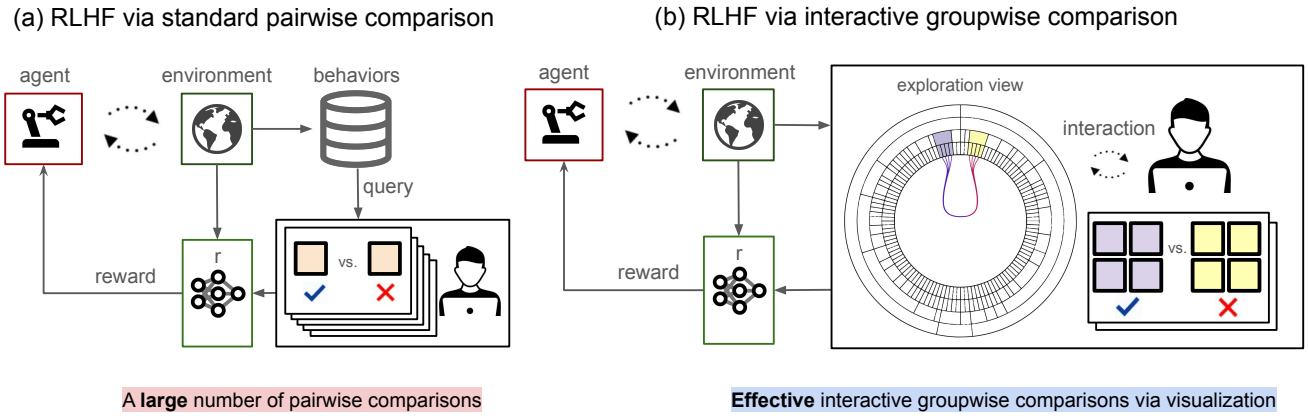
Eurographics Conference on Visualization (EuroVis) 2025
W. Aigner, N. Andrienko, and B. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 3

Interactive Groupwise Comparison for Reinforcement Learning from Human Feedback

J. Kompatscher[1], D. Shi[1], G. Varni[2], T. Weinkauf[3], and A. Oulasvirta[1]

[1]Aalto University, Finland
[2]University of Trento, Italy
[3]KTH Royal Institute of Technology, Sweden



(a) Standard RLHF requires high workload due to pairwise comparisons, and provides no ability to steer the process.

(b) Our RLHF approach requires less work due to groupwise comparisons, and the user can steer the process actively.

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.

Abstract

Reinforcement learning from human feedback (RLHF) has emerged as a key enabling technology for aligning AI behavior 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 visualization 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 behaviors organized in a hierarchical clustering structure; and 2) a comparison view displaying two selected groups of behaviors for user queries. Users can efficiently explore large sets of behaviors 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 policy returns 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.

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics;*** • ***Computing methodologies*** → *Learning from critiques;*

1. Introduction

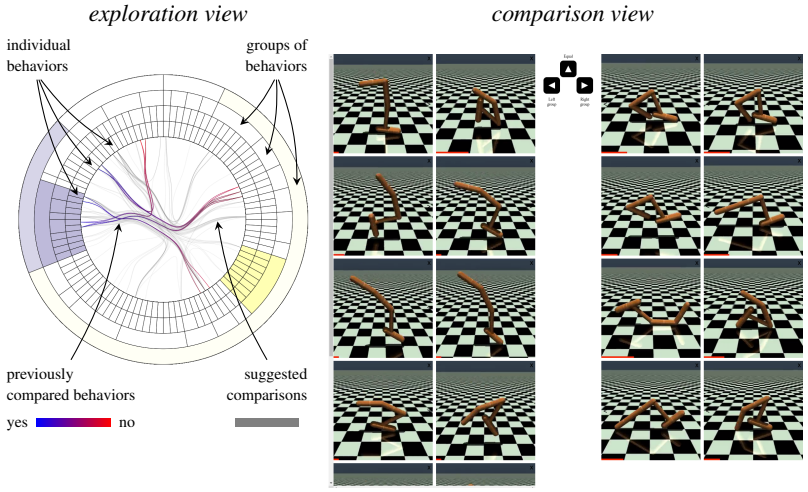


Figure 2: Our user interface consists of 2 connected views: The *Exploration View* on the left displays the sampled behaviors from the model in a hierarchical radial chart. Users can select groups or individual behaviors for comparison using a mouse. Suggestions for comparisons are shown as gray lines, while previously made comparisons are visualized in color. The *Comparison View* shows 2 groups of videos, and the user is tasked to provide the preference, i.e., to state which group comes closer to the desired behavior of 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.

Reinforcement learning (RL) is a method for training AI models from experience [SB18]. During the last decade, it has been successfully applied to a range of difficult tasks such as Go [SHM*16], Dota 2 [BBC*19], and Atari Games [MKS*15]. The main idea is to reward the preferred behaviors of the model, and punish the unpreferred ones. To achieve such a goal, a *reward function*, which judges the AI model's behavior with a numerical value that serves as a reward, is used. *Behaviors* in this context refer to segments of state-action sequences of the agent. For example, this could mean part of a written response, a generated image, or a time series of angles, positions, and torques of the joints of a robot.

However, it has turned out to be highly challenging to define reward functions that are related to human preferences in the form of mathematical equations [Mah96, GCJ*24]. Therefore, research has turned to exploit human feedback as a guidance for RL. *Reinforcement Learning from Human Feedback* (RLHF) is one of the most popular methods of this kind [CLB*17]. It works as follows: repeatedly, two different behaviors of the AI model (e.g., images or videos) are presented to human evaluators. The evaluator can then choose which of the two they prefer. This preference data is used to train a *reward model*, which serves as the reward function [CLB*17, ZSW*19, OWJ*22]. This concept has been used for hard-to-formulate objectives such as safety [DPS*23], factuality [SSC*23], or aesthetics [WDR*23], and to fine-tune models to generate images better aligned with human preferences [BJD*23, LLR*23], and to improve the grounding of textual explanations of images [YYZ*23].

Standard RLHF [CLB*17] uses *pairwise comparisons*, i.e., asking users to repeatedly compare pairs of behaviors (Fig. 1-a), until a sufficient number of comparisons is reached. This is highly laborious work that leaves no agency for the users, who cannot choose which behaviors to compare. The main limitations are:

- *Time inefficiency:* Comparing one pair of behaviors at a time is time-consuming. Gathering enough human feedback requires over 700 comparisons for a simple behavior like a robot walking forward [CLB*17], which is labor-intensive and costly.
- *Lack of user agency:* Users have their own idea of what the desired agent's behavior should look like. With the standard RLHF, users have no ability to explore and select behaviors *interactively*, to provide more effective feedback.
- *Inability to leverage contextual information:* the standard approach does not show and leverage valuable contextual information such as an overview of all behaviors, or a list of the comparisons already done by the user. This makes it impossible for users to understand the broader behavior space.

Thus, the standard RLHF can be impractical in applications that require specialized expertise (e.g., medical doctors), and in cases where users want to teach a model for a creative purpose (e.g., game design). Such cases would benefit from more agency and a lower workload than the standard RLHF approach can offer [DKF22].

Our work substantially expands the reward elicitation interface by visualizing the entire behavior space in a hierarchical manner and allowing users to freely navigate it, thereby providing more agency for the users. We enable them to categorize behaviors into different groups and compare these groups with each other. As we will show in our evaluation, this increases efficiency; i.e., more preferences can be recorded in the same time than with the standard approach, and produces higher final policy returns by letting the user provide more informative feedback. Furthermore, we augment the visualization of the behavior space indicating the achieved comparison progress and suggesting what to compare next. This allows to review previous decisions and adapt the work accordingly. See Fig. 2 for an overview of the user interface. Previous work on user interfaces for RLHF addresses modular environments for rapidly developing such interfaces and analyzing the feedback [MLB*23, YHM*24]. Groupwise comparison of behaviors has been discussed before by Zhang et al. [ZCBD22]. Our work significantly expands on this in how the behavior space is visualized, the complexity of the studied RL cases, and the level of empirical evaluation. We refer to Sec. 2.2 for a detailed discussion comparing their approach to ours.

This paper makes the following contributions:

- A novel user interface for interactive groupwise preference elicitation for RLHF based on thorough data and task abstractions (Sec. 3),
- Three case studies targeting the training of complex novel behaviors in the robotics domain using the interactive groupwise comparison. (Sec. 4),
- A simulation study showing that our interactive groupwise ap-

proach outperforms the standard pairwise approach by 69.34% in terms of policy return (Sec. 5),
- An expert evaluation showing an increase in efficiency, namely 86.7% more preferences have been elicited with interactive groupwise comparison compared to the standard pairwise comparison (Sec. 6),
- Open source code to support further research. †

2. Background

This section provides an overview of the key concepts of reinforcement learning from human feedback (RLHF). Then it reviews related work about visualizations for reinforcement learning and corresponding behavior data.

2.1. Prerequisites for RLHF. Reinforcement Learning (RL) is a branch of Machine Learning teaching autonomous agents how to perform tasks by interacting within an environment [SB18]. It is formulated as the sequential decision-making problem that is defined as 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 taking action a_t from the action space $\mathcal{A}$. The environment transits to 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 achieving the desired goal. An RL agent learns to maximize the accumulated reward through a trial-and-error process by trying out actions and observing the resulting reward which it tries to maximize. Usually, the policy is trained using deep reinforcement learning [FLHI*18], which combines traditional Reinforcement Learning algorithms with Deep Neural Networks. This enables their use with potentially complex observations, such as images and other high-dimensional observations.

When formulating an MDP for a real-world application, designing the reward is perhaps the most decisive [SSPS21] but, at the same time, most challenging part [Mah96, GCJ*24]. There is empirical evidence demonstrating the effectiveness of incorporating human preferences into RL to enhance robotics [AAC*22, HWP*24] and to fine-tune Large Language Models (LLMs) [ZSW*19, OWJ*22]. Learning from user preferences shows more efficiency in asking users to compare state-action trajectories based on their preferences [WAN*17, ASS12, FHCP12], rather than learning from user demonstrations [NR*00, AN04]. Research in the field of RL has looked at various methods of incorporating human feedback, such as scale ratings, rankings, or clusterings [ASS11, PDD*11, ASS12, DKV*15, EAPG*16, ZRL*18]. Other studies have explored using preferences rather than absolute rewards for reinforcement learning [FHCP12, ASSS14].

Christiano and colleagues studied how to elicit human preference from pairwise comparisons of trajectory segments [CLB*17]. This process of reward modeling from human feedback involves learning a user's preferences among a set of options by collecting feedback from the user. Users are asked to indicate their preferences using relative feedback, such as stating *"I prefer A over B."* This preference elicitation is often based on pairwise comparison, where a user query is defined as $q = \{(\tau_i, \tau_j; o)\}$, with $o = \{\prec, \succ, \sim\}$ indicating the preference relationship between the 2 trajectories. The preference order is commonly defined based on the estimated expected return $\hat{R}$ for the trajectories τ_i and τ_j. The following equation describes how the user will decide the preference order based on the expected return they implicitly give each trajectory and the level of noise:

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

where ϵ is a threshold, which specifies the level of random noise wich affects the perception of an instance. The average magnitude of ϵ is influenced by the cognitive abilities of the annotator, meaning that when faced with a behavior for a query, the noise affects how the user internally assesses the expected return $\hat{R}(\cdot)$ for the behavior and makes their choice accordingly. Training on the preference orders between many pairs, it is possible to train a neural network to directly predict the expected return of a trajectory, using the Bradley-Terry [BT52] model. Loosely speaking, the neural network serves as the inverse of function 1, i.e. the model learns to predict $\hat{R}(\cdot)$ for a trajectory τ from a set of n queries $\mathcal{Q} = \{q_1, ..., q_n\}$.

2.2. Visualizations and Graphical Interfaces for Deep Reinforcement Learning. Liu et al. demonstrate how visual analytics enhances the explainability and implementation of explainable AI [LYWY24]. SampleViz is a visual analytics tool to help with RL and for debugging problems [LLG*24]. Other works are occupied with visual analytics tools in the setting of Multi-Agent Reinforcement Learning [SZLS23, ZZL*24]. Not only in RL but also for new ML paradigms like foundation models, visual analytics can be useful [YLWL24].

Similarly, DQNViz [WGSY18] introduces a multi-level visualization system for Deep Q-Networks (DQNs), incorporating training statistics, trajectory displays, and segment-level details. This system enables users to diagnose agent behaviors and refine strategies through the interactive visual exploration of agent experiences in Atari environments. DRLViz [JVW20] and DRLIVE [WZY*21] focus on Recurrent Neural Network-based Deep Reinforcement Learning agents by visualizing their internal memory representations. A recent system, VISITOR [MBJ*23], has expanded these approaches by offering a general framework for exploring state sequences. Additionally, Interactive Reward Tuning [SZWO24] and RLHF-Blender [MLB*23] emphasize a human-in-the-loop approach for AI alignment, allowing users to interactively modify reward functions or provide different types of feedback.

CLRVis by Zhang et al. [ZCBD22] is the most related method to ours, since they also consider groupwise comparisons of behaviors. Human labelers explore the behavior space in a scatter plot created with t-SNE. As we will show in Sec. 3.3, t-SNE is not ideal for this scenario, since similar behaviors are often not grouped together by this method. Furthermore, CLRVis focusses on ranking time steps (images), whereas we enable to rank sequences (videos). They create their dataset from a comparison of 2 groups (size m and n) by enumerating all possible pairs and therefore getting $(m \times n)$ pairwise comparisons. However, that method has not been evaluated by real users. In contrast to their work, we visualize the behavior

† https://jankomp.github.io/interactive_rlhf

space quite differently, show comparison progress and suggestions, create the training dataset differently, provide empiric evaluation of our setup, and showcase its utility on more complex tasks.

Several works by Bernard et al. [BZL*18, BZSA18, BHS*21] handle Visual Interactive Labeling (VIL) and also compare it with Active Learning (AL) in particular [BHZ*18]. In simple terms, VIL is when the users find the samples that need to labeled with the help of data visualizations while AL means that algorithms are used to find those samples (in both cases, the labeling is done by the user). Their findings underline that VIL can outperform AL given that the dimensional reduction technique separates the data well. Crucially, VIL can help bridge the "cold start" problem that AL suffers from. Thus, they provide a strong foundation on which our paper can build. They did not work on RLHF or examin how VIL can be used in the context of providing human feedback for iteratively training a reinforcement learning agent.

2.3. Visualizing Agent Behaviors. Reinforcement Learning (RL) agent behaviors can be viewed as event sequences of varying length [WYYZ20]. Hierarchy-based visual representations adeptly organize and aggregate these sequences to uncover valuable insights. For instance, LifeFlow [WGGP*11] uses a tree structure to visualize common patterns through icicle plots (visualizations of hierarchical data using rectangular sectors that cascade from root to leaves [KL83]). Similarly, CoreFlow [LKD*17] illustrates branching patterns with nodes and links to highlight frequently occurring paths. GestureAnalyzer [JER14] offers a hierarchical clustering of behaviors into a pose tree, visually representing motion trends. The subsequent work, MotionFlow [JER15], emphasizes visualizing transitions through flow diagrams and facilitates direct user interaction for refining pose clustering. ViewFusion [TTD12] combines hierarchical structures with time-dependent activities using treemaps, and ActiviTree [VJC09] offers an interactive node-link tree layout for exploring event sequences. Although these methods effectively capture hierarchical relationships, displaying the data linearly can limit cross-group comparisons. Our design aims to enhance the visualization of hierarchical relationships, facilitate clustering, and improve comparative analysis.

3. Interactive Groupwise Comparison

In this section, we describe our approach. First, we provide a description of the data (Sec. 3.1) and a task analysis (Sec. 3.2). Based on these two preliminaries, we describe our visualization approach to making the behavior space accessible to the users through data clustering, discuss three design alternatives, (Sec. 3.3) and visualization (Sec. 3.4). We integrate these methods into an interactive RLHF system (Sec. 3.5) leading to our novel approach of interactive groupwise comparison for RLHF.

3.1. Data Description. The behavior space consists of the sequences of states and actions taken by the agent. What a state represents depends on the agent, but generally speaking, a state can be understood as a vector of numbers, and an agent's behavior is a series of these vectors. For the robotics examples from Sec. 4, a state describes the joint angles and positions of the robotic skeleton, which can easily be 20 and more dimensions.

Our work focuses on agents whose behavior can be represented in the form of videos or images . Our system is not set up to learn from the videos or images themselves, albeit this would not fundamentally change our setup as long as a distance metric between videos can be used to represent each behavior in a lower-dimensional space.

Since the state is changing over time, we can treat it as n-dimensional time series data. Different behaviors are likely to have different lengths. Hence, all our analysis and visualization algorithms will need to deal with multi-variate time series data of varying lengths. Given that the users shall explore the behavior data (see next section for details), there is a *need for reducing the dimensionality* of the behavior space.

3.2. Task Description. In a round of feedback in RLHF, users are asked to *identify desirable and undesirable behaviors* when presented with the behaviors of an agent (the typical number is between 100 and 200 [CLB*17, GTR*22]). The identified behaviors will be rewarded accordingly such that the system can learn from human feedback.

The standard RLHF approach [CLB*17] boils this overarching task down to an intriguingly simple task: *pairwise comparisons* (Fig. 1-a). Two behaviors τ_i and τ_j are shown to the user in the form of an image or video. Then the user has to decide whether they prefer τ_i over τ_j, or the other way around. It is also possible to not state a preference for τ_i and τ_j. The pairwise comparison task is easy to understand for users, yet it leads to a massive workload, since an update of the underlying model requires a fairly high number of such comparisons. Furthermore, the user has no agency over which behaviors to compare, and therefore ends up being a mere tool performing a simple task many times, stoically.

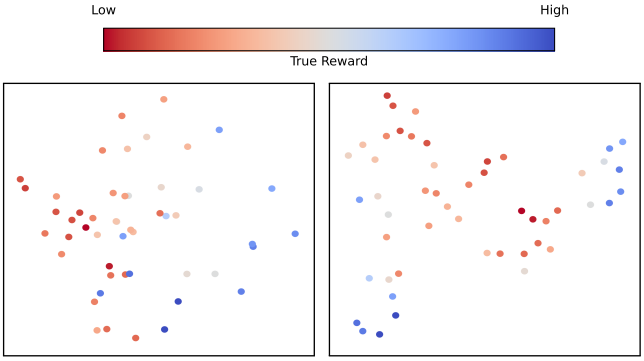
Our goal is to give the user more agency in the interaction with the system. We still address the overarching task from above, but we break it down into four sub-tasks:

T.1 *explore behaviors among the set of all behaviors*,
T.2 *categorize behaviors into 2 groups (preferred vs. unpreferred)*,
T.3 *compare groups of behaviors*, and
T.4 *track the comparison progress*.

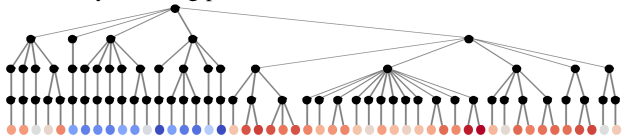
This way, more agency for the users is expected, since they can see which behaviors have and have not been compared, and are empowered to select which behaviors to compare next. We still provide machine support in this regard, but it is an offer that can be taken or not. Furthermore, we capitalize on the natural pattern recognition ability of humans to categorize behaviors and group them together [Zel13]. Lastly, a groupwise comparison leads to a much lower workload [ZCBD22], since it generates more feedback at one time than pairwise comparisons (See Fig. 1-b).

3.3. Hierarchical Clustering of the Behavior Space. To support users in Tasks T.1 and T.2, we represent the high-dimensional behavior space in a human-understandable manner. To achieve such a a goal, tried different methods for pre-processing the data, including PCA [MR93], t-SNE [VDMH08] used by CLRVis [ZCBD22], and agglomerative hierarchical clustering [DE84]. In all cases we used dynamic time-warping (DTW) [SC07] to address the temporal alignment of the behaviors.

Hierarchical clustering effectively supports group selections. With hierarchical clustering, or any other method that yields a hierarchical structure, we obtain ready-made clusters. Other clustering methods also yield ready-made clusters, however, hierarchical clustering has the advantage of finding clusters at different levels of granularity [CAKMTM17]. On the other hand, dimen-



(a) In PCA (left) and t-SNE (right), groups of similar behaviors (shown by color) are grouped reasonable well but still need to be selected by lasso-ing points with the mouse.

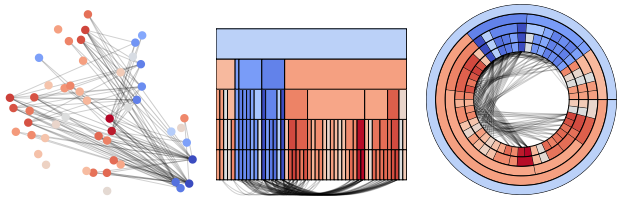


(b) Hierarchical clustering groups similar behaviors (shown by color) well. Groups at different levels of granularity are found automatically. Groups of behaviors are easy to select by choosing parent nodes in the hierarchical structure. Attention: The horizontal ordering of the tree nodes is not arbitrary but arises from the agglomerative clustering process to make the structure.

Figure 3: Comparison of different techniques for reducing the dimensionality of the behavior space. On the top, Panel (a) shows two dimensional reduction techniques that make it possible to display the behaviors in a 2d-plane e.g., by scatterplot. On the bottom, Panel (b) We show an n-ary tree structure found by directly clustering the high dimensional space. Note: the colors showing the true reward of the behaviors are used for illustration purposes only (there are no such colors in the real setting because the true reward is unknown in RLHF).

sional reduction techniques (like PCA and t-SNE) extract useful information from the high-dimensional data in order to create a low-dimensional representation that a human can perceive visually. However, when the user needs to select groups in the low-dimensional visualization, there is an intermediate step of finding clusters on their own. This intermediate step can cost time and introduce mistakes. If clustering does not perform worse in grouping together behaviors that should be similarly rewarded than dimensional reduction techniques, it will be better suited for our purposes, since it removes the intermediate step of grouping for the user.

Hierarchical clustering produces better clusters in our task domain. We compare the three design choices for our purposes. Fig. 3 compares the outcomes of the 3 methods. PCA and t-SNE project the high-dimensional space to 2D, where each point represents a behavior. The data points in the Figure are colored according to an optimal reward function for an RL environment where the true reward is known [TKT*24, ETT12] . These colors are in the Figures for illustration purposes, but would not be visible to a user working with the exploration view (since the true reward is unknown in RLHF). Similar colors should be grouped, which seems to be more the case in t-SNE than in PCA (Fig. 3a). But also in the hierarchical clustering of the behaviors, similarly colored behaviors are close to each other. The horizontal order of the tree nodes in the tree (Fig. 3b) are not due to chance but arise through the agglomerative clustering process used to find the tree structure. Hence, hierarchical clustering does not work worse for placing similar behaviors nearby than t-SNE, while providing ready-made clusters at different levels of granularity. Of course, Fig. 3 shows only one example case. We ran a comprehensive analysis over ten cases by computing the intra-cluster variance using the true reward function. If we compare using a clustering method on the scatter plots to simply selecting clusters from the hierarchical chart, paired t-tests reveal that hierarchical clustering (hc) outperforms both PCA and t-SNE with statistical significance in terms of showing lower variance within a cluster (hc < PCA: t-statistic = 6.5426, p-value = 0.0001; hc < t-SNE: t-statistic = 4.4657, p-value = 0.0016) – which is precisely required in our setup to make it easy for users to select groups. In summary, hierarchical clustering produces better clusters than first dimensionally reducing the dataset to find clusters in the reduced dataset. See the supplemental material for more details.



(a) t-SNE **(b)** Icicle plot **(c)** Radial layout

Figure 4: The user's progress in comparing behaviors shall be shown using lines. This works best in the radial layout, since there is a dedicated space for those lines, and it does not suffer from the collinearity problem as the regular icicle plot (cf. [Hol06]).

Hence, we deem hierarchical clustering best to support Tasks T.1 and T.2. Notably, it also allows for a structured overview of the behavior space on varying levels of granularity: users can understand the relationship among groups and explore different levels of the hierarchy [JER14, JER15].

3.4. Visualization of the Behavior Space. In addition to the hierarchical relationships between behaviors, our data contains adjacency information between the leaves of the hierarchy: namely, which behaviors have been compared with each other by the user, and which behaviors our system suggests for comparison. This is in direct support of Task T.4. The visualization of hierarchies with adjacency information has been explored in detail by Holten [Hol06]. We experimented with several different options for visualizing the adjacency information, some of which are shown in Fig. 4. A scatterplot with adjacency connections between the dots (4a) is a way of drawing adjacency edges in visualizations of dimensionally reduced data. Curved edges under an icicle plot (4b) and edges in the middle of a radial icicle plot (4c) are ways for drawing the edges in the hierarchical representations proposed by Holen et al. Based on these experiments, and the works of Schulz [Sch11] and Holten [Hol06], we can conclude the following for the purpose of the tasks T.1, T.2, T.4:

- The layout needs to be *space-efficient*. Treemaps and radial layouts make good use of the space available to them and can fit a

fairly large amount of tree nodes on the screen, whereas classic node-link representations are prone to wasting space, since they grow more in one direction than the other.
- Each node of the tree needs to be *selectable* by the user. Radial layouts and node-link representations afford that, yet treemaps render these items on top of each other.
- The adjacency information shall *not obstruct* the selectable tree nodes. Certain radial layouts can afford that, whereas treemaps cannot due to their space-filling nature, and classic node-link or icicle representations do not excel in this regard [Hol06].

Thus, we decided on the radial layout shown in Fig. 2. In such a layout, the inner segments represent leaf nodes and the outer layers correspond to parent nodes.

Within the radial chart, curved lines connect leaf nodes to show relationships between behaviors. There are 2 types of lines: gray lines indicate comparison suggestions based on the variance in the predictions (more technical details are provided in the next section, 3.5), while colored lines visualize the user's feedback history, which provides guidance and orientation throughout the interaction. The lines are bundled together to avoid visual clutter. The main purpose of the gray lines is to give an insight over which comparisons the reward predictors are most "unsure" about. The colored lines' main purpose is to show which comparisons have already been made so they are not repeated. Their secondary purpose is to show the preferred and unpreferred behavior of each past comparison to support exploration. Although similar behaviors can be connected by differently colored endpoints, users are able to easily track their past decisions to avoid repeating them, and they can use the color gradient as a cue for remembering their own preferences.

The visualization does not show the absolute predictions of the reward or the mean of the predicted rewards in order to not influence the users decision. It is key for RLHF that users focus on their own preference and are not influenced by outside inputs.

3.5. Interactive Comparison of Behavior Groups. Our user interface consists of 2 main views: (a) the *Exploration View*, and (b) the *Comparison View*, see Fig. 2. The behaviors visualized in these 2 views are linked. Clicking any item in the Exploration View means to select the corresponding behaviors for comparison. Notably, a group of behaviors can be selected by clicking a parent node, and individual behaviors can be added or removed from a group by clicking on a leaf. This combines efficiency with freedom for the selection of behaviors. The selected groups will be displayed in the Comparison View, where users can compare 2 groups of behaviors (Task T.3). The users are tasked to decide which group meets the desired behavior more. Users can manage each group by removing outliers or transferring behaviors to the other group. This increases flexibility and ensures the quality of preference feedback. After providing the feedback, the users are free to select the next groups via the Exploration View or ask for an automatically recommended group comparison query.

Label generation: Given the feedback about two groups of size m and n, our system samples $\max(m,n)$ pairs of behaviors from the 2 groups as the preference data. Instead of sampling the cartesian product of the two groups (giving $m \times n$ pairs), we only sample $\max(m,n)$ pairs so that each group member is present in a pair at least once. Sampling less pairs per group comparison avoids overfitting to a suboptimal reward. We tried sampling the cartesian product, as proposed by Zhang et al. [ZCBD22], but resulting policies were often stuck in local optima. Hence our suggestion to only sample $\max(m,n)$ pairs of behaviors.

Active learning: Our approach enables users to make more varied comparisons based on their feedback. As users have more agency through the Exploration View, they will not be able to explore all the possible comparisons (for n behaviors, there are $n(n-1)/2$). The more varied the comparisons they perform, the better the training data for the reward model. To explain this more concretely, imagine we want to teach a robot so walk forward. At the beginning, we will mostly get suboptimal behavior fragments where the robot just falls down since they are sampled from a randomly initialized policy. Now, if we only provide the most obvious comparisons, i.e. that standing is better than lying on the ground, the robot will not learn to walk forward; it will learn to stand still in order to not fall over. Instead, along with this obvious comparison, we should also provide less obvious ones (e.g., crawling forward is better than lying still). That way, the reward model will more accurately rate the RL policy during training and better policies can ultimately be achieved.

To support users at including more varied comparisons, we suggest places where the model is uncertain using Active Learning [CAL94]. In this approach, we give comparisons as a suggestion that show a large disagreement between members of an ensemble of reward predictors. The disagreement will not be indicative of the actual value of the comparison. That is something that only human can decide since their preference is unknown during training. However, the disagreement can be a useful proxy for the uncertainty of the model being trained [CLB*17].

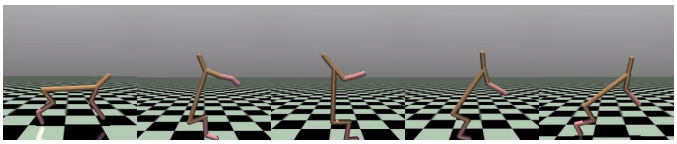
The active learning approach works through an ensemble of differently initialized preference models (in our implementation the ensemble size was 3 networks with a parameter size of 2 layers with 64 neurons each) to recommend the next pair of behaviors for comparison based on the variance between their predicted rewards. In general, the more preference models there are in our ensemble, the better the variance between them reflects uncertain comparisons. However, with every additional model the computation cost increases. The number of models used as well as the number of nodes and the number of layers in each network are hyperparameters that can be tuned to find a well performing RLHF architecture.

Our goal is that groups with lower variance within each group, higher variance between the groups, and a small difference in size are more likely to be suggested for comparison. We adapted the heuristic of Active Learning in pairwise comparison [CLB*17] to work for groupwise comparisons by calculating a group variance score s_g between 2 groups g_1 and g_2 using the following formula:

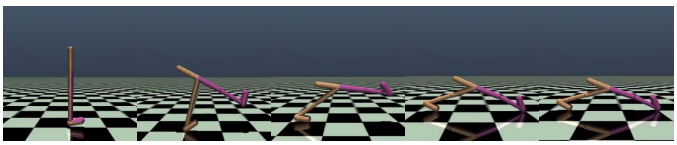
$$s_g(g_1,g_2) = \frac{v_{inter}}{r\,v_{intra} + \epsilon} \tag{2}$$

where v_{inter} represents the average variance between g_1 and g_2, v_{intra} represents the average variance within each group, ϵ is an error term (e.g. 1e-8) added to the denominator to avoid division by 0 (in case that the variance between the reward predictors within both of the groups is 0 which in practice never happens) and r is the the ratio between the two groups' sizes

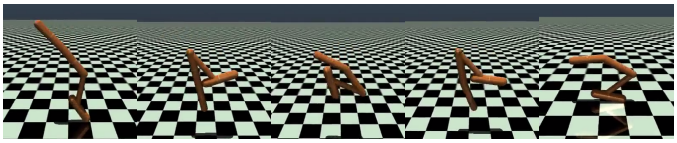
$$r = \frac{max(|g_1|,|g_2|)}{min(|g_1|,|g_2|)} \tag{3}$$



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



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



(c) Hopper learned to double backflip from 1032 preferences.

Figure 5: Behaviors that do not have ground-truth rewards can be effectively learned from interactive groupwise comparisons. We show 5 sequential frames for each behavior, and the video can be found linked in our project page.

where $|g|$ denotes the set cardinality of group g.

Inter-group variance is calculated averaging the variance of the cartesian product. The intra-group variance of a group is the average variance over all the unordered pairs within the set. The top pair of groups with the highest s_g is then recommended for comparison. This implies that groups with lower variance within each group, higher variance between the groups, and a small difference in size are more likely to be suggested for comparison.

4. Example Cases

In this section, we demonstrate our approach in 3 MuJoCo environments [TET12] (HalfCheetah, Walker, and Hopper) that are commonly utilized in research on RL and RLHF [CLB*17]. Mujoco is a popular free and open-source physics engine that aims to facilitate research and development in Robotics, Biomechanics, Graphics and Animation, and other areas where fast and accurate simulation is needed [TET12]. Our objective is to train policies to execute behaviors for which there are no predefined reward functions - therefore, RLHF can help us to train such policies. Using our interface, we could deliberately provide comparisons that are related to the final intended behavior.

The results of these 3 interactive-groupwise-RLHF case studies are shown in the accompanying *video* linked on the project page. The original objective of the **HalfCheetah** (Fig. 5a) is to apply torque to its joints to run forward. After approximately 35 minutes of exploration, selection of groups, and groupwise comparison in our interface, we provided 601 preferences, as each comparison done generated multiple preferences. The HalfCheetah successfully learned how to stand up and sit. The **Walker** (Fig. 5b) is originally designed to walk in the forward direction by applying torques on the 6 hinges connecting the 7 body parts, we instead taught Walker to do the splits based on 616 preferences found by exploration, selection of groups, and groupwise comparison in our interface in 40 minutes. The **Hopper**'s (Fig. 5c) original goal is to make hops that move it forward by applying torques on the 3 hinges connecting the 4 body parts. We drew inspiration from Christiano et al., where they taught the hopper to perform backflips with RLHF using 900 pairwise queries gathered in less than an hour [CLB*17]. Using our groupwise interface for exploration, selection of groups, active learning, and groupwise comparison, we got the hopper to perform a double backflip from 1032 preferences within 34 minutes.

5. Simulation Study

Models of of human decision-making are commonly used when evaluating visualization design, but more common in studies of decision policies in human-robot interaction [FPS*23, HA13, HAMA19]. We present the results of a simulation study aimed at understanding the *general* conditions in which human experts would be able to benefit from our approach. To this end, we simulate a DECISION MAKER (DM) doing RLHF. We run the experiments on 6 environments in MuJoCo, a physics simulator [TET12] with 5 runs of different seeds for each setting.

We model DMs with three different approaches: standard *Pairwise* comparison (PAIRWISE-DM), *Groupwise* comparison (GROUPWISE-DM), and *Interactive*-groupwise comparison (INTERACTIVE-DM). **PAIRWISE-DM** is the traditional approach of RLHF, where a pair of two behaviors is evaluated at a time. It relies fully on the pairs presented to the DM by active learning. **GROUPWISE-DM** is the approach only possible through our data pre-processing step (see section 3.3), where two groups are compared at a time. Also this approach relies fully on the samples chosen by active learning based on our proposed calculation 2. **INTERACTIVE-DM** is a groupwise comparison where the comparisons are not found through active learning but interactively, through exploration of the behavior space, something that is only possible through an *Exploration View*. In summary, PAIRWISE-DM is the baseline approach, INTERACTIVE-DM is our approach, and GROUPWISE-DM is an ablation of our approach (no exploration).

We chose six popular tasks in Robotics, executed in Mujoco: 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 placed target; e) Teach the agent in *GridWorld* to move to the goal position; f) Teach *MountainCar* to drive to the finish line. The order of complexity of the environments (based on the size of the observation and action vectors) from simplest to most complex environment is: MountainCar (3), Gridworld (9), Reacher (12), Hopper (14), HalfCheetah (23), Walker2d (23). A reward model was trained using the feedback of the DM, and the policy is trained based on that model. The true reward function was used as the utility function to evaluate how well the trained policies achieved these behaviors.

5.1. Decision Maker. We built DMs by modeling two key steps in PAIRWISE-DM, GROUPWISE-DM and INTERACTIVE-DM: 1) how users select pairs or groups for comparisons and 2) how they give their preferences. The variance score across the ensemble of reward predictors is used to determine the Active Learning suggestions.

- PAIRWISE-DM: The greatest variance in predictions between reward predictors is recommended for comparison.
- GROUPWISE-DM: The group variance score (calculation 2) is used to choose the groups to compare. The maximum size of

suggested groups is set to 8 behaviors to prevent an excessive number of behaviors in one group, which could make comparisons difficult for humans in a real setting.

- INTERACTIVE-DM: The comparisons between groups are found by comparing the real average return values of the groups – something only perceptible to the human who can explore and find comparisons, not through the reward predictor models. The employed strategy was to give an even spread of comparisons from behaviors with low rewards to behaviors with high rewards.

We modeled the preferences of the DM based on the user noisy model defined in Eq. 1 [CLB*17]. In PAIRWISE-DM, preferences were determined based on the true rewards plus noise value ϵ. For GROUPWISE-DM and INTERACTIVE-DM , the group with a higher mean of its behaviors' rewards, including noise value ϵ, was considered the preferred group. If there was a significant overlap between the rewards of the 2 groups, the comparison was skipped.

5.2. Results. The rewards achieved by the final trained policies are presented in Table 1, and the training curves can be seen in Figure 6. Easier RL environments (e.g. Gridworld) are solved properly every time by a relatively simple networks (64, 64) and limited number of feedback (400), harder RL problems (e.g. Walker2d) sometimes do not get solved. For this reason, the average training curve of the Walker2d environment goes down after 2-3 million steps and HalfCheetah shows really high variance. Generally, INTERACTIVE-DM outperformed PAIRWISE-DM across all environments.

There was a slight decrease in errors made in group comparisons (GROUPWISE-DM and INTERACTIVE-DM) in respect to PAIRWISE-DM. The probability of making incorrect comparisons decreases when choosing between 2 uniform groups over when just choosing between 2 single behaviors. The probability that the perceptions of the rewards are skewed into the wrong directions (so that the wrong result comes out) is lower for two groups of multiple observed rewards than it is for just two behaviors of one reward each. This finding was confirmed in our user study (Section 6).

Because in INTERACTIVE-DM the DM often compared more similar groups - ones that would unlikely be suggested by the Active Learning - they discarded groups more often after selection. Therefore, they made less comparisons as in GROUPWISE-DM with purely Active Learning. The DM gave similar number of preferences in PAIRWISE-DM and INTERACTIVE-DM. However, the remaining comparisons were enough to create a more robust reward model that had better knowledge about different degrees of "goodness". The optimal strategy for real users and DM alike is to find comparisons themselves that cover the range from the most desired to most undesired behaviors well to guide the agent with a more dense reward signal in all the necessary steps needed to reach a good policy.

When we look at the detailed results of training the agents with 400 comparisons in Table 1, we see for four out of six environments, INTERACTIVE-DM leads to higher returns on the true reward faster than GROUPWISE-DM. In the environment HalfCheetah as well as MountainCarContinuous, GROUPWISE-DM performs better, but not by a lot. If we look at the training curve of the HalfCheetah environment (Fig. 6), we can see that INTERACTIVE-DM reached the maximum reward before 1 million steps, while it took the reward model trained with GROUPWISE-DM 6 million steps until it overtook INTERACTIVE-DM.

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

Table 1: The average final reward and the standard deviation over 5 policies each for every tested environment. Active-learning-only groupwise comparison (GROUPWISE-DM) outperformed the standard pairwise comparison (PAIRWISE-DM) in four out of six environments. Interactive groupwise comparison (INTERACTIVE-DM) outperformed PAIRWISE-DM in every environment. INTERACTIVE-DM outperformed GROUPWISE-DM in four out of six environments. The rewards are very different from one environment to another because the true reward functions of the environments have very different scales. In every environment, a higher reward is better than a lower one.

In terms of normalized final rewards (normalized, so the inter quartile range is between 0 and 1) across all the runs and environments, GROUPWISE-DM achieves a 41.3% higher average PAIRWISE-DM but not with statistical significance (t = 1.636, p = 1.072e-01). INTERACTIVE-DM achieves 69.34% higher average than PAIRWISE-DM with statistical significance (t = 2.684, p = 9.456e-03). INTERACTIVE-DM achieves a sligthly higher average (28.04%) than GROUPWISE-DM, although the difference is not statistically significant (t = 1.147, p = 2.560e-01). If we count the number of compared pairs (preferences) that are provided with each approach, we see that on average PAIRWISE-DM returns 400, INTERACTIVE-DM 690, and GROUPWISE-DM 1209 preferences. With the t-test, we calculate that GROUPWISE-DM returns a lot more preferences than PAIRWISE-DM with statistical significance (t = 2.663, p = 1.129e-02). INTERACTIVE-DM returns more preferences than PAIRWISE-DM, but not with statistical significance (t = 1.033, p = 3.083e-01). In addition, INTERACTIVE-DM does return much less preferences than GROUPWISE-DM, with statistical significance (t = -2.107, p = 4.175e-02). In the pareto visualization (figure 7), we can see how the 3 approaches differ in final returns and in number of preferences returned. GROUPWISE-DM and INTERACTIVE-DM both outperform PAIRWISE-DM. But INTERACTIVE-DM does so with fewer preferences than GROUPWISE-DM. One might object to this analysis, since there are two dependent variables: the number of preferences and the final reward. Therefore, we conducted an additional analysis where we limited the number of preferences that each comparison approach is allowed to return. The setting is not realistic, since it is possible to provide many more preferences through the comparison of groups. We again train 5 runs for each method in all 6 environments. The results are that GROUPWISE-DM gets worse normalized rewards with a mean of -0.003 (*std* 1.238) than PAIRWISE-DM with a mean 0.582 (*std* 0.780) (t = -2.154, p = 3.544e-02). INTERACTIVE-DM achieves higher mean of 0.920 (*std* 0.650) over PAIRWISE-DM (t = 1.792, p = 7.834e-02) and over GROUPWISE-

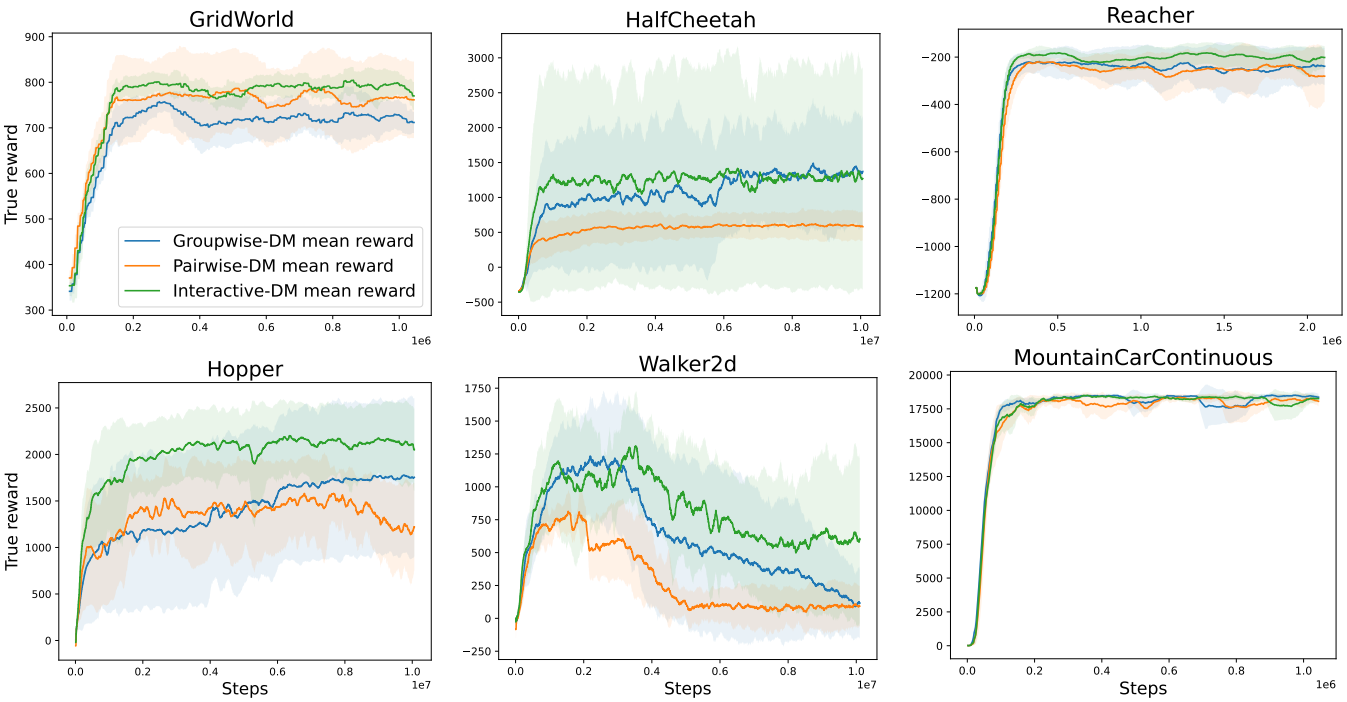


Figure 6: Simulation training logs on 6 environments measured on the true rewards. Each trajectory is the average of 5 individual runs; the shaded region shows the confidence interval. In most environments, groupwise, active-learning-only comparisons (GROUPWISE-DM) on average lead to higher true rewards in the final policy than pairwise comparisons PAIRWISE-DM. Also, in most environments, interactive groupwise comparisons (INTERACTIVE-DM) on average lead to higher true rewards in the final policy than GROUPWISE-DM. In all environments, INTERACTIVE-DM outperforms PAIRWISE-DM in terms of average final reward.

DM (t = 3.555, p = 7.601e-04). In Figure 8 we can see the distributions of the final returns achieved with each approach when the number of preferences is fixed. GROUPWISE-DM achieves the lowest returns on average if we limit the number of preferences to be the same as with PAIRWISE-DM. That makes sense, since the advantage of GROUPWISE-DM is that more preferences can be returned through it. However, INTERACTIVE-DM is a lot better than GROUPWISE-DM and slightly better than PAIRWISE-DM (although not with significance in this experiment) even with a limited number of comparisons.

In conclusion, mistakes seem to occur less frequently when comparing two homogeneous groups with each other rather than two single behaviors. The *general* conditions in which users can benefit from our approach are that they a) increase the number of judged pairs by using groupwise comparison to their advantage or b) find comparisons by exploring the behavior space and giving more meaningful comparisons to the reward model. When users explore, they might provide less overall comparisons, but can ensure that the provided feedback is more informative given the goal behavior they want to teach the RL agent.

6. User Study

We carried out a user study aimed at evaluating the *efficiency*, *uselfulness*, and *ease of use* of the interactive-groupwise comparison (INTERACTIVE-UI) vs. the standard pairwise comparison (PAIRWISE-UI) in real usage by expert users. Participants in this evaluation were users who use RL in their daily work.

6.1. Study Design.

Participants.: We recruited ten expert users E1-E10 (3 female) with at least a 1-year experience with RL. Their average years of experience with RL was 2.5 years (SD=1.58). Eight of the experts have known RLHF before the study. E4 and E5 were introduced to it for the first time during the study.

Experimental Procedure: Experts were invited to conduct 2 RLHF sessions, each with different tools: the first one implementing PAIRWISE-UI; the second one implementing INTERACTIVE-UI. Hopper was chosen as the test environment as its goal is easy for participants to understand. Participants were instructed as follows: "*the center of the robot is the joint closest to the pointy end. The first priority is for the center of the robot to move to the right (moving to the left is worse than not moving at all). If the 2 robots are roughly tied on this metric, then the tiebreaker is how high the center is.*" With INTERACTIVE-UI, the participants were instructed to follow the Active Learning suggestions two-thirds of the time.

Experimental Design: The experimental design consisted of a within-subjects design with one independent variable that has the following two levels: PAIRWISE-UI and INTERACTIVE-UI. We counterbalanced the order of using the tools.

Experimental Protocol: All sessions were held in a lab setting (Fig. 9), using a Firefox browser on a Ubuntu desktop with a 27-inch retina display (2560 x 1440, 60 fps) and a commodity GPU (NVIDIA GeForce RTX 2080). Each session began with a 30-minute training to ensure participants can get familiar with the usage of both tools. The first 10 minutes of training consisted of a

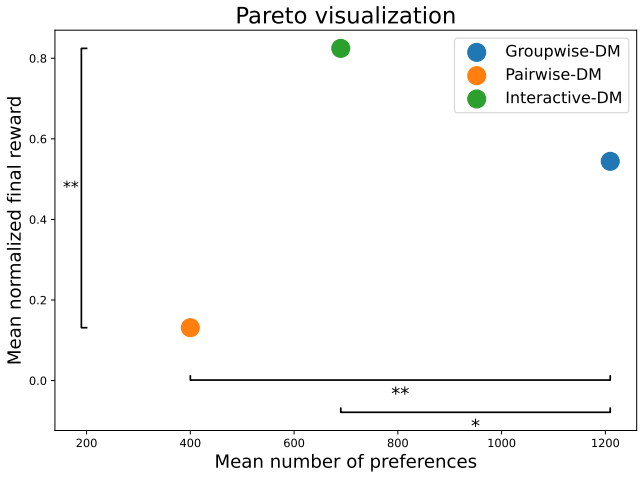


Figure 7: Pareto visualization: interactive-groupwise comparison (INTERACTIVE-DM) gives more useful feedback than both pairwise comparison (PAIRWISE-DM) and active-learning-only groupwise comparison (GROUPWISE-DM). The figure shows the mean normalized rewards vs. the mean number of preferences across all environments and all runs. We can see that the INTERACTIVE-DM is significantly better than PAIRWISE-DM in terms of normalized final reward across the environments. The reward of Groupwise-DM is in between the other two comparison types but without statistical significance. However, INTERACTIVE-DM only uses slightly more preferences than PAIRWISE-DM but not with significance. Whereas, GROUPWISE-DM utilizes significantly more preferences than both PAIRWISE-DM and INTERACTIVE-DM.

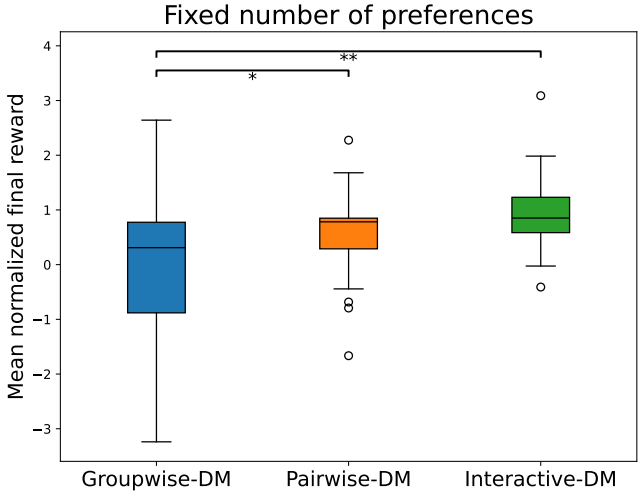


Figure 8: If we fix the number of preferences and run the simulation again, GROUPWISE-DM does not achieve higher average rewards than PAIRWISE-DM. The advantage of groupwise comparison is that more preferences can be elicited within the same time. However, INTERACTIVE-DM outperforms both approaches even with a constant number of preferences. The boxes denote the inter quartile range and the black line within the box denotes the median.

tutorial giving an introduction to the tools. Then, participants freely tested the environments and played with the tools. After the training, participants carried out the 2 RLHF sessions. Each session lasted about 35 minutes. The time for giving feedback was fixed and the participants performed 7 3-minute-rounds of giving feedback (a session with one tool had 21 minutes of user work). After each round, the models were retrained for about 2 minutes and new videos to be analyzed were generated. Each tool recorded the number of preferences and logged the training performance measured on the true reward. A researcher observed each session. After completing these sessions, experts completed a questionnaire and had a post-interview with the researcher lasting about 20 minutes. Each study lasted approximately 2 hours. We compensated participants with a voucher of 30 Euro.

6.2. Results. Overall ratings: The overall results of the questionnaire, illustrated in Fig. 10, show that our INTERACTIVE-UI comparison rated better than PAIRWISE-UI in terms of *efficiency* and *usefulness* . The participants rated PAIRWISE-UI higher in terms of *ease-of-use*.

Efficiency and accuracy of feedbacks: For each approach, we counted the number of preferences that the participants provided (see Fig. 11). All but one of the experts gave more preferences through INTERACTIVE-UI than PAIRWISE-UI. On average, the users gave 86.7% more preferences over the same time when using INTERACTIVE-UI. We also evaluated the number of mistakes that the experts made in these preferences based on the true reward we have for the task. Experts generally have a lower error rate when using INTERACTIVE-UI (10.8% error rate of the preferences) compared to PAIRWISE-UI (12.8 % of the preferences). To sum up, INTERACTIVE-UI yields more preferences in total, and the error rate is lower than PAIRWISE-UI. Often following our initial instructions, participants found comparisons with the exploration view on average one third of the time. They were often successful with lower error rates, more preferences, and better policies.

Quality of trained policies: The average reward from INTERACTIVE-UI is 1043, which is better than 648 from PAIRWISE-UI or an average 60.9% higher reward for INTERACTIVE-UI. However, the sample size of the study is quite low and we can not claim statistical significance. In seven out of ten sessions, INTERACTIVE-UI produced better policies than PAIRWISE-UI after the same time of giving feedback. During these seven sessions, the Hopper learned to move forward and achieved higher scores compared to PAIRWISE-UI. INTERACTIVE-UI yielded the four best policies with the best one having a true reward of over 2500.

Expert comments: All but one of the participants agreed that INTERACTIVE-UI was efficient to work with: e.g., “It is efficient because I have a global overview."; “Yes, more control over similar cases"; “Yes, I can control what I wanna compare". Only participant E1 found PAIRWISE-UI efficient because of its simplicity: “It’s efficient because it only provides 3 options" and INTERACTIVE-UI not efficient: “Not really, although I get more detailed visualization and more options, it is not efficient compared to only 3 options." 7 experts felt negative about the *information* of PAIRWISE-UI the other 3 were neutral. E.g., E3 stated: “Local, limited, does the job but minimally, so it’s slightly boring and repetitive". All participants gave positive assessments on the information provided

Figure 9: A participant performing the task using a Firefox browser on a Ubuntu desktop with a 27-inch retina display.

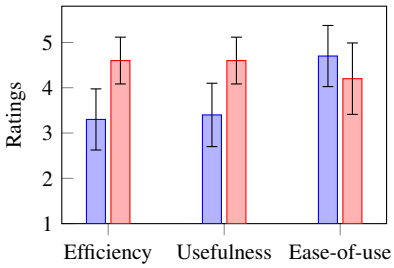


Figure 10: The ratings of participants on a 5-point Likert scale. The vertical lines represent the mean +- the standard deviation. The experts find the interactive-groupwise (*I*) approach more efficient and useful than the pairwise (*P*) approach.

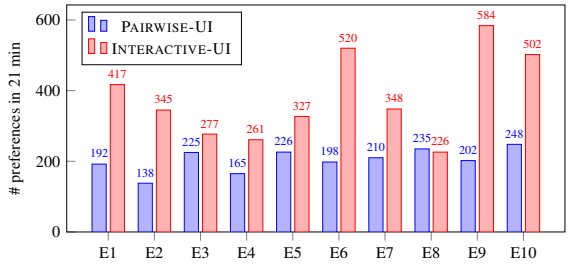


Figure 11: The number of preferences of participants (E1-E10) per type of comparison: PAIRWISE-UI vs. INTERACTIVE-UI. The vast majority of the experts E1-E10 is able to produce more preferences in the same amount of time with INTERACTIVE-UI than with PAIRWISE-UI.

by INTERACTIVE-UI, e.g., “More inclusive of the bigger picture, more complete visualization 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."; and “Yes, the tool suggestions [were] nice to select better videos.".

Three of the experts expressed neutral feelings toward the controllability of PAIRWISE-UI while the other seven felt negatively about it: “It is simple, but not helpful in exploring the behaviors"; “It does not allow to see a lot of behaviors.", and “The pairwise tool didn’t allow much exploration. I imagine that mistakes in the pairwise tool would be quite costly.". Whereas nine of the experts stated that they felt to have more control with INTERACTIVE-UI.

INTERACTIVE-UI was experienced as more difficult to use. For example, one participant stated: “[The way of comparing clips in INTERACTIVE-UI] needs several rounds to get used to." Even participant E7, who got the highest score of all experiments and the second highest score of the pairwise experiments with his PAIRWISE-UI session, said: “It was quick to select the best video and less cognitively demanding." about PAIRWISE-UI. About INTERACTIVE-UI, they said: “I could do more comparisons at the same time and choose better videos, but it was more cognitively demanding.".

Summary of the expert study: INTERACTIVE-UI offers efficient and flexible functionality, but it requires more cognitive effort. On the other hand, PAIRWISE-UI is simpler, with limited exploration and control capabilities. These trade-offs emphasize the importance of considering task complexity and user cognitive load in future iterations of comparison tools.

7. Discussion

Despite of the advantages that the exploration view and the possibility of groupwise comparison yield, there are some limitations.

Giving the user more power to provide a lot of quality feedback also gives them the power to provide more meaningless feedback and more noise if they do not pay attention or work in very broad strokes only. The user needs to strike a careful balance between being quick and providing varied and correct feedback. The cognitive load is increased in our interactive visualization compared to a tool that does not allow for exploration.

The scalability of our design is a clear limitation of the work. While our system enables user agency and exploration, it is limited in the radial design and would potentially not scale well to extremely large (e.g. LLMs) behavior spaces. Our specific design works better for smaller cases. However, enabling user agency and exploration could also be beneficial in those other cases.

One limitation of our design is that the visualization of the behavior space is quite abstract and remains so until a behavior or a group of behaviors is selected. A feature that shows the representative behaviors within a group could be worthwhile as it lets users see at a glance what the group is about before selecting it. As another idea, one of the participants suggested showing a preview of the clips next to the mouse when hovering over a behavior in the exploration view. In our current design these features are not supported. However, they could bring a lot of improvement in a future iteration of the design.

In summary, for ensuring the effectiveness of our approach, the sample from the behavior space should not be too large (below 200 behaviors sampled in a round is best), and users should be concentrated and well-trained. Some design features that benefit our approach could be tested in future works.

8. Conclusion

We presented a novel approach to interactive-groupwise comparison of behaviors for RLHF. Our interactive visualizations includes a hierarchical radial chart and edge bundling to aid in exploring and analyzing behaviors for comparison. Evaluations show that the return of the trained policy improve by 69.34% in respect to baseline. The evaluation study carried out with experts show that our approach increases the number of elicited preferences by 86.7% within the same time frame, highlighting its efficiency. Expert interviews indicate that our approach empowers users to control their exploration and gain a comprehensive understanding of the context.

When comparing interactive-groupwise comparison to the standard pairwise comparison, there is a trade-off to consider. Our approach provides greater efficiency, but may require a longer training time to familiarize users with its visual analysis features. It demands more cognitive effort during usage. However, users who tried our approach find that it offers efficient and flexible functionality, as evidenced in Section 6. In conclusion, visualization research can contribute to the training process of AI models (like RLHF) by designing interfaces that allow for more user agency and that enable users to take advantage of their cognitive capabilities.

9. Additional Information

Corresponding Author: Prof. Antti Oulasvirta, *antti.oulasvirta@aalto.fi*

Data Availability Statement: The code used to reproduce the experiments is available at: https://github.com/jankomp/interactive_rlhf. Data from the user study can be made available upon reasonable request to the corresponding author.

Conflict of Interest: The authors declare no conflict of interest.

Funders: JK, SD, and AO were supported by the Research Council of Finland (FCAI: 328400, 345604, 341763; Subjective Functions 357578) and the ERC (AdG project Artificial User: 101141916.). TW was supported by SeRC - the Swedish e-Science Research Centre.

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). 3

[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. 3

[ASS11] AKROUR R., SCHOENAUER M., SEBAG M.: Preference-based policy learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11* (2011), Springer, pp. 12–27. 3

[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, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23* (2012), Springer, pp. 116–131. 3

[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. 3

[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://api.semanticscholar.org/CorpusID:209376771. 2

[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 Trans. Interact. Intell. Syst. 11*, 3–4 (Sept. 2021). URL: https://doi.org/10.1145/3439333, doi:10.1145/3439333. 4

[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. doi:10.1109/TVCG.2017.2744818. 4

[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://api.semanticscholar.org/CorpusID:258833251. 2

[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. 3

[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. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13406, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13406, doi:https://doi.org/10.1111/cgf.13406. 4

[BZSA18] BERNARD J., ZEPPELZAUER M., SEDLMAIR M., AIGNER W.: Vial: a unified process for visual interactive labeling. *The Visual Computer 34* (09 2018). doi:10.1007/s00371-018-1500-3. 4

[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, arXiv:1704.02147. 4

[CAL94] COHN D., ATLAS L., LADNER R.: Improving generalization with active learning. *Machine Learning 15*, 2 (May 1994), 201–221. Publisher: Springer Science and Business Media LLC. URL: http://link.springer.com/10.1007/BF00993277, doi:10.1007/bf00993277. 6

[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://api.semanticscholar.org/CorpusID:4787508. 2, 3, 4, 6, 7, 8

[DE84] DAY W. H., EDELSBRUNNER H.: Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification 1*, 1 (1984), 7–24. 4

[DKF22] DANIELS-KOCH O., FREEDMAN R.: The expertise problem: Learning from specialized feedback, 2022. URL: https://arxiv.org/abs/2211.06519, arXiv:2211.06519. 2

[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. 3

[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://api.semanticscholar.org/CorpusID:264306078. 2

[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). 3

[ETT12] EREZ T., TASSA Y., TODOROV E.: Infinite-horizon model predictive control for periodic tasks with contacts. 5

[FHCP12] F"URNKRANZ J., H"ULLERMEIER E., CHENG W., PARK S.-H.: Preference-based reinforcement learning: A formal framework and a policy iteration algorithm. *Machine Learning 89* (2012), 123–156. 3

[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. 3

[FPS*23] FANG C., PETERNEL L., SETH A., SARTORI M., MOMBAUR K., YOSHIDA E.: Human modeling in physical human-robot interaction: A brief survey. *IEEE Robotics and Automation Letters* (2023). 7

[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). 2, 3

[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, arXiv:2211.11972. 4

[HA13] HARRIOTT C. E., ADAMS J. A.: Modeling human performance for human–robot systems. *Reviews of Human Factors and Ergonomics 9*, 1 (2013), 94–130. 7

[HAMA19] HENTOUT A., AOUACHE M., MAOUDJ A., AKLI I.: Human–robot interaction in industrial collaborative robotics: A literature review of the decade 2008–2017. *Advanced Robotics 33*, 15-16 (2019), 764–799. 7

[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://api.semanticscholar.org/CorpusID:40550. 5, 6

[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. 3

[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. 4, 5

[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. 4, 5

[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. 3

[KL83] KRUSKAL J. B., LANDWEHR J. M.: Icicle plots: Better displays for hierarchical clustering. *The American Statistician 37*, 2 (1983), 162–168. URL: http://www.jstor.org/stable/2685881. 4

[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. 4

[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. doi:10.1109/PacificVis60374.2024.00051. 3

[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://api.semanticscholar.org/CorpusID:257102772. 2

[LYWY24] LIU S., YANG W., WANG J., YUAN J.: Visualization for artificial intelligence. doi:10.1007/978-3-031-75340-4. 3

[Mah96] MAHADEVAN S.: Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning 22*, 1 (1996), 159–195. 2, 3

[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. 3

[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://api.semanticscholar.org/CorpusID:205242740. 2

[MLB*23] METZ Y., LINDNER D., BAUR R., KEIM D. A., EL-ASSADY M.: Rlhf-blender: A configurable interactive interface for learning from diverse human feedback. *arXiv abs/2308.04332* (2023). URL: https://api.semanticscholar.org/CorpusID:260704198. 2, 3

[MR93] MAĆKIEWICZ A., RATAJCZAK W.: Principal components analysis (pca). *Computers & Geosciences 19*, 3 (1993), 303–342. 4

[NR*00] NG A. Y., RUSSELL S., ET AL.: Algorithms for inverse reinforcement learning. In *ICML* (2000), vol. 1, p. 2. 3

[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. 2, 3

[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. 3

[SB18] SUTTON R. S., BARTO A. G.: *Reinforcement Learning: An Introduction*. MIT Press, 2018. 2, 3

[SC07] SALVADOR S., CHAN P.: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis 11*, 5 (2007), 561–580. 4

[Sch11] SCHULZ H.-J.: Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications 31*, 6 (2011), 11–15. doi:10.1109/MCG.2011.103. 5

[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://api.semanticscholar.org/CorpusID:515925. 2

[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://api.semanticscholar.org/CorpusID:262824780. 2

[SSPS21] SILVER D., SINGH S., PRECUP D., SUTTON R. S.: Reward is enough. *Artificial Intelligence 299* (2021), 103535. 3

[SZLS23] SHI X., ZHANG J., LIANG Z., SENG D.: Maddpgviz: a visual analytics approach to understand multi-agent deep reinforcement learning. *J. Vis. 26*, 5 (May 2023), 1189–1205. URL: https://doi.org/10.1007/s12650-023-00928-0, doi:10.1007/s12650-023-00928-0. 3

[SZWO24] SHI D., ZHU S., WEINKAUF T., OULASVIRTA A.: Interactive reward tuning: Interactive visualization for preference elicitation. 3

[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. 7

[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). 5

[TTD12] TRÜMPER J., TELEA A. C., DÖLLNER J.: Viewfusion: Correlating structure and activity views for execution traces. In *TPCG* (2012), Citeseer, pp. 45–52. 4

[VDMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9*, 11 (2008). 4

[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. 4

[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. 3

[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://api.semanticscholar.org/CorpusID:265352136. 2

[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. 4

[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. 3

[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. 4

[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. 3

[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://api.semanticscholar.org/CorpusID:267412958. 2

[YLWL24] YANG W., LIU M., WANG Z., LIU S.: Foundation models meet visualizations: Challenges andopportunities. *Computational Visual Media 10* (05 2024), 399–424. doi:10.1007/s41095-023-0393-x. 3

[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://api.semanticscholar.org/CorpusID:265608723. 2

[ZCBD22] ZHANG D., CARROLL M., BOBU A., DRAGAN A.: Time-efficient reward learning via visually assisted cluster ranking. *arXiv preprint arXiv:2212.00169* (2022). 2, 3, 4, 6

[Zel13] ZELAZO P. D.: *The Oxford Handbook of Developmental Psychology, Vol. 1: Body and Mind*. Oxford University Press, 2013. 4

[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. *arXiv Preprint arXiv:1802.07606* (2018). 3

[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). 2, 3

[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. doi:10.1109/TVCG.2024.3392587. 3