Yang X., Zha H.: Discovering temporal patterns for event sequence clustering via policy mixture model. *IEEE Transactions on Knowledge and Data Engineering 34*, 2 (2020), 573–586.

[WZY*21] Wang J., Zhang W., Yang H., Yeh C.-C. M., Wang L.: Visual analytics for RNN-based deep reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics 28*, 12 (2021), 4141–4155. URL: https://ieeexplore.ieee.org/document/9420254.

[YHM*24] Yuan Y., Hao J., Ma Y., Dong Z., Liang H., Liu J., Feng Z., Zhao K.-W., Zheng Y.: Uni-RLHF: Universal platform and benchmark suite for reinforcement learning with diverse human feedback. *arXiv abs/2402.02423* (2024). URL: https://arxiv.org/abs/2402.02423.

[YLWL24] Yang W., Liu M., Wang Z., Liu S.: Foundation models meet visualizations: Challenges andopportunities. *Computational Visual Media 10* (2024), 399–424. https://doi.org/10.1007/s41095-023-0393-x.

[YYZ*23] Yu T., Yao Y., Zhang H., He T., Han Y., Cui G., Hu J., Liu Z., Zheng H.-T., Sun M., Chua T.-S.: RLHF-V: Towards trustworthy MLLMS via behavior alignment from fine-grained correctional human feedback. *arXiv abs/2312.00849* (2023). URL: https://arxiv.org/abs/2312.00849.

[ZCBD22] Zhang D., Carroll M., Bobu A., Dragan A.: Time-efficient reward learning via visually assisted cluster ranking. *arXiv preprint arXiv:2212.00169* (2022), Neurips Human in the Loop Learning Workshop. URL: https://arxiv.org/abs/2212.00169.

[Zel13] Zelazo P. D.: *The Oxford Handbook of Developmental Psychology, Vol. 1: Body and Mind*. Oxford University Press, 2013.

[ZRL*18] Zintgraf L. M., Roijers D. M., Linders S., Jonker C. M., Nowé A.: Ordered preference elicitation strategies for supporting multi-objective decision making. *AAMAS 17th International Conference on Autonomous Agents and MultiAgent Sustems* 2018: p. 1477–1485. URL: https://dl.acm.org/doi/10.5555/3237383.3237920.

[ZSW*19] Ziegler D. M., Stiennon N., Wu J., Brown T. B., Radford A., Amodei D., Christiano P., Irving G.: Fine-tuning language models from human preferences. *arXiv Preprint arXiv:1909.08593* (2019). URL: https://arxiv.org/abs/1909.08593.

[ZZL*24] Zhang Y., Zheng G., Liu Z., Li Q., Zeng H.: Marlens: Understanding multi-agent reinforcement learning for traffic signal control via visual analytics. *IEEE Transactions on Visualization and Computer Graphics* (2024), 1–16. https://doi.org/10.1109/TVCG.2024.3392587.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information