Designing Deep Reinforcement Learning for Human Parameter Exploration

Hugo Scurto, Bavo van Kerrebroeck, Baptiste Caramiaux, Frédéric Bevilacqua

Designing Deep Reinforcement Learning for Human Parameter Exploration

HUGO SCURTO, STMS Lab, IRCAM–CNRS–Sorbonne Université, France
BAVO VAN KERREBROECK, STMS Lab, IRCAM–CNRS–Sorbonne Université, France
BAPTISTE CARAMIAUX, CNRS–LRI, Université Paris-Sud, France
FRÉDÉRIC BEVILACQUA, STMS Lab, IRCAM–CNRS–Sorbonne Université, France

Software tools for generating digital sound often present users with high-dimensional, parametric interfaces, that may not facilitate exploration of diverse sound designs. In this paper, we propose to investigate artificial agents using deep reinforcement learning to explore parameter spaces in partnership with users for sound design. We describe a series of user-centred studies to probe the creative benefits of these agents and adapting their design to exploration. Preliminary studies observing users' exploration strategies with parametric interfaces and testing different agent exploration behaviours led to the design of a fully-functioning prototype, called Co-Explorer, that we evaluated in a workshop with professional sound designers. We found that the Co-Explorer enables a novel creative workflow centred on human-machine partnership, which has been positively received by practitioners. We also highlight varied user exploration behaviors throughout partnering with our system. Finally, we frame design guidelines for enabling such co-exploration workflow in creative digital applications.

CCS Concepts: • **Human-centered computing → Sound-based input / output**; • **Applied computing → Sound and music computing**;

Additional Key Words and Phrases: Interaction Design, Machine Learning, Audio/Video.

1 INTRODUCTION

Reinforcement learning defines a computational framework for the interaction between a learning agent and its environment [73]. The framework provides a basis for algorithms that learn an optimal behaviour in relation to the goal of a task [97]. For example, reinforcement learning was recently used to learn to play the game of Go, simulating thousands of agent self-play games based on human expert games [92]. The algorithm, called *deep reinforcement learning*, leveraged advances in deep neural networks to tackle learning of a behaviour in high-dimensional spaces [71]. The autonomous abilities of deep reinforcement learning agents let machine learning researchers foresee prominent applications in several domains, such as transportation, healthcare, or finance [65].

Yet, one important current challenge for real-world applications is the ability for reinforcement learning agents to learn from interaction with human users. The so-called *interactive reinforcement learning* framework has been shown to hold great potential to build autonomous systems that are centered on human users [3], such as teachable and social robots [98], or assistive search engines [9]. From a machine learning perspective, the main challenge lies in learning an optimal behaviour from small, non-stationary amounts of human data [57]. From a human-computer

Authors' addresses: Hugo Scurto, STMS Lab, IRCAM–CNRS–Sorbonne Université, 1 Place Igor Stravinsky, Paris, France, Hugo.Scurto@ircam.fr; Bavo Van Kerrebroeck, STMS Lab, IRCAM–CNRS–Sorbonne Université, 1 Place Igor Stravinsky, Paris, France, Bavo.VanKerrebroeck@ircam.fr; Baptiste Caramiaux, CNRS–LRI, Université Paris-Sud, Bât 650, Rue Noetzlin, Orsay, F-91400, Paris, France, baptiste.caramiaux@lri.fr; Frédéric Bevilacqua, STMS Lab, IRCAM–CNRS–Sorbonne Université, 1 Place Igor Stravinsky, Paris, France, Frederic.Bevilacqua@ircam.fr.

interaction perspective, an important challenge consists in supporting human appropriation of agents' autonomous behaviours in relation to complex human tasks [95].

Our interest lies in investigating interactive reinforcement learning for human creative tasks, where a goal might not be well-defined by human users a priori [82]. One such case of a human creative task is *exploration* [45]. Exploration consists in trying different solutions to address a problem, encouraging the co-evolution of the solution and the problem itself [29]. For example, designers may produce several sketches of a product to ideate the features of its final design, or test several parameter combinations of a software tool to create alternative designs in the case where the product has a digital form. The creative, human-centred, use case of exploration thus fundamentally differs from standard, machine-centred, reinforcement learning use cases, where a problem is implicitly defined as a goal behaviour, before the agent actually learns to find an optimal solution [97]. Here, one could expect that humans would prefer agent autonomous behaviours—provoking surprise and discovery along exploration—over the learning of one optimal solution—forcing human users to teach agents one optimal behaviour.

In this paper, we aim at designing an interactive reinforcement learning system supporting human creative exploration. This question is addressed in the application domain of sound design, where practitioners typically face the challenge of exploring high-dimensional, parametric sound spaces. We propose a user-centred design approach with expert sound designers to steer the design of such a system and conceptualize exploration within this context. We conducted two case studies to evaluate two prototypes that we developed. The final prototype specifically designed deep reinforcement learning to foster human exploration. Specifically, it employs reinforcement learning as an algorithmic approach to dynamically suggest different sounds to users based on their feedback data—thus possibly contributing to their exploration process. Therefore, contrary to typical reinforcement learning-based tools, it does not aim at creating fully-trained agents that could be optimally reused by users across different sessions. Our overall proposed methodology thus radically differs from standard reinforcement learning approaches.

Our findings led to contributions at several levels. On the conceptual side, we were able to characterize different user approaches to exploration, and to what we have called *co-exploration*—exploration in cooperation with an interactive reinforcement learning agent. These range from analytical to spontaneous in the former case, and from user- to agent-as-leader in the latter. On the technical side, a user-centered approach let us adapt a deep reinforcement learning algorithm to the case of co-exploration in high-dimensional parameter spaces. This notably required creating additional interaction modalities to user reinforcement, jointly with an autonomous exploration behaviour for the reinforcement learning agent. Crucially, our qualitative results suggest that the resulting agent—especially its non-optimal nor predictable behaviour—may be well experienced by human users leading parameter exploration, in more creative ways than random conditions. Lastly, on the design side, we extracted a set of important challenges that we deem critical for joint HCI and machine learning design in creative applications. These include: (1) engaging users with machine learning, (2) foster diverse creative processes, and (3) steer users outside comfort zones.

2 RELATED WORK

In this section, we review related work on machine learning in the field of Human-Computer Interaction, encompassing creativity support tools, interactive machine learning, and interactive reinforcement learning, with a focus on human exploration.

2.1 Creativity Support Tools

Human creative exploration of multidimensional parameter spaces has long been studied in the NIME community (acronym for New Interfaces for Musical Expression, originally emerging as

a workshop at CHI [79]). Techniques for mapping input user parameters (e.g., gestural control data) to output multidimensional parameters (e.g., sound synthesis parameters) were developed to support exploration through new digital music instruments [48, 49]. Perceptual descriptors of sound enabled to reduce dimensionality of sound spaces, thus providing users with more intuitive interfaces to explore sound spaces [85]. Yet, besides a few exceptions [35], the creative process of human users leading parameter exploration with computer music software has still been hardy investigated from a research perspective.

Creativity support tools have long focused on human exploration as a central process to human creative work [91]. Design guidelines for supporting exploration were developed, which include aiming at simple interfaces for appropriating the tool and getting into sophisticated interaction more easily [28]. Flexible interaction modalities that can adapt to users' very own styles of thinking and creating may also be required [82]. In particular, parameter space exploration remains a current challenge for HCI research [16]. Recently, creativity-oriented HCI researchers underlined the need to move toward interdisciplinary research collaborations [77].

Machine learning was in this sense examined for its implications in design [58] and identified as an opportunity for user experience [30, 104, 105]. Yet, a large body of work in the machine learning research community has so far focused on constructing autonomous algorithms learning creative behaviour from large amounts of impersonal data—falling under the name of computational creativity [101]. While this have allowed the building of powerful tools and models for creation, one may be concerned in the question of how to include human users in the design of such models to support human-computer co-creation [53].

Davis et al. proposed a model of creativity that explicitly considers the computer as an enactive entity [23]. They notably stressed the potential of combining creativity support tools with computational creativity to enrich a collaborative process between the user and the computer [23]. The Drawing Apprentice, a co-creative agent that improvizes in real-time with users as they draw, illustrates their approach [22]. While their user study confirms the conceptual potential of building such artistic computer colleagues, its technical implementation remains specific to the use case at stake—*e.g.*, drawing. We propose to jointly design a conceptual and technical framework that could be easily transferable to other application domains—potentially realizing general mixed-initiative co-creativity [47, 106].

2.2 Interactive Machine Learning

Interactive machine learning [31] allows human users to build customized models by providing their own data examples—typically a few of them. Not only users can customize training examples, but they are also allowed to directly manipulate algorithm parameters [54, 102], as well as to receive information on the model's internal state [4, 76]. Applications in HCI cover a wide range of tasks, such as handwriting analysis [90], recommender systems [5], or prioritising notifications [6]. Interactive machine learning mainly builds on supervised learning, which defines a computational framework for the learning of complex input-output models based on example input-output pairs. The "human-in-the-loop" approach to supervised learning critically differs from the computational creativity approach, which typically relies on huge, impersonal databases to learn models [43].

Interactive machine learning is one such example of a generic framework for human-computer co-creation [3]. The technical framework was successfully applied across several creative domains, such as movement interaction design [39, 42, 108], web page design [63] or video games [56]. Specifically, research studying users building customized gestural controllers for music brought insight on the creative benefits of interacting with machine learning [34]. Not only were users able to accomplish their design goal—*e.g.*, demonstrating a given gesture input for controlling a given sound parameter output—, but they also managed to explore and rapidly prototype alternative

designs by structuring and changing training examples [35]. These patterns were reproduced by novice users who gained accessibility using examples rather than raw parameters as input [55]. The algorithms' sometimes surprising and unexpected outcomes favoured creative thinking and sense of partnership in human users [33].

Typical workflows in interactive machine learning tend to iterate on designing training examples that are built from a priori representative features of the input space to support exploration. Yet, in some creative tasks where a problem definition may be found only by arriving at a solution [29, 83], it might be unfeasible for users to define, a priori, such representative features of the final design [55]. Other approaches proposed methods to release such contraints, for example by exploring alternative machine learning designs by only defining the limits of some parameter space [87]. We propose to further investigate machine learning frameworks able to iteratively learn from other user input modalities, and explicitly considering mixed-initiative workflows, where systems autonomously adapt to users [26]. As reviewed in the next section, using interactive reinforcement learning offers such perspectives.

2.3 Interactive Reinforcement Learning

Interactive reinforcement learning defines a computational framework for the interaction between a learning agent, a human user, and an environment [3]. Specifically, users can communicate positive or negative *feedback* to the agent, in the form of a numerical reward signal, to teach it which action to take when in a certain environment state. The agent is thus able to adapt its behaviour to users, while remaining capable of behaving autonomously in its environment.

While user feedback has been used as input modality for applications in information retrieval [109], recommender systems [61], or affective computing [66], it was often included in algorithmic frameworks relying on pre-established, rule-based methods to provide users with adaptive behaviour. The data-driven abilities of reinforcement learning, in contrast, offers promising perspectives for open-ended, interactive applications that are centered on human users. In this sense, interactive reinforcement learning relies on small, user-specific data sets, which contrasts with the large, crowdsourced data sets used in creative applications in semantic editing [25, 62, 107]. Lastly, interactive approaches to reinforcement learning focuses on exploring agent actions based on human feedback on actions, which contrasts with the focus on optimising one parametric state based on user feedback over states—as used in Bayesian Optimisation [13, 67] or multi-armed bandits [68].

Interactive reinforcement learning has been recently applied in HCI [84], with promising applications in exploratory search [10, 44] and adaptive environments [40, 80]. Integrating user feedback in reinforcement learning algorithms is computationally feasible [94], helps agents learn better [57], can make data-driven design more accessible [68], and holds potential for rich human-computer collaboration [95]. Applications in Human-Robot Interaction informed on how humans may give feedback to learning agents [98], and showed potential for enabling human-robot co-creativity [36]. Recently, reinforcement learning has witnessed a rise in popularity thanks to advances in deep neural networks [71]. Powerful models including user feedback have been developed for high-dimensional parameter spaces [19, 99]. Design researchers have identified reinforcement learning as a promising prospective technique to improve human-machine "joint cognitive and creative capacity" [60].

We believe that interactive reinforcement learning—especially deep reinforcement learning—holds great potential for supporting creative tasks—especially exploration of high-dimensional parameter spaces. First, its computational framework, constituted by environment states, agent actions, and user feedback, remains fully generic [97], and thus potentially allows the design of

generic interaction modalities transferrable to different application domains. Second, the action-oriented, autonomous exploration behaviour intrinsic to reinforcement learning algorithms may be exploited to build a novel creative mixed-initiative paradigm, where the user and the agent would cooperate by taking actions that are "neither fully aligned nor fully in conflict" [20]. Finally, we consider that user feedback could be a relevant input modality in the case of exploration, notably for expressing on-the-fly, arbitrary preferences toward imminent modifications, as opposed to representative examples. As previously stated, this requires investigating a somewhat unconventional use of reinforcement learning: if previous works employed user feedback to teach agents an optimal behavior in relation to a task's goal, it is less obvious whether such an optimal behavior may be well-defined—or even exists—for human users performing exploration.

3 GENERAL APPROACH

In this section, we describe the general approach of our paper, applying interactive reinforcement learning for human parameter space exploration in the creative domain of sound design.

3.1 Problem Formulation

In this paper, we seek to address the following general research question: *How to design reinforcement learning to support human parameter exploration?* While many formulations of reinforcement learning (RL) could be imagined to address this question, our approach focused on an interactive use case of reinforcement learning, where feedback data is provided in real-time by a human user. We hypothesize that such formulation may be of interest for users leading parameter exploration.

We investigate the reinforcement learning problem in the context of classical sequential decision making. Let $\mathcal{S} = \{S\}$ denote the state space constituted by all possible parameter configurations $S = (s_1, ..., s_n)$ reachable by the agent, with n being the number of parameters, and $s_i \in [s_{min}, s_{max}]$ being the value of the i^{th} parameter living in some bounded numerical range. Let $\mathcal{A}(S) = \{A\}$ denote the corresponding action space as moving up or down one of the n parameters by one step a_i, except when the selected parameter equals one boundary value. As the agent selects actions and iteratively acts on parameters one by one, we assume that a human observes the state-action path and interactively provides positive of negative feedback, R, to the agent. The agent's goal is to maximize user feedback, which it does by learning a mapping between state and actions defined from user feedback. In order to allow such real-time human interaction with the currently-learning agent, we only consider on-policy settings of reinforcement learning—as opposed to off-policy settings, which separates a behaviour policy for real-time environment exploration from an estimation policy that learns from reinforcement [97]. Designing reinforcement learning for human parameter exploration thus consists in understanding what interactive mechanisms may be relevant for the agent to support human exploration.

Learning action values instead of state values constitutes the main reason for treating the task as an RL problem. By definition, human parameter exploration has a strong dependence between current state and action to the next state. It is precisely there that lies our rationale behind RL: to account for human actions in parameter state exploration. Conversely, Bayesian Optimisation (BO) algorithms do not take into account actions leading to states when exploring parameter states. As such, BO may be relevant for parameter search—e.g., optimization of a parametric state based on user feedback over states—, but not for parameter exploration—e.g., trial of parameter actions based on user feedback over actions. Contextual Bandits, on the other hand, only afford learning of action values from a small, discrete set of states. In our case study, we will show that learning in continuous state-action spaces is required to tackle parameter space exploration, and that reinforcement learning—especially its deep learning extension—suits this task. We will also observe that the interactive reinforcement learning formulation enables users to explore parameter spaces

by providing relatively small amounts of feedback data—an average of 235 feedback data during a typical session, with a large standard deviation corresponding to diverse user feedback behaviours. This is in contrast with the large datasets often required in more standard RL applications to continuous action spaces.

3.2 Application Domain

Sound design is an exemplary application domain for studying exploration—taking iterative actions and multiple steps to move from an ill-formed idea to a concrete realization [41]. Sonic exploration tasks can take myriad of forms: for example, composers explore various sketches of their musical ideas to write a final score; musicians explore different playing modes to shape an instrument's tone; sound designers explore several digital audio parameters to create unheard-of sounds [24, 72].

Most of today's digital commercial tools for sound synthesis, named Virtual Studio Technology (VST, see Fig. 1), still rely on complex interfaces using tens of technical parameters as inputs. These parameters often relate to the underlying algorithms that support sound synthesis, preventing users from establishing a direct perceptual relationship with the sound output. To that one may add the exponential number of parameter combinations, called presets, that eventually correspond to given sound designs. It is arguable that these interfaces may not be the best to support human exploration: as the perceptual outcome of acting on a given parameter may rapidly become unpredictable, they may hinder user appropriation [82, 91].

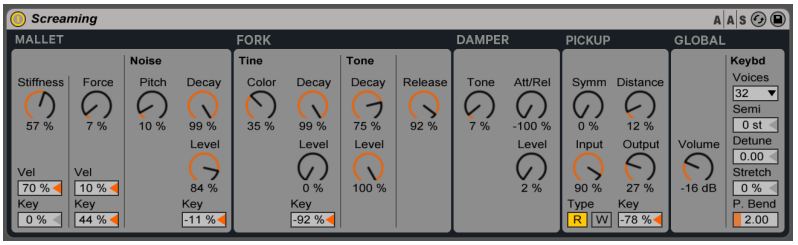

Fig. 1. A typical VST interface in *Ableton Live* (sound design software), containing many technical parameters.

By formalizing human parameter space exploration as an interactive reinforcement learning problem, we seek to tackle both issues at once. First, human navigation in high-dimensional parameter spaces may be facilitated by the reinforcement learning computational framework, made of sequences of states, actions, and rewards. Second, human creativity may be stimulated by the action-oriented, autonomous behaviour of reinforcement learning algorithms, suggesting other directions or design solutions to users along exploration.

3.3 Method

We adopted a user-centered approach to lead joint conceptual and technical work on interactive reinforcement learning for parameter space exploration. We decided to work with expert sound designers to get feedback on the creative task of parameter space exploration as it is led by specialized users. This qualitative understanding would in turn inform the design of interactive reinforcement learning in its application to sound design. Two design iterations—a pilot study and an evaluation workshop—were conducted over the course of our research. Two prototypes were designed and developed—one initial reinforcement learning prototype, and the *Co-Explorer*, our final deep reinforcement learning prototype. The process thus includes sequentially:

- Prototype 1: Implementing a reinforcement learning algorithm that learns to explore sound parameter spaces from binary human feedback

- Pilot study, Part 1: Observing and interviewing participants exploring sound spaces with standard parametric interfaces
- Pilot study, Part 2: Observing and interviewing participants using our initial reinforcement learning prototype to explore a sound space
- Prototype 2: Designing deep reinforcement learning in response to design ideas suggested by our pilot study, implementing it in the *Co-Explorer*
- Workshop, Part 1: Observing and discussing with participants using the Co-Explorer, our final prototype, in an exploration task related to discovery
- Workshop, Part 2: Observing and discussing with participants appropriating the Co-Explorer, our final prototype, in an exploration task related to creation

We worked with a total of 14 users (5 women, 9 men; all French) through the series of activities. From the 14 total, there were 2 who took part in all of the activities listed below, to testify of our prototype's improvements. Our users covered different areas of expertise in sound design and ranged from sound designers, composers, musicians, and artists to music researchers and teachers. Thus, they were not all constrained to one working methodology, one sonic practice or one application domain. Our motivation was to sample diverse approaches to exploration that sound design may provoke, in order to design a flexible reinforcement learning algorithm that may suit a variety of users' working styles [82].

4 PILOT STUDY

We organized a one-day pilot study with four of our expert participants. The aims of this pilot study were to: Observe approaches to exploration in standard parametric interfaces; Identify problems users experience; Introduce the reinforcement learning technology in the form of a prototype; Brainstorm ideas and possible breakdowns.

The study was divided in two parts: (1) parametric interface exploration, then (2) interactive reinforcement learning-based exploration. We conducted individual semi-structured interviews at the end of each part, having each participant do the study one by one. This structure was intended to bring each participant to become aware of their subjective experience of exploration [78]. Our intention was to open up discussions and let participants suggest design ideas about interactive reinforcement learning, rather than testing different algorithmic conditions in a controlled, experimental setup. We spent an average of 2 hours with each of our four participants, who covered different expertise in sound design (composition, sound design, interaction design, research).

4.1 Part 1: Parametric Interfaces

4.1.1 Procedure. In the first part of the study, participants were asked to find and create a sound preset of their choice using three different parametric interfaces with different number of parameters (respectively 2, 6, and 12, see Fig. 2). No reinforcement learning agent was used. We linked each interface to a different sound synthesis space (respectively created using FM synthesis[1], and one commercial VST from which we selected 6, then 12, parameters). Our goal was to investigate how the number of parameters on an interface might influence exploration of large perceptual spaces. As such, we used the FM synth because of the large perceptual space it offers relying on only two parameters.

Sound was synthesized continuously; participants' actions were limited to move the knobs using the mouse to explore the design space offered by all possible combinations. While we agree that tangible interfaces are extensively used by professional sound designers, we underline that the mouse remains used in other creative domains where real-time multi-dimensional control is needed

[1] *Frequency Modulation synthesis* (a classic algorithmic method for sound synthesis [18]).

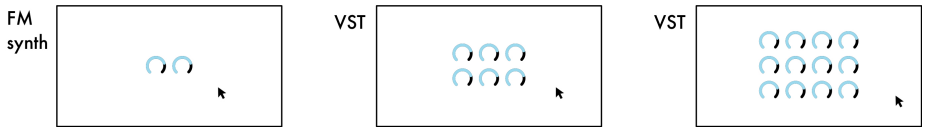

Fig. 2. Schematic view of the three parametric interfaces.

(e.g., graphic design). As such, we decided to use the mouse for this pilot experiment to study the general task of parameter space exploration. Knobs' technical names were hidden to test the generic effect of parameter dimensionality in interface exploration, and avoid any biases due to user knowledge of parameter function (which typically occur with labelled knobs). Interface order was randomized; we let participants spend as much time as they wanted on each interface to let them explore the spaces freely.

4.1.2 Analysis. We were interested in observing potential user strategies in parameter space exploration. We thus logged parameter temporal evolution during the task. It consists in an n-dimensional vector, with n being the number of parameters (respectively 2, 6, then 12). Sample rate was set to 100 ms, which is a standard value for interaction with sound and musical interfaces [52]. We used Max/MSP[2] and the MuBu[3] library to track user actions on parameters and record their evolutions. We used structured observation to study participants' interviews. This method was meant to provide a thorough qualitative analysis on user exploration strategies.

4.1.3 Observations. Qualitative analysis of parameter temporal evolution let us observe a continuum of approaches to parametric interface exploration. We call the first extremity of this continuum **analytical exploration**: this involves actioning each of the knobs one after the other over their full range. The second is called **spontaneous exploration**: this involves making random actions on the knobs. Figure 3 shows examples for each of these two approaches. One participant was consistently analytical over the three interfaces; one was consistently spontaneous over the three. The two others combined both approaches over the three interfaces.

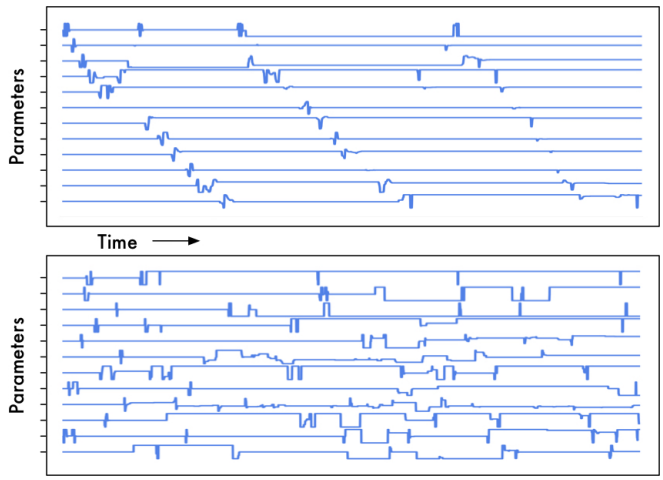

Fig. 3. Two user exploration strategies with a 12-dimensional parametric interface: Analytical (top) vs. spontaneous (bottom).

[2]https://cycling74.com/products/max/
[3]https://forum.ircam.fr/projects/detail/mubu/

Interview analysis let us map these approaches to different subgoals in exploration. The analytical approach concerns exploration of the interface at a parameter level: *"The strategy is to test [each knob] one by one to try to grasp what they do"*, one participant said. The goal of exploration is then related to building a mental map of the parameters to learn how to navigate in the design space. The spontaneous approach concerns exploration of the design space at a creative level: *"I moved the knobs more brutally and as a result of serendipity I came across into something different, that I preferred for other reasons..."*, another participant said. The goal of exploration is then related to discovering new parameter states leading to inspiring parts of the design space.

Discovery is critical to parameter space exploration. *"Once [the knobs] are isolated, you let yourself wander a bit more..."*, one participant analysed. Surprise is also important: *"To explore is to be in a mental state in which you do not aim at something precise"*, one participant said. Interestingly, we observed that participants often used words related to perceptual aspects rather than technical parameters. *"I like when you can get a sound that is... um... Consistent, like, coherent. And at the same time, being able to twist in many different ways. This stimulates imagination, often"*, one participant said. Two participants mentioned that forgetting the parametric interface may be enjoyable in this sense: *"I appreciate an interface that does not indicate [...], that has you go back into sound, so that you are not here reading things, looking at symbols..."*, one participant said.

All participants reported being hindered in their exploration by the parameter inputs of the three interfaces. As expected, the more parameters the interface contained, the larger the design space was, and the harder it was to learn the interface. *"For me, the most important difficulty is to manage to effectively organise all things to be able to re-use them."*, one participant said. Time must be spent to first understand, then to memorize the role of parameters, taking into account that their role might change along the path of exploration. This hampers participants' motivation, often restraining themselves to a subspace of the whole design space offered by the tool: *"after a while I was fed up, so I threw out some parameters"*, one participant said about the 12-knob interface.

Participants discussed the limitations encountered in the study in light of their real-world practice with commercial interfaces. Two participants mentioned using automation functions to support parameter space exploration. Such functions include randomizing parameter values, automating parameter modification over time, or creating new control parameters that *"speak more to your sensibility, to your ears, than to what happens in the algorithm"*, to cite one of the participants. Two participants also use factory presets to start exploration: *"I think that in some interfaces they are pretty well conceived for giving you the basis of a design space. Then it's up to you to find what parameters to move"*, one participant said. Two participants said that the graphical user interfaces, including parameter names, knob disposition, and visual feedback on sound, may help them manage to lead exploration of large parameter spaces.

4.2 Part 2: RL Agent Prototype

Results in first part let us identify different user approaches to parametric interface exploration, as well as different problems encountered in high-dimensional parameter spaces. In the second part, we were interested in having participants test the reinforcement learning technology in order to scope design ideas and possible breakthroughs in relation to exploration.

4.2.1 Implementation. We implemented an initial prototype for our pilot study, that we propose to call "RL agent" for concision purposes. The prototype lets users navigate through different sounds by only communicating positive or negative feedback to a reinforcement learning agent. The agent learns from feedback how to act on the underlying synthesis parameters in lieu of users (see Fig. 4). Formally, the environment is constituted by the VST parameters, and the agent iteratively acts on them. Computationally, we considered the state space $\mathcal{S} = \{S\}$ constituted by

all possible parameter configurations $S = (s_1, ..., s_n)$, with n being the number of parameters, and $s_i \in [s_{min}, s_{max}]$ being the value of the i^{th} parameter living in some bounded numerical range (for example, s_i can control the level of noise normalized between 0 and 1). We defined the corresponding action space $\mathcal{A}(S) = \{A\}$ as moving up or down one of the n parameters by one step a_i, except when the selected parameter equals one boundary value:

$$A(S) = \begin{cases} \pm a_i & \text{for } s_i \in]s_{min}, s_{max}[\\ +a_i & \text{for } s_i = s_{min} \\ -a_i & \text{for } s_i = s_{max} \end{cases}$$

An ε-greedy method defines the autonomous exploration behaviour policy of the agent—how it may take actions by exploiting its accumulated feedback while still exploring unvisited state-action pairs [97]. From a given state, it consists in having the agent take the best action with probability ε, and reciprocally, take a random action with probability $1 - \varepsilon$. For example, $\varepsilon = 1$ would configure an always exploiting agent—*i.e.,* always taking the best actions based on accumulated feedback—, while $\varepsilon = 0$ would configure an always exploring agent—*i.e.,* never taking into account the received feedback. Our purpose in this study was to examine whether different exploration-exploitation trade-offs could map to different user approaches to exploration. Finally, we propose that the user would be responsible for generating feedback. We directly mapped user feedback to the environmental reward signal R associated with a given state-action pair (S, A). The resulting formalization—where an agent takes actions that modify the environment's state based on feedback received from a user—defines a generic interactive reinforcement learning problem.

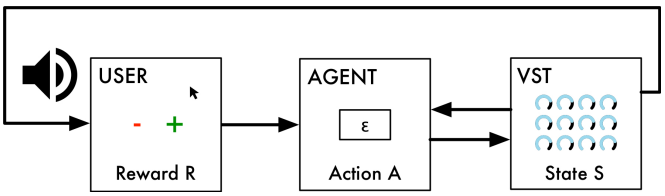

Fig. 4. Our RL agent prototype. Users can only provide feedback to the agent, which acts on hidden VST parameters.

We implemented *Sarsa*, which is a standard algorithm to learn how to act in many different environment state, *i.e.,* for each given parameter configuration [97]. It differs from multi-armed bandits, which learns how to act in one unique environment state [68]. Importantly, as evoked in Section 1, *Sarsa* was designed to learn one optimal behaviour in relation to the goal of a task. Our purpose in this study was to scope the pros and cons of such a standard reinforcement learning algorithm for human exploration tasks, judging how it may influence user experience, and framing how it may be adapted by design to support exploration. The convergence of the *Sarsa* algorithm in an interactive setup where users provide feedback was evaluated in a complementary work [86].

We used the largest VST-based 12-parameter space of the first part ($n = 12$) as the environment of our prototype. Because *Sarsa* is defined on discrete state spaces, each parameter range was discretized in three normalized levels ($s_i \in \{0, 0.5, 1\}, a_i = 0.5; 0 \leq i \leq n$). Our goal was to investigate how the RL agent could help exploration of large perceptual spaces. As such, we opted for more (twelve) parameters and less (three) discrete levels to design the largest perceptual space suitable for our RL-agent prototype. Although this would have been a design flaw in a perceptual experiment on typical VSTs, this allowed for obvious perceptual changes, which was required to investigate feedback-based interaction with a large variety of sounds.

4.2.2 Procedure. Our participants were asked to find and create a sound preset of their choice by communicating feedback to three different agents with different exploration behaviours (respectively $\varepsilon = 0$; $\varepsilon = 1$; and $\varepsilon = 0.5$). Sound was synthesized continuously, in a sequential workflow driven by the agents' algorithmic functioning. At step t, participants could listen to a synthesized sound, and give positive or negative feedback by clicking on a two-button interface (Fig. 5). This would have the agent take an action on hidden VST parameters, modify the environment's state, and synthesize a new sound at step $t + 1$. Participants were only told to give positive feedback when the agent took an action getting them closer to a sound that they enjoy, and negative feedback when it moves away from it. They were not explained the agent's internal functioning, nor the differences between the three agents. Each session started with a fully-untrained agent. The starting state for $t = 0$ was randomly selected. Agent order was randomized; we asked participants to spend between 5 and 10 minutes with each.

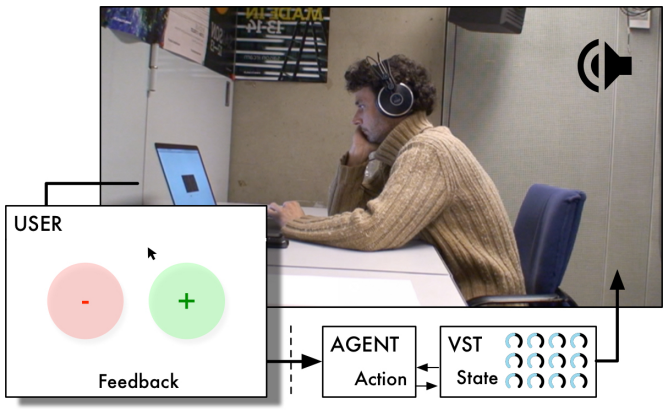

Fig. 5. One of our four participants using a two-button interface to communicate binary feedback to the RL agent prototype in the pilot study.

4.2.3 Analysis. We logged all participant actions in the graphical user interface. It consisted in timed onsets for positive feedback on the one hand, and negative feedback on the other hand. We also logged parameter temporal evolution to observe how the RL agent would act on parameters following user feedback. We used structured observation to study participants' interviews and discussions led at the end of the pilot study.

4.2.4 Reactions. All participants reported forgetting synthesis parameters to focus on the generated sound. The simplicity and straightforwardness of the new interface benefited their exploration. *"There's always this sensation that finally you are more focused on listening to the sound itself rather than trying to understand the technology that you have under your hands, which is really great, yeah, this is really great"*, one participant said. Participants went on voluntarily for 5.9 minutes with each of the agents on average ($\sigma = 1.3$).

The computational framework defined by reinforcement learning was well understood by all participants. *"There's somewhat a good exploration design [sic], because it does a bit what you do [with the parametric interface], you move a thing, you move another thing..."*, one participant said. All participants enjoyed following agents' exploration behaviours, mentioning a playful aspect that may be useful for serendipity. Three participants in turn adapted their exploration to that of the agent: *"you convince yourself that the machine helps you, maybe you convince yourself that it*

is better... and after you go on exploring in relation to this", one participant said. Interestingly, one participant that was skeptical about partnering with a computer changed his mind interacting with the RL agent: *"We are all different, so are they"*, he commented, not without a touch of humor.

4.2.5 Uses of Feedback. Descriptive statistics informed on how participants used the feedback channel. Three participants gave feedback every 2.6 seconds on average ($\sigma = 0.4$), globally balancing positive with negative (average of 44.8% positive, $\sigma = 0.02$). The fourth participant gave feedback every 0.9 seconds on average ($\sigma = 0.07$) which was mostly negative (average of 17.2% positive, $\sigma = 0.02$). All participants reappropriated the feedback channel, quickly transgressing the task's instructions toward the two-button interface to fulfill their purposes. One participant used feedback to explore agents' possible behaviors: *"Sometimes you click on the other button, like, to see if it will change something, [...] without any justification at all"*, he commented. Another used the '-' button to tell the agent to *"change sound"*. Two participants also noticed the difference between feedback on sound itself, and feedback on the agent's behavior: *"there's the 'I don't like' compared to the sound generated before, and the 'I don't like it at all', you see"*, one of them said.

4.2.6 Breakdowns. Rapidly, though, participants got frustrated interacting with the RL agent. All participants judged that agents did not always reacted properly to their feedback, and were leading exploration at the expense of them: *"sometimes you tell 'I don't like', 'I don't like', 'I don't like', but it keeps straight into it! (laughs)"*, one participant said. Contrary to what we expected, participants did not expressed a strong preference for any of the three tested agents. Only one participant noticed the randomness of the exploring agent, while the three other participants could not distinguish the three agents. This may be caused by the fact that the *Sarsa* algorithm was not designed for the interactive task of human exploration. Reciprocally, this may be induced by experiential factors due to the restricted interaction of our RL agent prototype, *e.g.,* preventing users to undo their last actions. Finally, two participants also complained about the lack of precision of the agent toward the generated sounds. This was induced by the tabular method that we used with the *Sarsa* algorithm, which required to discretize the VST parameter space.

4.2.7 Design Implications. Participants jointly expressed the wish to lead agent exploration. They suggested different improvements toward our RL agent prototype:

- Express richer feedback to the agent (*e.g.*, differentiating *"I like"* from *"I really like"*)
- Control agent path more directly (*e.g.*, commanding the agent to go back to a previous state, or to some new unvisited state in the parameter space)
- Improve agent algorithm (*e.g.*, acting more precisely on parameters, reacting more directly to feedback)
- Integrate agent in standard workspace (*e.g.*, directly manipulating knobs at times in lieu of the agent)

Interestingly, one participant suggested moving from current sequential workflow (where the agent waits for user feedback to take an action on the environment's state) to an autonomous exploration workflow (where the agent would continuously take actions on the environment's state, based on both accumulated and instantaneous user feedback). Three participants envision that such an improved RL agent could be useful in their practice, potentially allowing for more creative partnerships between users and agents.

5 CO-EXPLORER

Our pilot study led us to the design of a final prototype, called Co-Explorer. We decided to first design new generic interaction modalities with RL agents, based on users' reactions with both parametric interfaces and our initial prototype. We then engineered these interaction modalities,

developing a generic deep reinforcement learning algorithm fostering human exploration over learning one optimal behaviour, along with a new specific interface for sound design.

5.1 Interaction Modalities

Our initial prototype only employed user feedback as its unique interaction modality. This limited our participants, who suggested a variety of new agent controls to support exploration. We translated these suggestions into new interaction modalities that we conceptualized under three generic categories: (1) user feedback, (2) state commands, and (3) direct manipulations (as shown in Fig. 6).

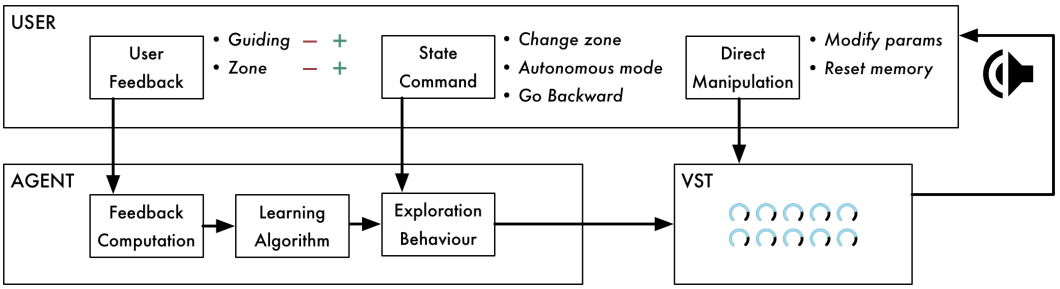

Fig. 6. Co-Explorer workflow. Users can have the agent explore parameters autonomously, communicating feedback and state commands to influence agent's actions. Reciprocally, they can directly explore parameters by hand, using a standard parametric interface. Users may be free to switch between these interactions modalities throughout their exploration.

5.1.1 User Feedback. Our design intention is to support deeper user customization of the parameter space, as suggested by our users in the pilot study, who wanted to "express richer feedback to the agent". We thus propose to enhance user feedback as defined in our initial prototype, distinguishing between *guiding* and *zone feedback*. Guiding feedback corresponds to users giving binary guidance toward the agent's instantaneous actions in the parameter space. Users can give either positive—*i.e.,* "keep going in that direction"—or negative guidance feedback—*i.e.,* "avoid going in that direction". Zone feedback corresponds to users putting binary preference labels on given state zones in the parameter space. It can either be positive—*i.e.,* "this zone interests me"—or negative—*i.e.,* "this zone does not interest me". Zone feedback would be used for making assertive customization choices in the design space, while guiding feedback would be used for communicating on-the-fly advice to the learning agent.

5.1.2 State Commands. Additionally, our design intention is to support an active user understanding of agent trajectory in the parameter space, as suggested by our users in the pilot study, who wanted to "control agent path more directly". We propose to define an additional type of interaction modality—we call them "state commands". State commands enable direct control of agent exploration in the parameter space, without contributing to its learning. We first allow users to command the agent to *go backward* to some previously-visited state. We also enable users to command the agent to *change zone* in the parameter space, which corresponds to the agent making an abrupt jump to an unexplored parameter configuration. Last but not least, we propose to let users start/stop an *autonomous exploration mode*. Starting autonomous exploration corresponds to letting the agent act continuously on parameters. As such, in autonomous exploration mode, the agent does not have to wait for feedback from the user to take actions. Thus, two different cases arise, at each time step. If the user gives feedback, then the next action is taken on that basis.

If the user does not give feedback, then the next action is taken based on past accumulated user feedback. Stopping autonomous exploration corresponds to going back to the sequential workflow implemented in our initial prototype, where the agent waits for user feedback before taking a new action on parameters.

5.1.3 Direct Manipulation. Lastly, our design intention is to augment, rather than replace, parametric interfaces with interactive reinforcement learning, as suggested by our users in the pilot study, who wanted to "integrate agent in standard workspace". We thus propose to add "direct manipulations" to support direct parameter modification through a standard parametric interface. It lets users explore the space on their own by only manipulating parameters without using the agent at all. It can also be used to take the agent to a given point in the parameter space—*i.e.,* "start exploration from this state"—, or to define by hand certain zones of interest using a zone feedback—*i.e.,* "this example preset interests me". Inversely, the parametric interface also allows to visualize agent exploration in real-time by observing how it acts on parameters.

A last, global interaction modality consists in *resetting agent memory*. This enables users to start exploration from scratch by having the agent forget accumulated feedback. Other modalities were considered, such as modifying the agent's speed and precision. Preliminary tests pushed us to decide not to integrate them in the Co-Explorer.

5.2 Deep Reinforcement Learning

As suggested by our users in the pilot study, who wanted to "improve agent algorithm", we developed a deep reinforcement learning agent at three intertwined technical levels. Our approach is based on *Deep TAMER* [99] for feedback formalization (Section 5.2.1), and learning algorithm (Section 5.2.2). Our original adaptations lie in exploration behaviour (Section 5.2.3), and the integration of our interaction modalities in the deep reinforcement learning framework (Section 5.3).

5.2.1 Feedback Formalization. One challenge consisted in addressing the non-stationarity of user feedback data along their exploration. We implemented *Deep TAMER*, a reinforcement learning algorithm suited for human interaction [99]. *Deep TAMER* leverages a feedback formalization that distinguishes between the environmental reward signal—*i.e.,* named R in the *Sarsa* algorithm of our initial prototype—and the human reinforcement signal—*e.g.,* feedback provided by a human user. This technique, already implemented in the *TAMER* algorithm [57], was shown to reduce sample complexity over standard reinforcement learning agents, while also allowing human users to teach agents a variety of behaviours.

Deep TAMER learns a user's reinforcement function by maximizing direct user feedback. This approach differs from conventional uses of RL, which seek to learn an optimal RL-policy by maximizing cumulative feedback. Yet, optimal RL-policies or fully-trained agents could not be suited to our application, since they make the assumption that users would provide consistent feedback all along the parameter exploration task. Despite being unusual from a RL perspective, we will show that this interactive RL formulation does suit our HCI application.

To deal with potential time delays between reinforcement communicated by the human and state-action paths made by the agent, *Deep TAMER* uses a weighting function $u(t)$ distributing credit over the sequence of lastly-visited state-action pairs. We set $u(t)$ similarly to the *Deep TAMER* implementation, that is, with a uniform distribution for the rewards received between 4 and 0.2 seconds before the most recent reward—*i.e.,* the most recent user feedback, as formulated in Section 5.1. We detail the differences between standard RL algorithms and *Deep TAMER* in Appendix A.

5.2.2 Learning Algorithm. Another challenge was to tackle learning in high-dimensional parametric spaces that are typical of our use case. *Deep TAMER* employs function approximation [97]

to generalize user feedback given on a subset of state-action pairs to unvisited state-action pairs. Specifically, a deep neural network is used to learn the best actions to take in a given environment state, by predicting the amount of user feedback it will receive [71, 99]. The resulting algorithm can learn in high-dimensional state spaces $\mathcal{S} = \{S\}$ and is robust to changes in discretization a_i of the space. For our application in sound design, we engineered the algorithm for $n = 10$ parameters. We normalized all parameters and set the agent's precision by discretizing the state space in one hundred levels ($s_i \in [0, 1], a_i = 0.01; 0 \leq i \leq n$).

A last challenge was to learn quickly from the small amounts of data provided by users during interaction. *Deep TAMER* uses a replay memory, which consists in storing the received human feedback in a buffer $\mathcal{D}$, and sampling repeatedly from this buffer with replacement [99]. This was shown to improve the learning of the deep neural network in high-dimensional parameter spaces in the relatively short amount of time devoted to human interaction. We set the hyperparameters of the deep neural network by performing a parameter sweep and leading sanity checks with the algorithm; we report them in Appendix B.

5.2.3 Exploration Behaviour. We developed a novel exploration method for autonomous exploration behaviour (see Fig. 7). It builds on an intrinsic motivation method, which pushes the agent to "explore what surprises it" [11]. Specifically, it has the agent direct its exploratory actions toward uncharted parts of the space, rather than simply making random moves—as in the ε-greedy approach implemented in our initial prototype. Our method is based on state visitation density to push the agent to take exploratory actions toward unknown state zones. It builds a density model of the parameter space based on an estimation of state-visitation counts, called the pseudo-count $\hat{V}(S)$, and a total visit pseudo-count $\hat{v}$ using a density model $\hat{p}_\phi(S)$. It then adds a reward bonus, R^+, to the agent, based on the novelty of the state. We parameterized ε with an exponential decay in such a way that its initial value would slowly decrease along user exploration. For our application in sound design, agent speed in autonomous exploration mode was set to one action by tenths of a second. We report the hyperparameters set for our exploration method after sanity checks in Appendix B, and detail the density model in Appendix C.

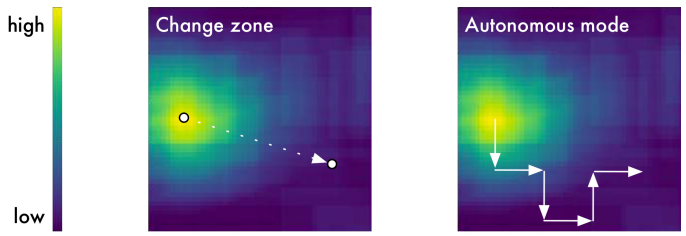

Fig. 7. Schematic representations for exploration behaviour. The color scale depicts the density model for a two-dimensional state space. Left: Changing zone has the agent jump to a state with lowest density. Right: Autonomous exploration has the agent take successive actions toward states with lowest density.

We used tile coding, a specific feature representation extensively used in the reinforcement learning literature to efficiently compute and update the density model $\hat{p}_\phi(S)$ in high-dimensional spaces [97, 100]. To our knowledge, it has not been used for density estimation or in relation with the pseudo-count technique as used here. Tile coding as density estimation techniques was preferred over other techniques such as Gaussian Mixture Models or using Artificial Neural Networks for its low computational cost and ability to scale to higher dimensions. Other exploration methods are based on Thompson sampling [93], bootstrapping neural networks for deep exploration [74]

or by adding parametric noise to network weights [38]. Approaches such as Thompson sampling have been used to find an appropriate exploration-exploitation balance but require a prior distribution on the parameter space. Bayesian methods can even be used to compute an optimal exploration-exploitation balance but often require much too great computation resources for the high-dimensional state-action spaces considered in reinforcement learning.

5.3 Integrating Interaction Modalities In Reinforcement Learning

To fully realize our interaction design, we integrated the modalities defined in Section 5.1 within the reinforcement learning framework defined in Section 5.2.

5.3.1 User Feedback. We developed generic methods corresponding to user feedback modalities defined in Section 5.1.1 that we used in the feedback formalization of Section 5.2.1. For guiding feedback, we assigned user positive or negative feedback value over the p last state-action pairs taken by the agent (see Fig. 8, left), with a decreasing credit given by a Gamma distribution [57]. For zone feedback, we computed all possible state-action pairs leading to the state being labelled and impacted them with positive or negative feedback received (see Fig. 8, right). This enables to build attractive and repulsive zones for the agent in the parameter space. This reward bonus is computed using the density model described in Section 5.2.3.

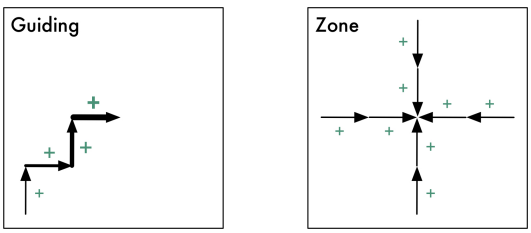

Fig. 8. Schematic representations for feedback computation methods. Here, positive feedback is given in some state situated at the center of the square. Left: Guiding feedback is distributed over the p lastly-visited state-action pairs. Right: Zone feedback impacts all state-action pairs potentially leading to the labelled state.

5.3.2 State Commands. We developed generic methods corresponding to state commands defined in Section 5.1.2 using the exploration behaviour defined in Section 5.2.3. Changing zone has the agent randomly sampling the density distribution and jump to the state with lowest density (see Fig. 7, left). Autonomous exploration mode has the agent take exploratory actions that lead to the nearest state with lowest density with probability ε (see Fig. 7, right).

5.3.3 Direct Manipulation. We integrated direct manipulations as defined in Section 5.1.3 by leveraging the learning algorithm defined in Section 5.2.2. When parameters are modified by the user, the reinforcement learning agent converts all parameters' numerical values as a state representation, taking advantage of the algorithm's robustness in changes of discretization. As such, direct manipulation is almost continuous in the Co-Explorer, which strongly differs from the coarse-grained, three-level implementation of our initial RL agent prototype. Finally, reseting agent memory has the reinforcement learning algorithm erase all stored user feedback and trajectory, and load a new model.

5.4 Implementation

5.4.1 Agent. We implemented the Co-Explorer as a Python library[4]. It allows to interface the deep reinforcement learning agent to any external input device and output software, using the OSC protocol for message communication [103]. This was done to enable future applications outside the sound design domain. Each of the features described in Section 5.2 are implemented as parameterized functions, which supports experimentation of interactive reinforcement learning with various parameter values as well as order of function calls. The current version relies on TensorFlow [1] for deep neural network computations. The complete algorithm implementation is described in Appendix D.

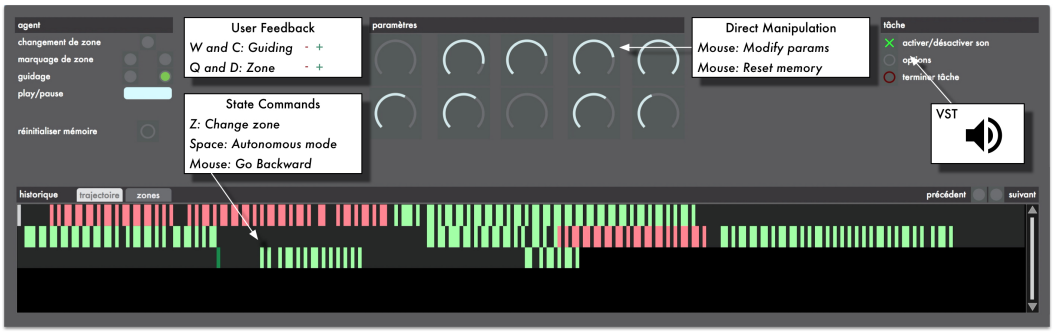

Fig. 9. Co-Explorer interface.

5.4.2 Interface. We implemented an interactive interface for our application in sound design (Fig. 9), which integrates all interaction modalities defined in Section 5.1. It builds on Max/MSP, a visual programming environment for real-time sound synthesis and processing. Standard parametric knobs enable users to directly manipulate parameters, as well as to see the agent act on it in real-time. An interactive history allows users to command the agent to go to a previously-visited state, be they affected by user feedback (red for negative, green for positive) or simply passed through (grey). Keyboard inputs support user feedback communication, as well as state commands that control agent exploration (changing zone, and start/stop autonomous exploration mode). Lastly, a clickable button enables users to reset agent memory.

6 EVALUATION WORKSHOP

We evaluated the Co-Explorer in a workshop with a total of 12 professional users (5 female, 7 male). The aims of the workshop were to: Evaluate each interaction modality at stake in the Co-Explorer; understand how users may appropriate the agent to support parameter space exploration.

The workshop was divided in two tasks: (1) explore to discover, and (2) explore to create. This structure was intended to test the Co-Explorer in two different creative tasks (described in Section 6.1 and 6.2, respectively). Participants ranged from sound designers, composers, musicians, and artists to music researchers and teachers. They were introduced to the agent's interactive modalities and its internal functioning at the beginning of the workshop. In each part, they were asked to report their observations by filling a browser-based individual journal. Group discussion was carried on at the end of the workshop to let participants exchange views over parameter space exploration. The workshop lasted approximately three hours.

[4]https://github.com/Ircam-RnD/coexplorer

6.1 Part 1: Explore to Discover

6.1.1 Procedure. In the first part of the workshop, participants were presented with one parameter space (see Fig. 10). They were asked to use the Co-Explorer to explore and discover the sound space at stake. Specifically, we asked them to find and select five presets to constitute a representative sample of the space. We defined the parameter space by selecting ten parameters from a commercial VST. Participants were encouraged to explore the space thoroughly. The task took place after a 10-minute familiarizing session: individual exploration lasted 25 minutes, followed by 5 minutes of sample selection, and 20 minutes of group discussion. Each session started with a fully-untrained agent.

Fig. 10. Our participants testing the Co-Explorer in the evaluation workshop.

6.1.2 Analysis. All participant's actions were logged into a file. These contained timed onsets for user feedback—*i.e.,* binary guiding and zone feedback—, state commands—*i.e.,* backward commands in the history, changing zone commands, and autonomous exploration starting/stopping—, and direct manipulations—*i.e.,* parameter temporal evolutions. We also logged timed onsets for preset selection in relation to the task, but did not include the five presets themselves into our analysis. Our motivation was to focus on the process of exploration in cooperation with the Co-Explorer, rather than on the output of it. We used structured observation to extract information from individual journals and group discussion.

6.1.3 Results. We first looked at how users employed state commands. Specifically, the autonomous exploration mode, which consisted in letting the agent act cotinuously on parameters on its own, was an important new feature compared to our sequentiam initial RL agent prototype. Participants spent more than half of the task using the Co-Explorer in this mode (total of 13 minutes on average, $\sigma = 4.7$). Ten participants used autonomous exploration over several short time slices (average of 50 seconds, $\sigma = 25$s), while the two remaining participants used it over one single long period (respectively 9 and 21 minutes). P5 commented about the experience: *"It created beautiful moments during which I really felt that I could anticipate what it was doing. That was when I really understood the collaborative side of artificial intelligence".*

The changing zone command, which enabled to jump to an unexplored zone in the parameter space, was judged efficient by all participants to find diverse sounds within the design space. It was used between 14 and 90 times, either to start a new exploration (P1: *"Every time I used it, I found*

myself in a zone that was sufficiently diametrically opposed to feel that I could explore something relatively new"), or to rapidly seize the design space in the context of the task (P12: *"I felt it was easy to manage to touch the edges of all opposite textures"*). Interestingly, P2 noticed that the intrinsic motivation method used for agent exploration behaviour *"brought something more than a simple random function that is often very frustrating"*.

We then looked at how users employed feedback. Guiding feedback, enabling guidance toward agent actions, was effectively used in conjunction with autonomous exploration by all participants, balancing positive with negative (55% positive on average, σ = 17%). Participants gave various amounts of guiding feedback (between 54 and 1489 times). These strategies were reflected by different reactions toward the Co-Explorer. For example, one participant was uncertain in controlling the agent through feedback: *"if the agent goes in the right direction, I feel like I should take time to see where it goes"*, he commented. On the contrary, P1 was radical in his controlling the agent, stating that he is *"just looking for another direction"*, and that he uses feedback *"without any value judgement"*. This reflects the results described in Section 4.2.4 using our initial RL agent prototype.

Zone feedback, enabling customization of the space with binary state labels, was mostly given as positive by participants (72%, σ = 18%). Two participants found the concept of negative zones to be counter-intuitive. *"I was a bit afraid that if I label a zone as negative, I could not explore a certain part of the space"*, P8 coined. This goes in line with previous results on applying interactive reinforcement learning in the field of robotics [98]. All participants agreed on the practicality of combining positive zone feedback with backward state commands in the history to complete the task. *"I labeled a whole bunch of presets that I found interesting [...] to after go back in the trajectory to compare how different the sounds were, and after continue going in other zones. I found it very practical"*, P8 reported. Overall, zone feedback was less times used than guiding feedback (between 10 and 233 times).

Finally, direct manipulation was deemed efficient by participants in certain zones of the design space. *"When I manage to hear that there is too much of something, it is quicker to parametrize sound by hand than to wait for the agent to find it itself, or to learn to detect it"*, P4 analyzed. P10 used them after giving a backward state command, saying she *"found it great in cases where one is frustrated not to manage to guide the agent"*. P11 added that she directly manipulate parameters to *"adjust the little sounds that [she] selected"*. P1 suggested that watching parameters move as the agent manipulates them could help learn the interface: *"From a pedagogical point of view, [the agent] allows to access to the parameters' functioning and to the interaction between these parameters more easily [than without]"*. This supports the fact that machine learning visualizations may be primordial in human-centred applications to enable interpretability of models [3].

6.1.4 Relevance to Task. Three participants wished that the Co-Explorer reacted more quickly to feedback in relation to the task: *"I would really like to feel the contribution of the agent, but I couldn't"*, P12 said. Also, P3 highlighted the difficulties to give evaluative feedback in the considered task: *"without a context, I find it hard"*, he analysed. Despite this, all participants wished to spend more time teaching the Co-Explorer, by carefully customizing the parameter space with user feedback. For example, five participants wanted to slow the speed of the agent during autonomous exploration to be able to give more precise guidance feedback. Also, three participants wanted to express sound-related feedback: *"There, I am going to guide you about the color of the spectrum. [...] There, I'm going to guide you about, I don't know, the harmonic richness of the sound, that kind of stuff..."*, P4 imagined.

6.2 Part 2: Explore to Create

6.2.1 Procedure. In the second part of the workshop, participants were presented with four pictures (Fig. 11) created by renowned artists and photographers. For each of these four pictures, they were asked to explore and create two sounds that subjectively depict the atmosphere of the picture. In this part, we encouraged participants to appropriate interaction with the Co-Explorer and feel free to work as they see fit. We used a new sound design space for this second part, which we designed by selecting another ten parameters from the same commercial VST than in Part 1. Individual exploration and sound selection lasted 30 minutes, followed by 20 minutes of group discussion and 10 minutes of closing discussion. The session started with a fully-untrained agent.

Fig. 11. The four pictures framing the creation task of the workshop.

6.2.2 Analysis. All participant actions were logged into a file, along with timed parameter presets selected for the four pictures. Again, we focused our analysis on the process of exploration rather than on the output of it. Specifically, for this open-ended, creative task, we did not aim at analysing how each agent interaction modality individually relates to a specific user intention. Rather, we were interested in observing how users may appropriate the mixed-initiative workflow at stake in the Co-Explorer.

We used Principal Component Analysis (PCA [51]), a dimensionality reduction method, to visualize how users switched parameter manipulation with agents. We first concatenated all participants' parameter evolution data as an n-dimensional vector to compute the two first principal components. We then projected each participant data onto these two components to support analysis of each user trajectory on a common basis. By doing this, relatively distant points would correspond to abrupt changes made in parameters (*i.e.*, to moments when the user takes the lead on exploration). Continuous lines would correspond to step-by-step changes in parameters (*i.e.*, to moments when the Co-Explorer explores autonomously). PCA had a stronger effect in the second part of our workshop. We interpret this as a support to the two-part structure that we designed for the workshop, and thus did not include analysis of the first part. Finally, we used structured observation to extract information from individual journals and group discussion.

6.2.3 Exploration Strategies. All participants globally expressed more ease interacting with the Co-Explorer in this second task. *"I felt that the agent was more adapted to such a creative, subjective... also more abstract task, where you have to illustrate. It's less quantitative than the first task"*, P9 analysed. User feedback was also reported to be more intuitive when related to a creative goal: *"all parameters took their sense in a creative context. [...] I quickly found a way to work with it that was very efficient and enjoyable"*, P5 commented. Figure 12 illustrates the PCA for two different users.

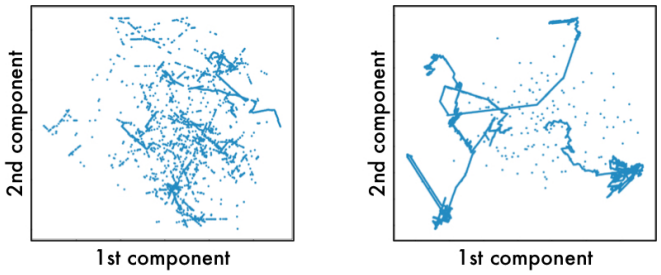

Fig. 12. Two types of co-exploration partnerships shown in PCA visualizations of parameter evolution: User-as-leader (P9, left) and agent-as-leader (P7, right). Relatively distant points correspond to abrupt changes made in parameters (*i.e.*, to moments when the user takes the lead). Continuous lines correspond to step-by-step changes in parameters (*i.e.*, to moments when the Co-Explorer takes the lead).

Qualitative analysis of PCAs let us conceptualize a continuum of partnerships between our participants and the Co-Explorer. These could be placed anywhere between the two next endpoints:

- **User-as-leader**: This workflow involves users first building a customized map of the design space, then generating variations over these custom presets. In terms of interaction modalities, this consists in first iteratively using changing zone and positive zone feedback to store custom presets, then either using direct manipulation or short autonomous explorations to generate variations of these presets.
- **Agent-as-leader**: This workflow involves users letting the Co-Explorer lead most of parameter manipulation. In terms of interaction modalities, this consists in using autonomous exploration mode combined with guiding feedback over long periods of time, occasionally using changing zone or direct manipulation to choose a start point for the Co-Explorer to lead autonomous exploration.

Our interpretation is as follows. User-as-leader partnership may correspond to user profiles that approach creative work as a goal-oriented task, where efficacy and control are crucial (P10: *"I am accustomed... Where I work, if you prefer, we have to get as quick as possible to the thing that works the best, say, and I cannot spend so much time listening to the agent wandering around"*). Reciprocally, agent-as-leader partnership may correspond to user profiles that approach creative work as an open-ended task, where serendipity is essential for inspiration (P5: *"I did not try to look for the sound that would work the best. I rather let myself be pushed around, even a bit more than in my own practice"*). Some participants did not stabilize into one single partnership, but rather enjoyed the flexibility of the agent. *"It was quite fun to be able to let the agent explore, then stop, modulate a bit some parameters by hand, let it go and guide it again, changing zones too, then going back in the history... Globally, I had the impression of shaping, somewhat... I found it interesting"*, P11 coined.

Agent memory was handled with relevance to various creative processes toward the pictures. Seven participants disposed all four pictures in front of them (P7: *"to always have them in mind. Then, depending on the agent's exploration, I told myself 'hey, this sound might correspond to this picture'"*). Three participants focused on one picture at a time, *"without looking at the others"*. Four participants never reset the memory (P11: *"my question was, rather, in this given sonic landscape, how can I handle these four pictures, and reciprocally"*), and three participants reset agent memory for each of the different atmospheres shared by the pictures. Overall, participants benefited from partnering with the Co-Explorer in parameter space exploration: *"It's a mix of both. I easily managed to project a sound on the picture at first glance, then depending on what was proposed, it gave birth to many ideas"*, one participant said.

6.2.4 Toward Real-World Usages. All participants were able to describe additional features for the Co-Explorer to be usable in their real-world professional work environments—examples are, among others, connection to other sound spaces, memory transfer from one space to another, multiple agent memory management, or data exportation. They also anticipated creative uses for which the Co-Explorer were not initially designed. Half of the participants were enthusiastic about exploiting the temporal trajectories as actual artifacts of their creation (P6: *"What I would find super interesting is to be able to select the sequences corresponding to certain parameter evolution, or playing modes. [...] It would be super great to select and memorize this evolution, rather than just a small sonic fragment"*). Finally, two participants further imagined the Co-Explorer to be used as musical colleagues—either as improvisers with which one could *"play with both hands"* (P2), or as *"piece generators"* (P6) themselves.

7 DISCUSSION

Our process of research, design, and development led to contributions at three different levels: (1) conceptual insight on human exploration; (2) technical insight on reinforcement learning; and (3) joint conceptual and technical design guidelines on machine learning for human creativity.

7.1 Conceptual Insight

7.1.1 From Exploration to Co-Exploration. Our work with interactive reinforcement learning allowed for observing and characterizing user approaches to parameter space exploration, and supported it. While manipulating unlabelled parametric knobs of sound synthesizers, participants alternated between an *analytical* approach—attempting to understand the individual role of each parameter—and a *spontaneous* approach that could lead to combinations in the parameter space that might not be guessed with the analytical approach. While interacting with a reinforcement learning agent, participants tended to alternate the lead in new types of mixed-initiative workflows [47] that we propose to call *co-exploration* workflows. *User-as-leader* workflow was used for gaining control over each parameter of the design space. *Agent-as-leader* workflow allowed to relax users' control and provoke discoveries through the specific paths autonomously taken by the agent in the parameter space. Importantly, the benefit of interactive reinforcement learning for co-exploring sound spaces was dependent on the task. We found that this co-exploration workflow were more relevant to human exploration tasks that have a focus on creativity, such as in our workshop's second task, rather than discovery. Therefore, we believe that this workflow is well-suited in cases where exploration is somehow holistic (as in the creative task) rather than analytic (as in the discovery task where the goal is to understand the sound space to find new sounds). In a complementary work, described in Appendix E, we were able to validate this hypothesis, by proving that guiding the RL-agent better supports user creativity than simply using a standard parametric interface.

7.1.2 Methodology. Our user-centered design approach to interactive reinforcement learning and exploration allowed us to rapidly evaluate flexible interaction designs without focusing on usability. This process let us discover innovative machine learning uses that we may not have anticipated if we had started our study with an engineering phase. The simple, flexible, and adaptable designs tested in our first pilot study (parametric vs. RL) could in this sense be thought as technology probes [50]. Working with professional users of different background and practices—from creative coders to artists less versed in technology—was crucial to include diverse user feedback in the design process. Our results support this, as many user styles were supported by the Co-Explorer. That said, user-driven design arguably conveys inherent biases of users. This is particularly true when promoting AI in interactive technology [7, 15]. As a matter of fact, alongside a general

enthusiasm, we did observe a certain ease among our professional users for expressing tough critiques, at times being skeptical on using AI, especially when the perception of the algorithm choice would contradict their spontaneous choice. Yet, the two professional users that took part to both our pilot study and workshop found the use of AI as welcome, testifying of its improvement along the development process.

7.1.3 Evaluation. Lastly, evaluation of reinforcement learning tools for creativity remains to be investigated more deeply. While our qualitative approach allowed us to harvest thoughtful user feedback on our prototypes' interaction modalities, it is still hard to account for direct links between agent computations and user creative goals. Using questionnaire methods, such as the Creativity Support Index [17], may enable to measure different dimensions of human creativity in relation to different algorithm implementations. As a first step toward this direction, we report a preliminary summary that maps some of participants' quotes in our evaluation workshop to Creativity Support Index dimensions in Appendix F. Also, focusing on a specific user category could also allow more precise evaluation in relationship to a situated set of creative practices and uses. Alternatively, one could aim at developing new reinforcement learning criteria that extends standard quantitative measures—such as convergence or learning time [97]—to the qualitative case of human exploration. Research on interactive supervised learning has shown that criteria usually employed in the field of Machine Learning may not be adapted to users leading creative work [34]. We believe that both HCI and ML approaches may be required and combined to produce sound scientific knowledge on creativity support evaluation.

7.2 Technical Insight

7.2.1 Computational Framework. Our two working prototypes confirmed that interactive reinforcement learning may stand as a generic technical framework for parameter space exploration. The computational framework that we proposed in Section 4.2.1, leveraging states, actions, and rewards, strongly characterized the mixed-initiative co-exploration workflows observed in Section 6.2—*e.g.,* making small steps and continuous trajectories in the parameter space. Other interactive behaviours could have been implemented—*e.g.,* allowing the agent to act on many parameters in only one action, or using different a_i values for different action sizes—to allow for more diverse mixed-initiative behaviours. Alternatively, we envision that domain-specific representations may be a promising approach for extending co-exploration. In the case of sound design, one could engineer high-level state features based on audio descriptors [85] instead of using raw parameters. This could allow RL agents to learn state-action representations that would be independent from the parameter space explored—potentially allowing memory transfer from one parameter state space to another. This could also enable agent adaptation of action speed and precision based on perceptual features of the parameter space—potentially avoiding abrupt jumps in sound spaces.

7.2.2 Learning Algorithm. Reinforcement learning algorithmic functioning, enabling agents to learn actions over states, was of interest for our users, who were enthusiastic in teaching an artificial agent actions by feedback. Our deep reinforcement learning agent is a novel contribution to HCI research compared to multi-armed bandits (which explore actions over one unique state [68]), contextual bandits (which explore in lower-dimensional state spaces [59]), and bayesian optimization (which explores states at implicit scales [89]). We purposely implemented heterogeneous ways of teaching with feedback based on our observations of users' approaches to parameter space exploration, which extends previous implementations such as those in the Drawing Apprentice [22]. We also decided to have the agent maximize direct user feedback (for which *Deep TAMER* was adapted [99], as opposed to *Sarsa* [97]), rather than to optimize one general RL policy. Indeed, our observations in the pilot study suggested that exploring users may not generate one goal-oriented

feedback signal, but may rather have several sub-optimal goals. They may also make feedback mistakes, act socially toward agents, or even try to trigger surprising agent behaviours over time. In this paper, we focused on qualitative evaluation of the learned policies to provide a proof of interest of interactive reinforcement learning from an HCI perspective. At time of writing, we know of no way to provide quantitative evaluation of the interactive learning of such policies with an exploring user from a machine learning perspective. We believe that such evaluation methods do constitute a matter of research beyond the scope of this paper, which is currently an emerging topic in RL [46]. Beyond agent customisation, future research may address agent generalisation to other sessions or users, for example having users start a session with a partially-trained agent instead of fully-untrained.

7.2.3 Exploration Behaviours. The exploration behaviours of reinforcement learning agents were shown promising for fostering creativity in our users. Both ε-greedy and intrinsic motivation method were adapted to the interactive case of a user leading exploration. One of our users felt that intrinsic motivation had agents behave better than random. In a complementary work [86], we confirmed that users perceived the difference between a random agent and an interactively-learning RL agent. Interestingly, what they perceive may be more related to the agent's global effect in exploring the parameter space, rather than the difference between various implementations of agent exploration. Future work may investigate how user perception of agent exploration may relate to specific implementations of exploration methods. Complementary to such an approach, future work may study co-exploration partnerships in real-world applications to inquire co-adaptation between users and agents over longer periods of time [70]. On the one hand, users could be expected to learn to provide better feedback to RL agents to fulfill their creative goals—as it was shown in interactive approaches to supervised learning [34]. On the other hand, agents could be expected to act more in line with users by exploiting larger amounts of accumulated feedback data—as it is typical with interactive reinforcement learning agents [97]. A more pragmatic option would be to give users full control over agent epsilon values—*e.g.,* using an interactive slider [59]—to improve partnership in this sense.

7.3 Guidelines for Designing With Machine Learning in Creative Applications

Based on our work with reinforcement learning, we identified a set of design challenges for leading joint conceptual and technical development of other machine learning frameworks for creative HCI applications. We purposely put back quotes from our participants in this section to inspire readers with insights on AI from users outside our design team.

7.3.1 Engage Users with Machine Learning. The Co-Explorer enabled users to fully engage with reinforcement learning computational framework. Users could explore as many states, provide as much feedback, and generate as many agent actions as they wanted to. They also had access to agent memory, be it by navigating in the interactive history, or by reseting the learned behaviour. In this sense, they had full control over the algorithmic learning process of the agent. This is well articulated by a participant, whose quote can be reported here: *"I did not feel as being an adversary to, or manipulated, by the system. A situation that can happen with certain audio software that currently use machine learning, where it is clear that one tries to put you on a given path, which I find frustrating—but this was not the case here".*

These observations suggest that user engagement at different levels of machine learning processes may be essential to create partnering flows [75]. That is, users should be provided with interactive controls and simple information on learning to actively direct co-creation. This goes in line with previous works studying user interaction with supervised learning in creative tasks [3], which showed how users can build better partnerships by spending time engaging with algorithms [34].

Careful interaction design must be considered to balance full automation with full user control and aim at creating flow states among people [21]. Aiming at such user engagement may also constitute a design opportunity to demystify AI systems, notably by having users learn from experience how algorithms work with data [32].

7.3.2 Foster Diverse Creative Processes. Our work showed that the Co-Explorer supported a wide diversity of creative user processes. Users could get involved in open-ended, agent-led exploration, or decide to focus on precise, user-led parameter modification. Importantly, none of these partnerships were clearly conceptualized at the beginning of our development process. Our main focus was to build a reinforcement learning agent able to learn from user feedback and to be easily controllable by users. In this sense, the Co-Explorer was jointly designed and engineered to ensure a dynamic human process rather than a static media outcome. As a matter of fact, we report one participant's own reflection, which we believe illustrate our point: *"What am I actually sampling [from the parameter space]? Is is some kind of climate that is going to direct my creation afterwards? [...] Or am I already creating?"*.

This suggests that supporting the process of user appropriation may be crucial for building creative AI partnerships. Many creative tools based on machine learning often focus on engineering one model to ensure high performance for a given task. While these tools may be useful for creative tasks that have a focus on high productivity, it is arguable whether they may be suited to creative work that has a focus on exploration as a way to build expression. For the latter case, creative AI development should not focus on one given user task, but should rather focus on providing users with a dynamic space for expression allowing many styles of creation [82]. The massive training datasets, which are usually employed in the Machine Learning community to build computational creativity tools, may also convey representational and historical biases among end users [96]. Interactive approaches to machine learning directly address this issue by allowing users to intervene in real-time in the learning process [33].

7.3.3 Steer Users Outside Comfort Zones. The Co-Explorer actively exposed the exploration behaviour of reinforcement learning to users. This goes in opposition with standard uses of these algorithms [14], and may provoke moments where agents behaviours may not align with users creative drive [20]. Yet, it managed to build *"playful"* and *"funny"* partnerships that led some users to reconsider their approach to creativity, as one participant confessed: *"At times, the agent forced me to try and hear sounds that I liked less—but at least, this allowed me to visit unusual spaces and imagine new possibilities. This, as a process that I barely perform in my own creative practice, eventually appeared as appealing to me"*.

This suggests that AI may be used beyond customisation aspects to steer users outside their comfort zones in a positive way. That is, designers should exploit non-optimal algorithmic behaviours in machine learning methods to surprise, obstruct, or even challenge users inside their creative process. Data-driven user adaptation may be taken from an opposite side to inspire users from radical opposition and avoid hyper-personalization [8]. Such an anti-solutionist [12] approach to machine learning may encourage innovative developments that fundamentally reconsider the underlying notion of universal performance commonly at stake in the field of Machine Learning and arguably not adapted to the human users studied in the field of Human-Computer Interaction. It may also allow the building of imperfect AI colleagues, in opposion to "heroic" AI colleagues [27]: being impressed by the creative qualities of an abstract artificial entity may not be the best alternative to help people develop as creative thinkers [81]. The Co-Explorer fairly leans toward such an unconventional design approach, which, in default of fitting every user, surely forms one of its distinctive characteristics.

Several machine learning frameworks remains to be investigated under the light of these human-centred challenges. Evolutionary computation methods [37] may be fertile ground for supporting user exploration and automated refinement of example designs. Active learning methods [88] may enable communication flows between agents and users that go beyond positive or negative feedback. Dimensionality reduction methods for interactive visualization [69] may improve intelligibility of agent actions in large parameter spaces and allow for more trustable partnerships. Ultimately, combining reinforcement learning with supervised learning could offer users with the best of both worlds by supporting both example and feedback inputs. Inverse reinforcement learning [2] may stand as a technical framework supporting example input projection and transformation into reward functions in a parameter space.

8 CONCLUSION

In this paper we presented the design of a deep reinforcement learning agent for human parameter space exploration. We worked in close relationship with professional creatives in the field of sound design and led two design iterations during our research process. A first pilot study let us observe users interacting with standard parametric interfaces, as well as with an initial interactive reinforcement learning prototype. The gathered user feedback informed the design of the Co-Explorer, our fully-functioning prototype, for which we led joint design and engineering for the specific task of parameter space exploration. A final workshop allowed us to observe a wide range of partnerships between users and agents, in tasks requiring both quantitative, media-related sampling and qualitative, creative insight.

Our results raised contributions at different levels of research, development, and design. We defined properties of user approaches to parameter space exploration within standard parametric interfaces, as well as to what we called parameter space co-exploration—exploring in cooperation with a reinforcement learning agent. We adapted a deep reinforcement learning algorithm to the specific case of parameter space exploration, developing specific computational methods for user feedback input in high-dimensional spaces, as well as a new algorithm for agent exploration based on intrinsic motivation. We raised general design challenges for guiding the building of new human-AI partnerships, encouraging interdisciplinary research collaborations [77] that value human creativity over machine learning performance. We look forward to collaborating with researchers, developers, designers, artists, and users from other domains to take up the societal challenge of designing partnering AI tools that nurture human creativity.

ACKNOWLEDGMENTS

We are grateful to our participants for their precious time and feedback. We thank Benjamin Matuszewski, Jean-Philippe Lambert, and Adèle Pécout for their support in designing the studies. This research was partly supported by the ELEMENT project (ANR-18-CE33-0002) from the French National Research Agency.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

[2] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 1.

[3] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.

[4] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on*

Human Factors in Computing Systems. ACM, 337–346.
[5] Saleema Amershi, James Fogarty, and Daniel Weld. 2012. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 21–30.
[6] Saleema Amershi, Bongshin Lee, Ashish Kapoor, Ratul Mahajan, and Blaine Christian. 2011. CueT: human-guided fast and accurate network alarm triage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 157–166.
[7] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for Human-AI Interaction. (2019).
[8] Kristina Andersen and Peter Knees. 2016. Conversations with Expert Users in Music Retrieval and Research Challenges for Creative MIR.. In *ISMIR*. 122–128.
[9] Kumaripaba Athukorala, Alan Medlar, Antti Oulasvirta, Giulio Jacucci, and Dorota Glowacka. 2016. Beyond relevance: Adapting exploration/exploitation in information retrieval. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 359–369.
[10] Kumaripaba Athukorala, Alan Medlar, Antti Oulasvirta, Giulio Jacucci, and Dorota Glowacka. 2016. Beyond Relevance: Adapting Exploration/Exploitation in Information Retrieval. In *Proceedings of the 21st International Conference on Intelligent User Interfaces (IUI '16)*. ACM, New York, NY, USA, 359–369. https://doi.org/10.1145/2856767.2856786
[11] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. 2016. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*. 1471–1479.
[12] Mark Blythe, Kristina Andersen, Rachel Clarke, and Peter Wright. 2016. Anti-Solutionist Strategies: Seriously Silly Design Fiction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 4968–4978.
[13] Eric Brochu, Tyson Brochu, and Nando de Freitas. 2010. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 103–112.
[14] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
[15] Baptiste Caramiaux, Fabien Lotte, Joost Geurts, Giuseppe Amato, Malte Behrmann, Frédéric Bimbot, Fabrizio Falchi, Ander Garcia, Jaume Gibert, Guillaume Gravier, et al. 2019. AI in the media and creative industries. (2019).
[16] Mark Cartwright, Bryan Pardo, and Josh Reiss. 2014. Mixploration: Rethinking the audio mixer interface. In *Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM, 365–370.
[17] Erin Cherry and Celine Latulipe. 2014. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 4 (2014), 21.
[18] John M Chowning. 1973. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the audio engineering society* 21, 7 (1973), 526–534.
[19] Paul Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741* (2017).
[20] Jacob W Crandall, Mayada Oudah, Fatimah Ishowo-Oloko, Sherief Abdallah, Jean-François Bonnefon, Manuel Cebrian, Azim Shariff, Michael A Goodrich, Iyad Rahwan, et al. 2018. Cooperating with machines. *Nature communications* 9, 1 (2018), 233.
[21] Mihaly Csikszentmihalyi. 1997. Flow and the psychology of discovery and invention. *HarperPerennial, New York* 39 (1997).
[22] Nicholas Davis, Chih-PIn Hsiao, Kunwar Yashraj Singh, Lisa Li, and Brian Magerko. 2016. Empirically studying participatory sense-making in abstract drawing with a co-creative cognitive agent. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 196–207.
[23] Nicholas M Davis, Yanna Popova, Ivan Sysoev, Chih-Pin Hsiao, Dingtian Zhang, and Brian Magerko. [n. d.]. Building Artistic Computer Colleagues with an Enactive Model of Creativity.
[24] Stefano Delle Monache, Davide Rocchesso, Frédéric Bevilacqua, Guillaume Lemaitre, Stefano Baldan, and Andrea Cera. 2018. Embodied Sound Design. *International Journal of Human-Computer Studies* (2018).
[25] Ruta Desai, Fraser Anderson, Justin Matejka, Stelian Coros, James McCann, George Fitzmaurice, and Tovi Grossman. 2019. Geppetto: Enabling Semantic Design of Expressive Robot Behaviors. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 369.
[26] Christoph Sebastian Deterding, Jonathan David Hook, Rebecca Fiebrink, Jeremy Gow, Memo Akten, Gillian Smith, Antonios Liapis, and Kate Compton. 2017. Mixed-Initiative Creative Interfaces. In *CHI EA'17: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM.
[27] Mark d'Inverno, Jon McCormack, et al. 2015. Heroic versus Collaborative AI for the Arts. (2015).
[28] Alan Dix. 2007. Designing for appropriation. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it-Volume 2*. British Computer Society, 27–30.

[29] Kees Dorst and Nigel Cross. 2001. Creativity in the design process: co-evolution of problem–solution. *Design studies* 22, 5 (2001), 425–437.
[30] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX Design Innovation: Challenges for Working with Machine Learning as a Design Material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 278–288.
[31] Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*. ACM, 39–45.
[32] Rebecca Fiebrink. 2019. Machine Learning Education for Artists, Musicians, and Other Creative Practitioners. *ACM Transactions on Computing Education* (2019).
[33] Rebecca Fiebrink and Baptiste Caramiaux. 2016. The machine learning algorithm as creative musical tool. *Handbook of Algorithmic Music* (2016).
[34] Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. 2011. Human Model Evaluation in Interactive Supervised Learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 147–156. https://doi.org/10.1145/1978942.1978965
[35] Rebecca Fiebrink, Daniel Trueman, N Cameron Britt, Michelle Nagai, Konrad Kaczmarek, Michael Early, MR Daniel, Anne Hege, and Perry R Cook. 2010. Toward Understanding Human-Computer Interaction In Composing The Instrument.. In *ICMC*.
[36] Tesca Fitzgerald, Ashok Goel, and Andrea Thomaz. [n. d.]. Human-Robot Co-Creativity: Task Transfer on a Spectrum of Similarity.
[37] David B Fogel. 2006. *Evolutionary computation: toward a new philosophy of machine intelligence*. Vol. 1. John Wiley & Sons.
[38] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, et al. 2017. Noisy networks for exploration. *arXiv preprint arXiv:1706.10295* (2017).
[39] Jules Francoise and Frederic Bevilacqua. 2018. Motion-Sound Mapping through Interaction: An Approach to User-Centered Design of Auditory Feedback Using Machine Learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 8, 2 (2018), 16.
[40] Rémy Frenoy, Yann Soullard, Indira Thouvenin, and Olivier Gapenne. 2016. Adaptive training environment without prior knowledge: Modeling feedback selection as a multi-armed bandit problem. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, 131–139.
[41] Jérémie Garcia, Theophanis Tsandilas, Carlos Agon, and Wendy Mackay. 2012. Interactive paper substrates to support musical creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1825–1828.
[42] Marco Gillies. 2019. Understanding the Role of Interactive Machine Learning in Movement Interaction Design. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26, 1 (2019), 5.
[43] Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frederic Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, et al. 2016. Human-centred machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 3558–3565.
[44] Dorota Glowacka, Tuukka Ruotsalo, Ksenia Konuyshkova, kumaripaba Athukorala, Samuel Kaski, and Giulio Jacucci. 2013. Directing Exploratory Search: Reinforcement Learning from User Interactions with Keywords. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces (IUI '13)*. ACM, New York, NY, USA, 117–128. https://doi.org/10.1145/2449396.2449413
[45] Yuval Hart, Avraham E Mayo, Ruth Mayo, Liron Rozenkrantz, Avichai Tendler, Uri Alon, and Lior Noy. 2017. Creative foraging: An experimental paradigm for studying exploration and discovery. *PloS one* 12, 8 (2017), e0182133.
[46] Xu He, Haipeng Chen, and Bo An. 2020. Learning Behaviors with Uncertain Human Feedback. *arXiv preprint arXiv:2006.04201* (2020).
[47] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 159–166.
[48] Andy Hunt and Ross Kirk. 2000. Mapping strategies for musical performance. *Trends in gestural control of music* 21, 2000 (2000), 231–258.
[49] Andy Hunt and Marcelo M Wanderley. 2002. Mapping performer parameters to synthesis engines. *Organised sound* 7, 2 (2002), 97–108.
[50] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, et al. 2003. Technology probes: inspiring design for and with families. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 17–24.
[51] Ian Jolliffe. 2011. Principal component analysis. In *International encyclopedia of statistical science*. Springer, 1094–1096.
[52] Sergi Jorda. 2005. *Digital Lutherie Crafting musical computers for new musics' performance and improvisation*. Ph.D. Dissertation. Universitat Pompeu Fabra.

[53] Anna Kantosalo, Jukka M Toivanen, Ping Xiao, and Hannu Toivonen. 2014. From Isolation to Involvement: Adapting Machine Creativity Software to Support Human-Computer Co-Creation.. In *ICCC*. 1–7.

[54] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1343–1352.

[55] Simon Katan, Mick Grierson, and Rebecca Fiebrink. 2015. Using interactive machine learning to support interface development through workshops with disabled people. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 251–254.

[56] Andrea Kleinsmith and Marco Gillies. 2013. Customizing by doing for responsive video game characters. *International Journal of Human-Computer Studies* 71, 7-8 (2013), 775–784.

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

[58] Janin Koch. 2017. Design implications for Designing with a Collaborative AI. (2017).

[59] Janin Koch, Andrés Lucero, Lena Hegemann, and Antti Oulasvirta. 2019. May AI?: Design Ideation with Cooperative Contextual Bandits. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 633.

[60] Janin Koch and Antti Oulasvirta. 2018. Group Cognition and Collaborative AI. In *Human and Machine Learning*. Springer, 293–312.

[61] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[62] Yuki Koyama. 2016. Computational design driven by aesthetic preference. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 1–4.

[63] Ranjitha Kumar, Jerry O Talton, Salman Ahmad, and Scott R Klemmer. 2011. Bricolage: example-based retargeting for web design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2197–2206.

[64] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and Usability Engineering Group*. Springer, 63–76.

[65] Yuxi Li. 2018. Deep reinforcement learning. *arXiv preprint arXiv:1810.06339* (2018).

[66] Changchun Liu, Pramila Agrawal, Nilanjan Sarkar, and Shuo Chen. 2009. Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback. *International Journal of Human-Computer Interaction* 25, 6 (2009), 506–529.

[67] Wanyu Liu, Rafael Lucas d'Oliveira, Michel Beaudouin-Lafon, and Olivier Rioul. 2017. Bignav: Bayesian information gain for guiding multiscale navigation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 5869–5880.

[68] J Derek Lomas, Jodi Forlizzi, Nikhil Poonwala, Nirmal Patel, Sharan Shodhan, Kishan Patel, Ken Koedinger, and Emma Brunskill. 2016. Interface design optimization as a multi-armed bandit problem. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 4142–4153.

[69] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[70] Wendy E Mackay. 1990. *Users and customizable software: A co-adaptive phenomenon*. Ph.D. Dissertation. Citeseer.

[71] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[72] Stefano Delle Monache, Pietro Polotti, and Davide Rocchesso. 2010. A toolkit for explorations in sonic interaction design. In *Proceedings of the 5th audio mostly conference: a conference on interaction with sound*. ACM, 1.

[73] Yael Niv. 2009. Reinforcement learning in the brain. *Journal of Mathematical Psychology* 53, 3 (2009), 139–154.

[74] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. 2016. Deep exploration via bootstrapped DQN. In *Advances in neural information processing systems*. 4026–4034.

[75] François Pachet, Pierre Roy, Julian Moreira, and Mark d'Inverno. 2013. Reflexive loopers for solo musical improvisation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2205–2208.

[76] Kayur Patel, Steven M Drucker, James Fogarty, Ashish Kapoor, and Desney S Tan. 2011. Using multiple models to understand data. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22. 1723.

[77] Jonas Frich Pedersen, Michael Mose Biskjaer, and Peter Dalsgaard. 2018. Twenty Years of Creativity Research in Human-Computer Interaction: Current State and Future Directions. In *Designing Interactive Systems*. Association for Computing Machinery (ACM).

[78] Claire Petitmengin. 2006. Describing one's subjective experience in the second person: An interview method for the science of consciousness. *Phenomenology and the Cognitive sciences* 5, 3-4 (2006), 229–269.

[79] Ivan Poupyrev, Michael J Lyons, Sidney Fels, et al. 2001. New interfaces for musical expression. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*. ACM, 491–492.

[80] Landy Rajaonarivo, Matthieu Courgeon, Eric Maisel, and Pierre De Loor. 2017. Inline Co-Evolution between Users and Information Presentation for Data Exploration. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, 215–219.
[81] Mitchel Resnick. 2007. All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*. ACM, 1–6.
[82] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. Design principles for tools to support creative thinking. *Working Paper* (2005).
[83] Horst WJ Rittel. 1972. *On the Planning Crisis: Systems Analysis of the "First and Second Generations"*. Institute of Urban and Regional Development.
[84] Tuukka Ruotsalo, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski. 2014. Interactive Intent Modeling: Information Discovery Beyond Search. *Commun. ACM* 58, 1 (Dec. 2014), 86–92. https://doi.org/10.1145/2656334
[85] Diemo Schwarz and Norbert Schnell. 2009. Sound search by content-based navigation in large databases. In *Sound and Music Computing (SMC)*. 1–1.
[86] Hugo Scurto, Frédéric Bevilacqua, and Baptiste Caramiaux. 2018. Perceiving Agent Collaborative Sonic Exploration In Interactive Reinforcement Learning. In *Proceedings of the 15th Sound and Music Computing Conference (SMC 2018)*.
[87] Hugo Scurto, Rebecca Fiebrink, et al. 2016. Grab-and-play mapping: Creative machine learning approaches for musical inclusion and exploration. In *Proceedings of the 2016 International Computer Music Conference*.
[88] Burr Settles. 2010. Active learning literature survey. *University of Wisconsin, Madison* 52, 55-66 (2010), 11.
[89] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. 2016. Taking the human out of the loop: A review of bayesian optimization. *Proc. IEEE* 104, 1 (2016), 148–175.
[90] Michael Shilman, Desney S Tan, and Patrice Simard. 2006. CueTIP: a mixed-initiative interface for correcting handwriting errors. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. 323–332.
[91] Ben Shneiderman. 2007. Creativity support tools: Accelerating discovery and innovation. *Commun. ACM* 50, 12 (2007), 20–32.
[92] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484.
[93] Malcolm Strens. 2000. A Bayesian framework for reinforcement learning. In *ICML*, Vol. 2000. 943–950.
[94] Simone Stumpf, Vidya Rajaram, Lida Li, Margaret Burnett, Thomas Dietterich, Erin Sullivan, Russell Drummond, and Jonathan Herlocker. 2007. Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on Intelligent user interfaces*. ACM, 82–91.
[95] Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies* 67, 8 (2009), 639–662.
[96] Harini Suresh and John V Guttag. 2019. A Framework for Understanding Unintended Consequences of Machine Learning. *arXiv preprint arXiv:1901.10002* (2019).
[97] Richard S Sutton and Andrew G Barto. 2011. *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.
[98] Andrea L Thomaz and Cynthia Breazeal. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence* 172, 6-7 (2008), 716–737.
[99] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. 2017. Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces. *arXiv preprint arXiv:1709.10163* (2017).
[100] Christopher John Cornish Hellaby Watkins. 1989. Learning from delayed rewards. (1989).
[101] Geraint A Wiggins. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19, 7 (2006), 449–458.
[102] Weng-Keen Wong, Ian Oberst, Shubhomoy Das, Travis Moore, Simone Stumpf, Kevin McIntosh, and Margaret Burnett. 2011. End-user feature labeling: A locally-weighted regression approach. In *Proceedings of the 16th international conference on Intelligent user interfaces*. 115–124.
[103] Matthew Wright. 2005. Open Sound Control: an enabling technology for musical networking. *Organised Sound* 10, 3 (2005), 193–200.
[104] Qian Yang, Nikola Banovic, and John Zimmerman. 2018. Mapping Machine Learning Advances from HCI Research to Reveal Starting Places for Design Innovation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 130.
[105] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating How Experienced UX Designers Effectively Work with Machine Learning. In *Proceedings of the 2018 Designing Interactive Systems Conference*. ACM, 585–596.
[106] Georgios N Yannakakis, Antonios Liapis, and Constantine Alexopoulos. 2014. Mixed-initiative co-creativity.. In *FDG*.

[107] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. 2015. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 86.

[108] Bruno Zamborlin, Frederic Bevilacqua, Marco Gillies, and Mark D'inverno. 2014. Fluid gesture interaction design: Applications of continuous recognition for the design of modern gestural interfaces. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 3, 4 (2014), 22.

[109] Xiang Sean Zhou and Thomas S Huang. 2003. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems* 8, 6 (2003), 536–544.

APPENDIX A

The TAMER [57] and Deep TAMER [99] algorithms can be seen as value-based algorithms. They have been applied in settings that allow to quickly learn a policy on episodic tasks (small game environments or physical models) and aim to maximise direct human reward. This opposed to the traditional RL training objective to maximise the discounted sum of future rewards. These algorithms learn the human reward function R using an artificial neural network and construct a policy from R taking greedy actions. In addition, to accommodate sparse and delayed rewards from larger user response times, the algorithms include a weighting function $u(t)$ to past state trajectories and a replay memory in the case of Deep TAMER. Specifically, while traditional RL algorithms aim to optimise the Mean-Square Error (MSE) loss

$$MSE = \left[R_{t+1} + \gamma q(S_{t+1}, A_{t+1}, \mathbf{w}_t) - q(S_t, A_t, \mathbf{w}_t)\right]^2, \tag{1}$$

with R_t the reward at time t, γ the discount rate, and $q(S_t, A_t, \mathbf{w}_t)$ the computed state-action value function with parameters $\mathbf{w}$, (Deep) TAMER aims to optimise

$$MSE = u(t)\left[R_t - \hat{R}(S_t, A_t)\right]^2 \tag{2}$$

with R_t and $u(t)$ respectively the user-provided feedback and weighting function at time t, and $\hat{R}(S_t, A_t)$ the average reward.

APPENDIX B

Deep neural network [99]	**Agent** [99]
# hidden layers = 2	state dim. $n = 10$
# units per layer = 100	$s_i \in [0, 1], 0 \leq i \leq n$
batch size = 32	$a_i = 0.01, 0 \leq i \leq n$
learning rate $\alpha = 0.002$	reward value $\lvert R\rvert = 1$
replay memory $\mathcal{D} = 700$	reward length = 10

Exploration [11, 99]	**Density** $\hat{p}_{\phi'}()$
ε decay = 2000	# tiles = 64
ε start = 0.1	tile size = 0.4
ε end = 0.0	$C = 0.01$ [11]
action freq. = 10 Hz	$\beta = 1$ [11]

Table 1. Hyper-parameters for Deep TAMER (left) and the Co-Explorer exploration behaviour (right).

APPENDIX C

The exploration behaviour implemented in the Co-Explorer is based on the notion of optimistic exploration (the assumption that the unknown is good for the agent) and the addition of an 'exploration bonus' to the reward. As shown in [11], this exploration bonus can be based on an estimation of state-visitation counts called the pseudo-count $\hat{V}(S)$ and a total visit pseudo-count $\hat{v}$ using a density model $\hat{p}_\phi(S)$. Formally, one calculates the total reward with bonus R_t^+ as

$$R_t^+ = R_t + \beta \sqrt{\frac{1}{\hat{V}(S_t) + C}} \tag{3}$$

with

$$\hat{V}(S_t) = \hat{v}\hat{p}_\phi(S_t) \tag{4}$$

$$\hat{v} = \frac{1 - \hat{p}_{\phi'}(S_t)}{\hat{p}_{\phi'}(S_t) - \hat{p}_\phi(S_t)}\hat{p}_\phi(S_t) \tag{5}$$

with β and C pre-defined constants. We used tile-coding to estimate a density model $\hat{p}_\phi(S)$ over the high-dimensional state spaces considered in our work.

APPENDIX D

ALGORITHM 1: Deep TAMER with exploration bonus and user controls for estimating $\hat{R}() \approx R()$.

Input: reward function $\hat{R}(S, A, \mathbf{w})$, policy $\pi()$ as $\varepsilon-$greedy with exponential decay, reward distribution function $Env\_dist(R) = R * exp(-t), 0 \leq t \leq R_{length}$;

Initialise: weights $\mathbf{w} = \mathbf{0}$, average-reward $\hat{R} = 0$, $s_i(t = 0) = 0.5$ for $0 \leq i \leq n$ and $A(t = 0) = \pi(S(t = 0))$, $x_j = 0, 0 \leq j \leq R_{length}$;

while *running* **do** `// Start autonomous exploration mode`

- Take action A_t and observe next state S_{t+1} ;
- Select new action $A_{t+1} \sim \pi(\cdot|S_{t+1})$;
- Store $(S_{t+1}, A_{t+1}, 0)$ in reward length vector x (R_{t+1} stored as 0);
- Update density model $\hat{p}()$;
- Observe reward as R_{t+1} ;
- $S_t \leftarrow S_{t+1}$;
- $A_t \leftarrow A_{t+1}$;
- **if** $R \neq 0$ **and** $t > R_{length}$ **then** `// Train on user feedback + exploration bonus`
  - Compute guiding feedback $x = Env\_dist(R)$;
  - Store x in $\mathcal{D}$;
  - Compute $\hat{R}_{t+1}$ using SGD [99] and x ;
- **else if** $|\mathcal{D}| > 2 * batchsize$ **then** `// Train on past user feedback`
  - $\mathcal{D}_{t+1}$ = random sample from $\mathcal{D}$;
  - Compute $\hat{R}_{t+1}$ using SGD [99] and $\mathcal{D}_{t+1}$;
- **else if** $t > R_{length}$ **then** `// Train on exploration bonus`
  - Compute $\hat{R}_{t+1}$ using SGD [99] and R^+ ;
- **while** *Paused* **do** `// Stop autonomous exploration mode, allow direct manipulation`
  - agent.get_currentstate() ;
- **end**
- **if** *Change_zone* **then** `// Change zone state command`
  - **for** i *in range*$(n_{samples})$ **do**
    - Randomly sample state s_i ;
    - Evaluate predictiongain$(s_i) = log(\hat{p}_{t+1}(s_i)) - log(\hat{p}_t(s_i))$;
  - **end**
  - $S_t = argmax(predictiongain(s_i))$;
- **if** *Zone_feedback* **then** `// Zone feedback computation`
  - $S_{00} = x[0]$ and $A_{00} = Zone\_feedback$;
  - **for** i *in range*(R_{length}) **do**
    - **for** j *in range*$(|S|)$ **do**
      - Take action A_{ij} and observe state S_{ij} ;
      - Store S_{ij} in $\mathcal{D}_{t+1}$;
    - **end**
  - **end**
  - Compute $\hat{R}_{t+1}$ using SGD [99] and $\mathcal{D}_{t+1}$;
  - $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{t+1}$;

end

APPENDIX E

In a complementary work, we validated the success of our deep reinforcement learning method for human parameter exploration, by proving that guiding the *Co-Explorer* better supports user creativity than simply using a standard parametric interface. We recruited 14 participants (aged between 22 and 42 years old; 3 Female and 11 Male) with varying background in sound design (such as sound engineering students, (post-)doctoral researchers in computer music, sound engineers and IT researchers/developers).

Procedure

Participants were asked to explore a parameter sound space by successively using two types of interfaces: standard parametric interface, and guiding feedback to the *Co-Explorer*. The succession of interfaces was set automatically by our experimental setup, which alternated each interface three times in a row (P—C—P—C—P—C). This alternation was chosen to remove bias towards user preference of an interaction type resulting from knowledge of the timbral space. Each interface was shown to participants during 5 minutes, which made the experiment last 18 minutes. The experiment was preceded by a short interface demonstration. We defined the parameter space by selecting ten parameters from a commercial VST. Participants were also asked to save an arbitrary number of varied sounds they appreciated during exploration, following their creative will.

Analysis

At the end of the experiment, participants received a questionnaire in which they were asked to evaluate their experience and the interaction itself, followed by a brief and informal discussion of their experience. Participants were asked to compare the parametric interface and *Co-Explorer* using scores between contrasting adjectives using a Likert scale of 1-5 (see Table 2). These adjectives were based on the user questionnaire developed by Laugwitz et al., identifying several adjectives representing criteria in classes such as perceived ergonomic quality, perceived hedonic quality (hedonic quality focuses on non-task oriented quality aspects, for example the originality of the design or the beauty of the interface) and perceived attractiveness of a product [64]. We replaced the counterpart adjective for “understable” by “non-understable”, instead of “ambiguous” as proposed in [64]. Our motivation is that we wanted to insert a negative adjective as a positive characteristic for creativity.

Classes	**Criteria**	**Adjectives**	
Hedonic quality	Novelty	Conventional	Inventive
		Dull	Creative
	Stimulation	Demotivating	Motivating
		Boring	Exciting
Ergonomic quality	Perspicuity	Confusing	Clear
		Ambiguous	Understandable
	Dependability	Obstructive	Supportive
	Efficiency	Inefficient	Efficient
Attractiveness		Annoying	Enjoyable

Table 2. Contrasting adjectives used to measure and compare the experiences of guiding the *Co-Explorer* and using a standard parametric interface in human parameter exploration (based on [64]).

Results

We performed a 2-factor ANOVA with replication using the scores of each participant for all adjectives, for both the *Co-Explorer* and parametric interface setups. We found that the *Co-Explorer* had significantly better hedonic qualities than the parametric interface [$F = 4.95, p < 0.05$]. This may validate our hypothesis that the *Co-Explorer* successfully supports user creativity in parameter exploration. Alternatively, the parametric interface proved to be more ergonomic than the *Co-Explorer* [$F = 22.1, p < 0.001$]. This is not a surprise, as all participants reported experience in using parametric interfaces for sound design. Last and interestingly, attractiveness for both interfaces modes was not significantly different [$F = 2.23, p = 0.15$]. This may suggest that the *Co-Explorer* did not benefit from a positive effect due to its novelty compared to the parametric interface.

APPENDIX F

Collaboration	*(not relevant here)*
Enjoyment	P5 (Section 6.1.3, first paragraph) P11 (Section 6.2.3, third paragraph)
Exploration	P1 (Section 6.1.3, second paragraph) P12 (Section 6.1.3, second paragraph)
Expressiveness	P2 (Section 6.1.3, second paragraph) P8 (Section 6.1.3, fourth paragraph) P5 (Section 6.2.3, first paragraph)
Immersion	P9 (Section 6.2.3, first paragraph) P5 (Section 6.2.3, third paragraph)
Results Worth Effort	P4 (Section 6.1.3, fifth paragraph) P10 (Section 6.1.3, fifth paragraph) P5 (Section 6.2.3, first paragraph)

Table 3. Preliminary summary that maps participants' quotes in the evaluation workshop to Creativity Support Index dimensions [17].