



# AutoDOViz: Human-Centered Automation for Decision Optimization

Daniel Karl I. Weidele  
Shazia Afzal  
Abel N. Valente  
Cole Makuch  
Owen Cornec  
Long Vu  
Dharmashankar Subramanian  
Werner Geyer  
daniel.karl@ibm.com  
IBM Research  
USA

Rahul Nair  
Inge Vejsbjerg  
Radu Marinescu  
Paulito Palmes  
Elizabeth M. Daly  
IBM Research  
Ireland

Loraine Franke  
Daniel Haehn  
University of Massachusetts Boston  
USA

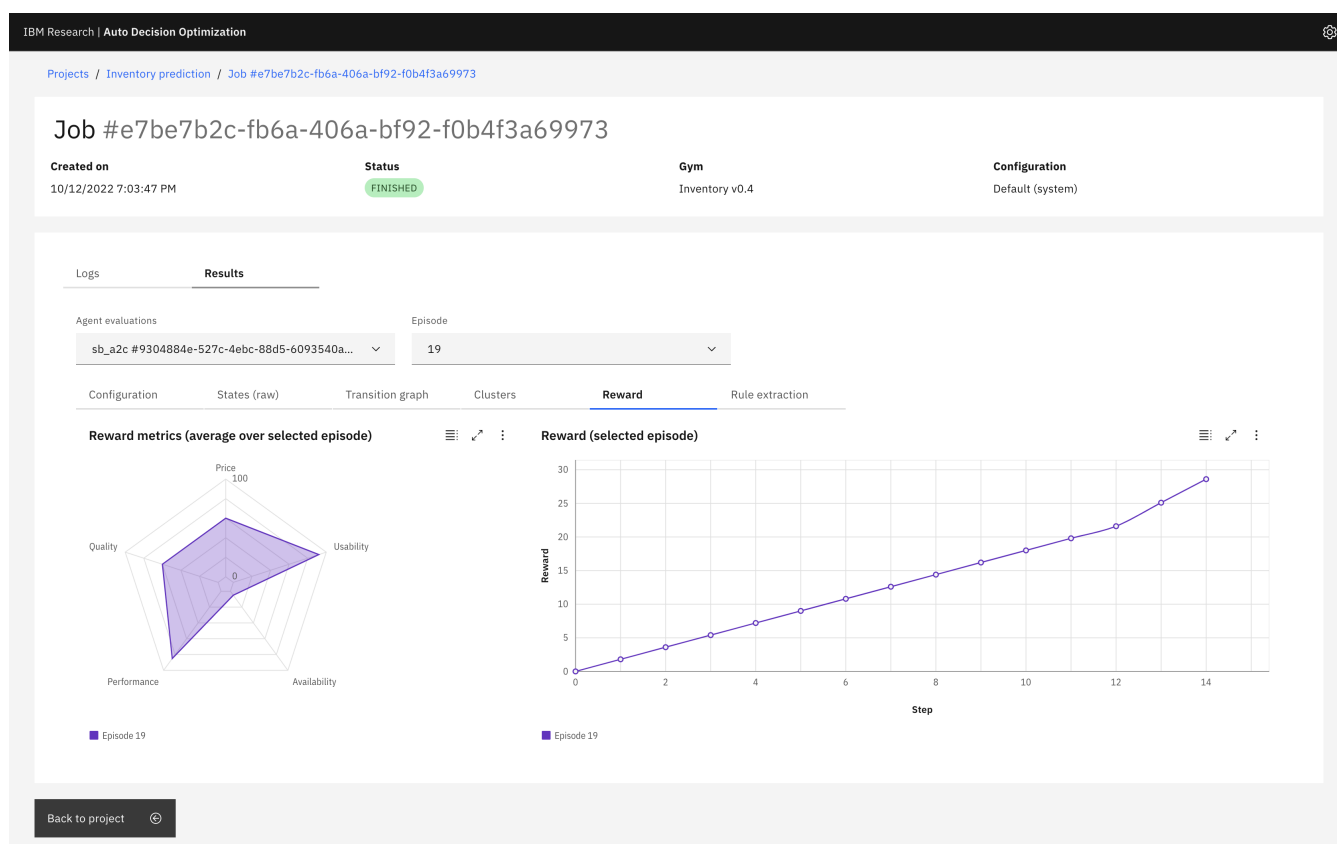


Figure 1: Reinforcement learning agent evaluation in AutoDOViz.

## ABSTRACT

We present AutoDOViz, an interactive user interface for automated decision optimization (AutoDO) using reinforcement learning (RL).



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs International 4.0 License.

IUI '23, March 27–31, 2023, Sydney, NSW, Australia

© 2023 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0106-1/23/03.  
<https://doi.org/10.1145/3581641.3584094>

Decision optimization (DO) has classically been practiced by dedicated DO researchers [43] where experts need to spend long periods of time fine tuning a solution through trial-and-error. AutoML pipeline search has sought to make it easier for a data scientist to find the best machine learning pipeline by leveraging automation to search and tune the solution. More recently, these advances have been applied to the domain of AutoDO [36], with a similar goal to find the best reinforcement learning pipeline through algorithm selection and parameter tuning. However, Decision Optimization requires significantly more complex problem specification when compared to an ML problem. AutoDOViz seeks to lower the barrier of entry for data scientists in problem specification for reinforcement learning problems, leverage the benefits of AutoDO algorithms for RL pipeline search and finally, create visualizations and policy insights in order to facilitate the typical interactive nature when communicating problem formulation and solution proposals between DO experts and domain experts. In this paper, we report our findings from semi-structured expert interviews with DO practitioners as well as business consultants, leading to design requirements for human-centered automation for DO with RL. We evaluate a system implementation with data scientists and find that they are significantly more open to engage in DO after using our proposed solution. AutoDOViz further increases trust in RL agent models and makes the automated training and evaluation process more comprehensible. As shown for other automation in ML tasks [33, 59], we also conclude automation of RL for DO can benefit from user and vice-versa when the interface promotes human-in-the-loop.

## CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Visualization**; • **Computer systems organization** → **Real-time systems**; • **Computing methodologies** → **Reinforcement learning**; **Artificial intelligence**; • **Applied computing** → **Enterprise computing**.

## KEYWORDS

decision optimization, reinforcement learning, automation

### ACM Reference Format:

Daniel Karl I. Weidele, Shazia Afzal, Abel N. Valente, Cole Makuch, Owen Cornec, Long Vu, Dharmashankar Subramanian, Werner Geyer, Rahul Nair, Inge Vejsbjerg, Radu Marinescu, Paulito Palmes, Elizabeth M. Daly, Loraine Franke, and Daniel Haehn. 2023. AutoDOViz: Human-Centered Automation for Decision Optimization. In *28th International Conference on Intelligent User Interfaces (IUI '23)*, March 27–31, 2023, Sydney, NSW, Australia. ACM, New York, NY, USA, 17 pages. <https://doi.org/10.1145/3581641.3584094>

## 1 INTRODUCTION

Decision optimization (DO) is a discipline tightly integrated with business domain users to enable better, faster, and more efficient decision-making. DO systems aim to recommend single decisions or present multiple alternatives to users and improve continuously while in production. DO finds application in virtually every industry and business practice that involves decision-making and optimization of resource allocation, such as uncertain demand in inventory networks or intelligent scheduling of heating, ventilation, and air conditioning in building control [57]. However, DO

is a reasonably specialized, knowledge-oriented field that requires expertise in operations research and mathematical optimization and a deep understanding of specific business processes. This combination is challenging to find in individuals, so tight integration across teams or specifically tailored software solutions is necessary to succeed in DO projects. Recent advances in reinforcement learning (RL) have created new opportunities to add to the DO toolbox [36]. RL gyms flexibly encode problem settings in which RL agents decide actions to optimize a given reward function. Furthermore, advances in automation have shown how the process of designing high-quality machine learning pipelines accelerates the democratization of Artificial Intelligence (AI) from machine learning researchers and engineers towards data scientists and business users [10, 15, 25, 32, 42, 44]. In this paper, we want to take the next step towards a *user experience*<sup>1</sup> for automation of decision optimization with RL to accelerate democratization from DO researchers to data scientists and business users. Our goal is to empower a broader set of people to develop optimization models and make automated decision-making more accessible overall. A fundamental task on this path is to precisely understand the decision optimization process, the tools used, and associated challenges to identify opportunities and scope for automation and support. Additionally, characterizing the range of business domain stakeholders and users, their skills, goals, and expectations can help design a more practical and effective solution. We, therefore, conducted extensive semi-structured one-on-one interviews with DO practitioners and business domain experts to provide insights into these questions. Regarding business experts, we aimed to characterize potential stakeholders, such as their ability to use such a system as well as their needs, goals, and expectations, which can help design a solution that is more practical and effective as well as enhancing the democratization of AI for our target group, such as the general data scientist. Based on our requirements, we propose a novel system design for automated decision optimization using reinforcement learning, AutoDOViz. AutoDOViz builds on the AutoDO core engine [36], which leverages search techniques to automate algorithm selection and hyperparameter optimization. In AutoDOViz, we bring together novel UI interfaces to simplify the problem formulation process, leveraging the AutoDO capabilities to generate optimized agents and, finally, policy visualizations and metrics to enable better communication between the agent designer and the domain expert to align the end solution with domain expert needs and inputs. We rigorously test our design in an evaluation study with data scientists and report our findings, which also leads to future work.

*The contribution of this paper is 4-fold.* After reviewing related work in section 2, we present insights from expert interviews with (1) decision optimization researchers as well as (2) business domain consultants, which lead to system design requirements in section 3. Section 4 provides details about our (3) implementation of a user experience for an automated DO system with RL (*"AutoDOViz"*). Lastly, section 5 contains the evaluation of AutoDOViz in the form

<sup>1</sup>Throughout this paper we use the term *user experience* in a broader sense for the system design, its implementation and usability but would like to point the reader in acknowledgement to more granular differentiation as suggested in previous work [11, 20, 31].

of (4) a user study with data scientists. We discuss our results in section 6 before drawing conclusions and suggesting directions for future work in section 7.

## 2 RELATED WORK

### 2.1 Decision Optimization and Reinforcement Learning

In decision optimization, the goal is to find the best solution for an optimization problem according to a given criterion among a set of possible solutions. Therefore, the use of optimization algorithms as general step-by-step procedures plays a crucial role. Optimization algorithms are exact if they can find the optimal solution to the problem. They are heuristic if they propose an appropriate solution that is not necessarily optimal in the solution set [3]. Optimization problems are commonly found in scheduling problems, transportation, logistics, or supply chain planning [3]. In general, decision optimization can benefit from creating interactive methods, especially dynamic interaction between the user and the system [9]. Interaction with end-users makes the optimization procedure more effective by enriching the selected optimization model or informing the decision maker regarding available solutions proposed by the optimization system [38]. Using machine learning for combinatorial optimization to learn an approximate relation between decisions and their impact on the system plays a crucial role in current research. Such systems suggest optimal decision outcomes in complex real-world settings by taking advantage of the recent developments in artificial intelligence [35]. Recent literature confirms that agent-based approaches can successfully cope with a large spectrum of optimization problems. Agent-based computing has computational advantages as problems can be divided into multiple sub-problems, which can then be solved by massive parallelization (e.g., involving GPU). This is preferable for large problem spaces [3]. Earlier approaches also already proposed solving optimization problems, such as dynamic job shop scheduling, by using reinforcement learning agents [2].

Reinforcement learning (RL) or deep reinforcement learning (DRL) have become prominent techniques in machine learning to address sequential decision-making problems by training autonomous agents [26, 50]. A trained model is usually referred to as an independent agent that interacts in a particular environment with specific predefined actions. RL uses a trial-and-error approach to learn an optimal policy or behavioral strategy [4]. The RL algorithm surveys the optimal action space given observed feedback on decisions, for example, a reward from the environment, to learn which actions are beneficial in a given state. Primarily, RL's success originates from RL agents' ability to solve decision-making problems in complex and uncertain environments. RL is popular in various fields such as autonomous driving, robotics, health, finance, smart grids, video games, and education [34]. Other applications of RL include inventory management with multiple stages and suppliers with strict lead times under demand uncertainty or control problems such as autonomous manufacturing operations with resource allocation constraints. RL can also reduce the need for repetitive human interaction by decreasing manual work like calibrating or monitoring a system, allowing systems to adapt faster to

changes [47]. RL agents aim to maximize the total expected reward by learning an optimal policy.

However, RL agents might lack semantic knowledge on the current goal, i.e., *why* some actions and elements are preferable over others [48]. The agent should be able to explain their reasoning and decision-making process to users, which could increase the user's trust towards the RL agent. Explainable Reinforcement Learning (XRL) is a form of Explainable Artificial Intelligence (XAI), focusing on the importance of providing explanations to gain the trust of users [8, 14, 41, 46]. Several methods have been proposed to provide greater transparency for users [12, 13, 51]. With the increasing use of intelligent agents, it is essential to introduce methods to uncover agent behaviors and limitations. For example, algorithms provide users a behavioral summary of the agent in different situations by extracting critical trajectories from simulations of the agent [1]. Gajcin et al. [16] further suggests that understanding the differences in strategies between RL policies is required to enable users to choose between those policies.

### 2.2 Visual Analytics for Reinforcement Learning

Next to XRL, visual analytics is a promising technique for making the RL training and evaluation process more understandable and trustworthy. The use of visual analytics tools and interactive user interfaces provides insights on decisions and representations [23, 60]. Within RL, ReLVis [47] is a visual analytics tool designed to help data scientists keep track of RL experimentation. It visualizes relations between hyperparameter changes and outcomes, such as behaviors associated with specific rewards and how agent behavior evolves. DRLViz [24] is an interface to interpret internal memory of an agent. DRLIVE [54] is a visual analytics system for RL that tracks agents through interactive visualizations and further diagnoses the DRL model by perturbing its inputs. PolicyExplainer [39] is a visual analytics interface that allows users to query an autonomous agent directly. The tool extracts data from a trained agent, summarizes the optimal policy, analyzes states and expected rewards, and visualizes a trajectory to see the agent's progression. Moreover, tools for the visualization of specific RL algorithms have been presented. For example, DQNViz [53] visualizes states of Deep Q-networks (DQNs) during the training of a deep RL agent. DynamicsExplorer [21] diagnoses an RL policy under different dynamic settings. Direct visualization of the internals of RL agents has been discussed in past research for games [19], including dimensionality reduction, reward curves, or Jacobian saliency maps [40, 61]. However, they all point out the growing need for human-interpretable explanations of deep RL agents and difficulty for non-experts to interpret. Especially in real-world scenarios, where design problems consist of more than one conflicting or cooperative objective ("*multi-objective optimization problems*"), visualization can be challenging [9]. Overall, more research and exploration on how to connect RL with visualization is required [55].

### 2.3 User Experiences for Automation in Machine Learning

Novel methods have been developed to automate parts of the data science process, from data pre-processing, feature engineering, and

model selection to hyperparameter optimization (HPO) [27, 28, 30, 56, 63]. While much work has been shown to help solve specific problems, only a few can run in a fully automated and interactive way to enhance usability and decrease manual work. Mainly HPO relies on data scientists' experience and the dataset itself. Therefore, automated methods for HPO are also an active area of research in AutoML [17]. Recently, new research has emerged to automate the full artificial intelligence lifecycle, so-called Automated Artificial Intelligence (AutoAI) or Automated machine learning (AutoML). The goal is to enable domain experts to automatically build ML applications without extensive statistics or AI knowledge. Furthermore, such systems should reduce repetitive, manual workloads for data scientists and shift their time spent on summarizing and controlling visualization techniques [58, 59]. Commercial AutoML and AutoAI products have been released as fully managed machine learning services. Examples are Amazon SageMaker[10], Azure AutoML[42], H2O AutoML [32], Google AutoML [5] and IBM AutoAI [49, 52, 59]. Additionally, open-source software packages are available, such as AutoKeras [25], TPOT [44], Auto-Sklearn [15] and LALE [22]. Recently, Marinescu et al. [36] have proposed an automation engine towards data- or knowledge-driven dynamic optimization problems using reinforcement learning.

### 3 EXPLORATIVE INTERVIEWS

#### 3.1 Decision Optimization Researchers' perspective

Semi-structured interviews were conducted to collect individual opinions and experiences of DO practitioners while allowing flexibility for deeper discussions or exploration of people's views or narratives. Initial research questions were framed as prompts to guide the interview process but were not strictly adhered to formally or linearly. Instead, the interviews evolved into free-flowing conversations where participants delved into their subjective experiences and reflected on topics of interest.

**3.1.1 Participants.** These interviews were conducted with six people with extensive applied industry experience in DO. Five of the six people we interviewed had over 20 years of experience in optimization and vast experience developing and consulting optimization solutions for clients. Only one was a data scientist but had practical experience working on optimization problems. All interviews were conducted remotely and recorded using standard commercial video conference software with an automatic transcribing facility. The interviews took about 60 minutes on average, giving rich insights into our research questions. We know that this sample size is limited in terms of the number of participants and range of optimization experience. However, since our interview participants are experts with extensive experience in optimization, their views can be considered reliable and valid for the early insights we were looking for. Note that optimization is an extremely niche domain. One of the aims of AutoDO is to increase the accessibility of DO solutions to a broader user base that may not have significant OR/DO expertise. So, it is meaningful to engage with DO experts to design for the non-DO experts we aim for as target users.

**3.1.2 Analysis.** Thematic analysis is a popular method to uncover meanings from qualitative descriptions of experiences and contexts.

Using thematic analysis, the interview data was analyzed to find patterns and themes of interest. The initial set of research questions guided the coding of this data and helped to formulate some actionable insights.

**3.1.3 Results.** The interviews' findings and implications for an automated solution are structured by the various themes and dimensions we identified during the analysis.

**Domains.** Our interview participants reported first-hand experience across various domains like energy, transportation, aviation, retail, finance, education, defense, semiconductors, telecommunication, and healthcare. This emphasizes the relevance of decision optimization and its potential to help businesses drive timely and efficient decisions. Optimization models are required for tasks like determining production capacity, planning inventory, supply chain routing, operations, and maintenance scheduling, workforce planning, and manufacturing and warehouse management. The challenge is that these models are sensitive to a given business context, and they must be customized considering the constraints and objectives of each business or client. This makes it difficult to think of a generic DO model that is portable across contexts and domains. However, it appears that various optimization tasks can be broadly categorized into the following four categories based on factors like time and precedence: resource assignment (1), selection & allocation (2), supply & demand planning (3) and scheduling (4).

For automated DO, this simplifies the problem space into a categorization that can enable building resources aligned to these four generic templates [R1]. For example, visualizations that are an important mode of communicating results with the stakeholders can be built to incorporate the essential characteristics and requirements for each category instead of customizing each business problem [R2].

**User Identification and Characterization.** Various internal and external stakeholders can be identified based on their needs and goals in generating a DO model. The interview data's roles, titles, and personas ranged from business representatives, business analysts, data engineers, data scientists, planners, quantitative analysts, developers, IT specialists, business consultants, and salespeople to one or more optimization experts. This is a broad spectrum of technical and non-technical expertise with clearly different but highly dependent roles in the workflow. This implies that any automated solution should ideally be collaborative if it intends to be a one-stop workbench for creating an end-to-end DO model [R3]. This, in turn, necessitates the characterization of skills and identification of goals explicitly when designing the user interface [R4]. One can also consider creating levels of complexity in the solution interface so that supported personas can easily navigate between more transparent and detailed or complex views of the pipeline. A hierarchical granularity or visibility into the execution flow may well serve the requirements of the diverse stakeholders involved at various stages of the solution process. Given the complexity of designing and implementing an end-end solution, it is likely that initial versions of automated DO can only support model development with explanations and a provision to import and export process modules (like input and output) for communicating with the broader set of stakeholders.

**Tools.** We found that DO practitioners use a range of tools, often building customized UIs for clients. The popular modelling languages used are OPL, Python, Java and even C++ with a clear preference towards OPL by our expert sample. Specifically, OPL seemed to be the choice for building quick prototypes for client demos. From informal notes, whiteboard discussions, excel sheets, to the use of special and general purpose modelling languages and enterprise software suites – there seems to be a lack of a powerful one-stop workspace that offers end-to-end workflow support. A need for interactive visualizations at various levels of granularity was a key missing piece of technology identified in our data. It is worthwhile to explore designing a solution that supports an end-end workflow providing users with an integrated framework for ease and usability [R5].

**Data Scientists as DO Practitioners.** Data scientists are being considered as the primary target users for AutoDO. Even though most data scientists have a mathematical background they generally use statistical or machine learning techniques to derive insights from data or build ML models. Since AutoDO is also data-driven, the data scientists are a natural choice to be potential end-users. Our interviews with DO experts sought to understand potentials for introducing DO as a method to data scientists and the challenges they might face. While it is uncommon for data scientists to solve DO problems, we did have one participant (ID 1) who had some experience in working on optimization problems. Interestingly, the challenges he faced had less to do with the modelling part but were mostly related to approaching the DO problems, framing or scoping them and explaining the solution to the clients. This echoes exactly with views of participant ID 2 who believes that optimization is just another technique that a data scientist can use but will require an understanding of the business and what one needs to model to be able to formulate the right problem. Understanding the business problem and not blindly taking data to try different things is what differentiates a data scientist from an optimization expert for ID 4. As ID 3 puts it – “DO is a different way of thinking that requires mapping between business to decision variables and constraints” and a lot depends on what would motivate a data scientist to consider DO for problem-solving (ID 4). This is an interesting aspect also mentioned by ID 5 saying that a “data scientist would be completely lost if they don’t already know that the problem is an optimization one” or even ID 6 mentioning that “Data scientists do not uncover optimization opportunities”. Highlighting the importance of business knowledge, ID 6 quantifies the role of a data scientist as relying about 70% on data and 30% on the business process and vice-versa for an optimization expert. For the former the question is about how to build an ML model from given data and infuse it into the existing process; while for an optimization expert a deep understanding of the business process and scoping it determines the data requirements. It is the thinking and refinement of the specified business problem within the given constraints that distinguish the optimization expert from the typical data scientist. To consider data scientists as target users for AutoDO depends a great deal on which aspects of the DO workflow are incorporated into it. Scoping or problem specification and model evaluation are the most challenging phases and in the real-world are conducted in close collaboration with other stakeholders like business analysts

or subject matter experts. In the standard DO process, testing and validation of the solution on KPIs or against various scenarios and use-cases determines its acceptability. A data scientist in the role of an optimization solution builder has neither the mandate nor the proficiency to take this decision individually.

**Leveraging Automation and AI.** We asked our participants about how automation or AI could help in overcoming some of the challenges and bottlenecks they currently face. We find that developers need tools to monitor performance and understand execution flow, while consumers need to trust and have confidence in the solution [R6]. Therefore, explanations and interactive visualizations that help monitor the execution and enable communication between all the stakeholders was unanimously called out. Testing the solution on historical data or using multiple scenarios, replaying decisions, ML guided tuning of solution, using visualizations to uncover bottlenecks, graphical interactive monitoring of solution, and gathering of insights are some areas where our participants desired tooling or automation support. Interestingly, there is a strong requirement for interactive UIs and rich visualizations in the community and AutoDO has an opportunity to make a convincing contribution in this space.

### 3.2 Business Domain Consultants’ perspective

We further aim to improve AutoDO’s relevance to real-world use cases by conducting semi-structured interviews with business consultants. We sought consultants with deep knowledge in particular industries, rather than generalists, to learn about potential optimization problems within that specific industry.

The interviews were based on the following six semi-structured questions:

- (1) What is your primary industry?
- (2) What kinds of clients do you work with in this industry?
- (3) What are main problems these clients face?
- (4) Have you seen integration of AI in this industry? What does it do?
- (5) Which problems in this industry cannot be solved with AI?
- (6) Reinforcement Learning trains agents via reward and punishment for actions and can be good at making discrete decisions such as driving instructions for self-driving cars, moves in games such as chess and checkers, or recommendations in the stock market. Can you think of decisions in your industry that would be particularly helpful to automate?

The interviews were scheduled for 30 minutes in length, and conducted remotely over video conference during the consultant’s workday with two interviewers in attendance.

A total of eight interviews were conducted with consultants (henceforward, ‘specialists’) specializing in Advertising, Agriculture, Automotive, Government (two consultants), Industrial Manufacturing, Oil, and Retail. Everyone except of the Oil specialist was hand-selected from a staff list of 547 global consultants. Our interviews consisted of three women and five men, ranging in age from early-career to mid-career. Information on age or race was not solicited.

**3.2.1 Summary of findings.** The **Advertising** specialist was an early-career generalist consultant who communicated that many

of the core tasks in advertising are still time-consuming manual processes, and approximated the breakdown of big data, AI, and human input for each stage of the digital advertising work cycle. They believed that algorithms are efficient at optimizing many decisions, but that humans were necessary to drive high-level strategy and focus algorithms on particular tasks. They gave specific examples of automated processes that they believe required human input to function optimally. Based on our explanation of RL in question six, they believed that RL could be most useful for identifying the types of content that resonate with people.

The **Agriculture** specialist was a mid-career consultant who grew up on a farm, had a formal education in Agronomy (agricultural consulting), and spent their entire pre-consulting career with agriculture firms. They claimed that farmers were generally distrustful of new technology, but that application of Big Data was increasingly common in agriculture. For example, farmers are now able to purchase crop insurance based on conditions such as the maximum outside temperature during a particular period, which is calculated using advanced underwriting models.

The **Automotive** specialist was a mid-career consultant with additional background in retail consulting. They identified current trends in the automotive industry and felt RL's impact in the automotive industry were mainly related to self-driving cars as referenced in question six. When challenged to propose applications unrelated to the driving experience, they suggested RL could help manufacturers identify how to configure future cars according to consumer preferences.

The first **Government** specialist had spent over a decade consulting for multiple government agencies. They have observed the demand for data within government agencies is changing. Particularly, they described how legacy systems had been adapted to meet modern demands for information access and transparency. They identified handling requests for security clearances as a potential application for RL. The second Government specialist had worked for various federal agencies, including defense and security agencies, before their time as a consultant. They believe that distributing and optimizing the use of resources across locations and agencies is the primary problem for the government organizations they had interacted with. They presented the potential issue of getting security clearance for AI models and programs. Their proposed RL use cases were for generating government reports and testing new applications.

The **Manufacturing** specialist had prior experience at a coal power and mining firm, and had primarily consulted for coal, oil, and large pharmaceutical companies during their consulting career. The key insight from this conversation was the importance of constant analysis of the entire supply chain. They believed that much of the technical innovation in the manufacturing industry is related to accurate modelling and analysis of when and how machines might fail, and that one of the biggest current problems in manufacturing is insufficient information. This specialist believed that AI could have the biggest impact on manufacturing by improving efficiency, and proposed use of RL to monitor the consumption, use, and disposal of raw materials according to material-specific considerations.

The **Oil** specialist was a geologist by training, and had spent their consulting career working with oil companies in and off the

coast of Texas. They identified pieces of the drilling process that are currently trial-and-error rather than data-driven, and offered that proper modelling and responses to models could significantly increase the longevity and output of wells. They also shared that oil infrastructure was an interconnected superstructure of equipment from many manufacturers, and navigating the supply chain required constant real-time communication between many different stakeholders.

The **Retail** specialist had experience with some of the world's largest big-box retailers and responsible for overseeing the creation and acquisition of multiple e-commerce retail organizations. They shared thoughts on the emerging issues presented by the demand for retail delivery, the current state of warehousing and changing philosophies on stocking, and how to render the in-store experience online.

**3.2.2 Discussion.** The consultants offered useful and nuanced insight into how AI is currently employed in their respective industries and potential future applications. We found that questions related specifically to RL yielded less actionable results than general discussion about AI, which leads us to believe that business people have a good sense for AI's capabilities, but a limited sense of the differences between training methods. It seems useful for a tool to provide insights on the RL process to end users, so that it becomes clear how the training process yields the trained model [R7].

The discussion of specialized problems within particular industries highlighted the usefulness of breaking down templates of DO problems by industry. An interface that helps users identify the type of business their problem is centered on would help them more easily locate relevant solutions [R8]. We also learned of types of non-industry-specific business processes that are very relevant to multiple or all industries. These kinds of general business DO problem templates should also be made available within industries, even when they are shared among different industries [R9].

### 3.3 Design Requirements

Capturing experts' needs and interests from our detailed semi-structured interviews, table 1 summarizes requirements for human-centered automation for DO with RL.

## 4 SYSTEM DESIGN

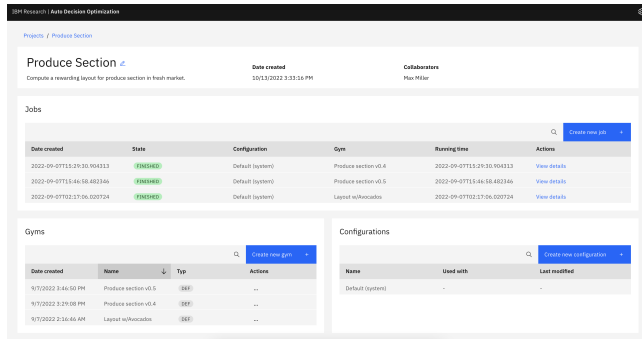
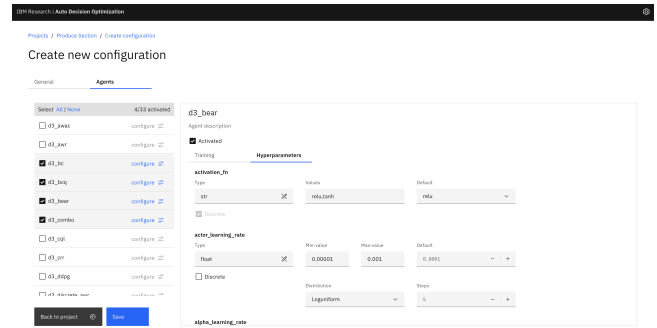
### 4.1 Integrating Core Engine and User Interface Elements

AutoDO [36] is an end-to-end system for automated reinforcement learning using a bi-level nested search to optimize RL pipelines. More specifically, in the first level, the system admits pipelines with different flavours of RL, namely online and offline methods combined with model-free and model-based approaches. In the inner level, it further admits a mixed discrete and continuous search space to tune the RL agent hyperparameters as well as the neural architecture choices. The system employs different search strategies to traverse this complex search space such as random search, Bayesian optimization or a more recent multi-fidelity strategy based on limited discrepancy search [29].

The input to the system consists of a collection of RL agents and either a gym-compatible environment or a dataset containing tuples

**Table 1: System requirements for human-centered automation for decision optimization.**

<b>R1</b>	Generic templates which align with common categories of DO problems.
<b>R2</b>	Visualizations for DO problem categories rather than for each business problem.
<b>R3</b>	Encourage collaboration between stakeholders.
<b>R4</b>	Characterization of skills and identification of goals for supported personas.
<b>R5</b>	Support end-to-end workflow with an integrated framework for ease and usability.
<b>R6</b>	Promote trust and confidence in the solution.
<b>R7</b>	Explain the RL training process for non-technical users.
<b>R8</b>	Classify templates by industry for ease of location.
<b>R9</b>	Provide templates for general business problems shared by multiple industries.

**Figure 2: Project dashboard in AutoDOViz Users can manage gyms, engine configurations and trigger RL agent training.****Figure 3: Optional AutoDO engine configuration in AutoDOViz. Users can toggle individual agents and set hyperparameter search constraints.**

$(s, a, r, s')$ , where  $s$  and  $s'$  represent the current and respectively the next state of the underlying Markov decision process,  $a$  is the current action and  $r$  is the reward. Optimization problems can be provided in the form of RL environments based on the open-source toolkit OpenAI Gym [7], a Python programming interface. The RL agent then acts on the OpenAI Gym in a series of episodes, where each episode starts with a user-defined initial state followed by interactions with the environment until a user-defined termination criteria is reached. The performance of RL agents is assessed via user-defined a reward function. The output consists of a the set of top  $k$  pipelines produced by the joint pipeline and hyperparameter optimizer and representing the best performing agents with their hyperparameters.

Based on this specification of the core engine, we identify three entities that need to be managed by the user of AutoDOViz: gyms (or environments), engine configurations, and resulting jobs (or executions). Regarding requirement **R3**, we also propose organizing these entities in *projects*, which can then be shared among stakeholders. For example, a business domain user does not necessarily need to be able to modify the gym implementation. However, they might want to trigger a gym execution and assess high-level visualizations. While gyms are only accessible within a project, configurations of the AutoDO engine can be shared across projects. Thus, users can bring along configurations that work well in other cases when they move on to a new project. Figure 2 shows the project dashboard with a clear layout and a simple table view for each of the three entities.

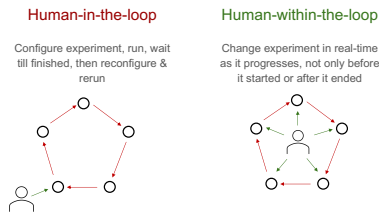
Figure 3 depicts the configuration wizard of AutoDOViz. The left hand side pane shows the list of selected RL agents. The right hand side shows the tunable hyperparameters of a particular agent (e.g., `d3_bear`). For each hyperparameter we specify the type, possible values if discrete or the range if continuous, as we well as the default value. The result of this configuration phase is the definition of the joint pipeline and hyperparameter search space traversed by the AutoDO engine.

We handle the user interface elements for gyms in more detail throughout sections 4.3 and 4.4, as well as for job results/executions in sections 4.5 and 4.6.

## 4.2 (Remote) Execution with Streaming Architecture

User experiences for automation often incorporate an explicit Human-in-the-loop (HitL) mechanism [18, 33, 38, 52, 59]. Users of HitL optimization systems are part of the optimization process after every turn of the automation (see Figure 4). In our implementation, specifically under consideration of **R5**, we aim to evolve this approach towards a more involved Human-*within*-the-loop (HwtL) process (Figure 4 right). With HitL being an important aspect of designing automated optimization systems, designing more towards HwtL should lead to beneficial progress of an intelligent user interface [38]. Since HwtL constitutes a very high technical challenge, in this prototype we currently only support real-time *tracking* of the AutoDO engine, which means the user already has full transparency of the process at any step of the process. As we





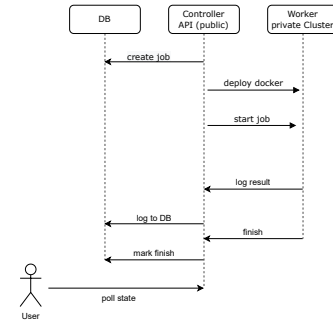
**Figure 4: Human-within-the-Loop.** Instead of round-based interaction at the beginning and end of each cycle (left), we aim to include the user in automation in real-time to promote high interactivity and usability.

will now describe details of our streaming architecture, we would like to highlight that full real-time *manipulation* of the engine is *not yet* part of this work, but will benefit from the same architectural design in a future iteration of the software. For example, users will then also be able to add/remove RL agents or modify hyperparameter search constraints in real-time during the execution of the engine to focus computational resources or to allow for quick intermediate *What-If?* scenarios.

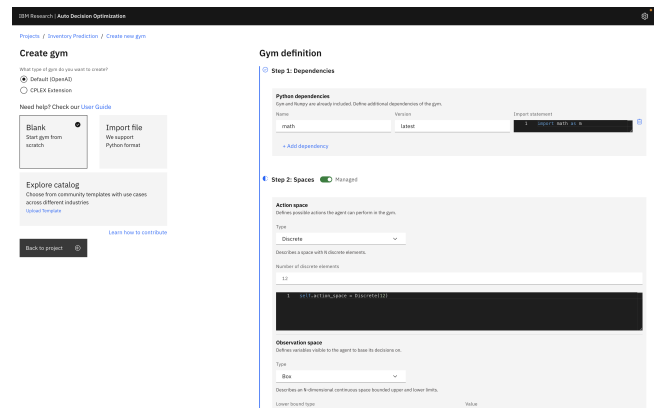
Figure 5 shows a sequence diagram of our proposed architecture towards HwtL featuring a controller plane in action. When users want to execute a job in our user interface, the controller manages workers, their execution, and logging of progress in real time. The controller first only creates an entry about the job to be executed in the database. In return, the controller receives a job ID. The controller then deploys a Dockerized version of the AutoDO engine in a custom cluster provided by the user or executes in a shared cluster. In both cases the job ID is handed to the worker, together with an API token. The worker starts execution of the AutoDO agent search given the selected gym and configuration in the database found via the job ID. During the execution, the worker can log events via the controller in real-time. The user interface can now present logs, charts, graphs or agent policies in real-time as events stream in from the worker. This offers an substantive improvement over the classical turn-based execution of experiments where results are made available at termination of a run. The next iteration of our HwtL approach will allow for direct manipulation of running experiments.

### 4.3 Gym Composer

Our system greatly expands on the OpenAI *gym* interface and provides a mechanism to create gyms through a composer. The composer provides several abstractions to ease creation of the gym. Specifically, definitions of observation and action space, along with reward function primitives allow the user to focus on specifying the main logic of state transitions. The system, as shown in Figure 6, aids in composing gyms using industry-specific templates for a broad range of sectors. An end-user still has several choices to make on how best to model the environment. The first choice users face is on deciding state variables, a minimal set of quantities that describe the system dynamics. Next the user is guided through the definition of a transition function, the reward function which can



**Figure 5: (Remote) Execution in AutoDOViz:** The sequence diagram outlines the interplay between users, API controller and compute cluster workers to track RL agents, allowing the user interface to present logs, graphs and report on agent progress in real-time.



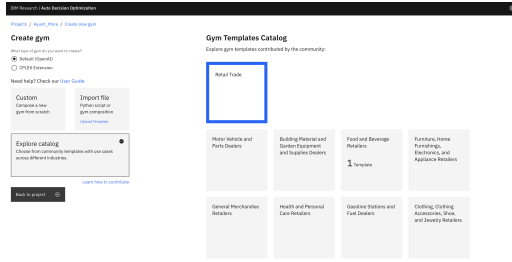
**Figure 6: Gym creation in AutoDOViz.** Users can provide OpenAI gyms in three alternative ways: (A) Compose gym via Guiding Gym Composer. (B) Directly upload gym from Python file. (C) Choose gym from Gym Template Catalog.

be comprised of a linear combination of individual metrics, as well as a termination criteria. Once the user is satisfied with their inputs to the gym composer, our API controller can auto-generate the final Python implementation script from these, which in turn serves as input to the AutoDO engine.

### 4.4 Gym Template Catalog

With respect to **R1**, **R8**, **R9** we propose a gym template catalog. While a type-driven categorization according to **R1** is already provided by the DO researchers, we parsed options for a more domain-driven hierarchy towards **R8**. Our research on methods for classifying different kinds of business problems led us to the North American Industry Classification System (NAICS) as a robust and continually-improving breakdown of different industries. It was created jointly between the governments of The United States, Canada, and Mexico, and is updated every five years to capture emerging industries and technologies. The NAICS summarizes industries into





**Figure 7: Gym Template Catalog in AutoDOViz. Gym templates can be explored optimization-type-based or industry-based (according to the NAICS standard).**

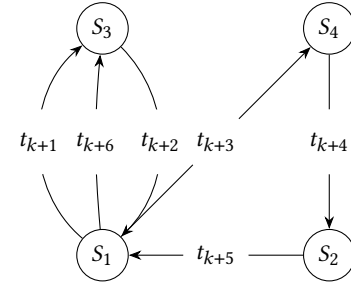
20 sectors, and hierarchically expands into 1,012 unique industries.<sup>2</sup> Figure 7 shows the design of our template catalog in AutoDOViz. Users can click through the catalog hierarchy via tiles on each level, with the parent node pinned to the top with a blue border. When arriving on a leaf-level, the catalog shows the list of available templates. Upon loading a gym template from the catalog the gym composer (ref. section 4.4), the composer is prefilled with the corresponding inputs so the user can study the environment and make customized changes according to their exact optimization problem.

#### 4.5 State-Transition Visualizations

In order to promote trust and confidence into agents **R6** as well as to add to more generic visualization techniques **R2** we propose two types of visualizations as part of AutoDOViz. Consider a typical evaluation protocol as provided in table 2, consisting of chosen action, state of the environment and reward at any step in the process. Figure 8 then depicts a graphical representation of the evaluation protocol, with states as vertices of the graph and the trajectory of the agent shown as edges. The time step of transition is shown as labels on the edges.

**Table 2: Typical RL agent evaluation protocol.**

Step	Action	State	Reward	$\delta_{\text{Reward}}$
...	...	...	...	...
$t_k$	$A_1$	$S_1$	72	+1
$t_{k+1}$	$A_2$	$S_3$	75	+3
$t_{k+2}$	$A_3$	$S_1$	74	-1
$t_{k+3}$	$A_1$	$S_4$	78	+4
$t_{k+4}$	$A_3$	$S_2$	81	+3
$t_{k+5}$	$A_2$	$S_1$	80	-1
$t_{k+6}$	$A_3$	$S_3$	82	+2
...	...	...	...	...



**Figure 8: Temporal transition graph.**

*Transition Matrix.* Since edge labels become impractical with larger evaluation protocols, we propose a level of abstraction as shown in table 3. Here, we infer an adjacency matrix from the evaluation protocol and count the number of the times the agent moves between any two states. A similar abstraction can be made for the action space as shown in table 4. While not preserving the information on sequence of steps in such transition matrices, it allows for a very compact representation of the transition pattern, which can favor comparison of many RL agents and large protocols.

**Table 3: State transition matrix.**

	$S_1$	$S_2$	$S_3$	$S_4$
$S_1$			2	1
$S_2$	1			
$S_3$	1			
$S_4$		1		

**Table 4: Action transition matrix.**

	$A_1$	$A_2$	$A_3$
$A_1$		1	1
$A_2$			2
$A_3$	1	1	

*Transition Graph.* From the abstracted adjacency matrices we can render graphs, as shown in figures 9 and 10 respectively. Here, the number of times the agent transitioned between any two states is encoded via the thickness or opacity of the edge. Note that compared to prior work, where projection of the state space happens solely based on the feature space of the environment, the proposed graph-based approach adjusts proximity of states taking into account the transitional behavior of the agent.

<sup>2</sup><https://www.census.gov/naics/>

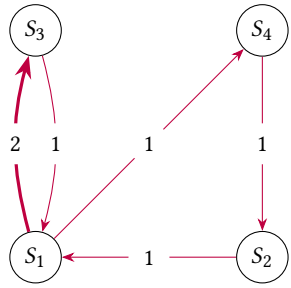


Figure 9: State transition graph

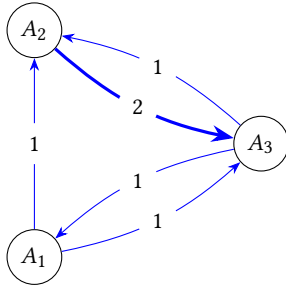


Figure 10: Action transition graph.

**Scalability.** In AutoDOViz, we implemented both the matrix and graph-based technique to present behavioral information about the agent. With real-world evaluation protocols having larger action or state spaces and going into the thousands of steps per episode, our visualizations need to be able to scale well. With respect to transition matrices, where the scalability bottleneck is the dimensionality of the matrix, the user can cluster states prior to abstraction of the matrix. This scales since the matrix will then be limited to a user-defined dimensionality  $k$  the number of clusters. Furthermore, states can be filtered based on certain criteria even before going into the clustering algorithm. Figures 11 and 12 depict a 10-means clustering of states before and after training. Initial random behavior of the agent is replaced by more selective transitions during training, as to avoid less rewarding states of the environment.

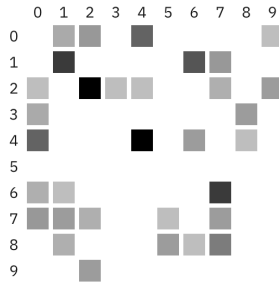


Figure 11: Clustered state transitions of early training episode.

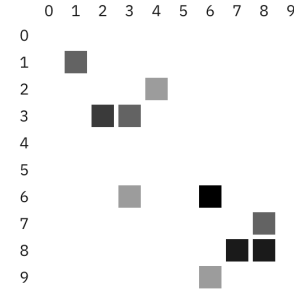


Figure 12: Clustered state transitions of late training episode.

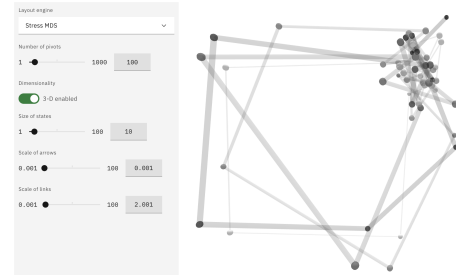


Figure 13: Transition graph: States are projected with multi-dimensional scaling before augmenting the tour of the RL agent.

While the transition graph could in principle cope with thousands of states directly in the visualization (even without prior clustering) the bottleneck is the time to compute the projection of the states, i.e. the exact  $X,Y$  or  $X,Y,Z$  coordinates of the state nodes. For AutoDOViz, we therefore implemented state-of-the-art graph layout algorithms [6, 45, 62] in JavaScript to speedily compute layouts in the browser in 2D or 3D. Figure 13 shows an example where states of an evaluation protocol are projected with stress-based dimensionality reduction. We then depict the tour of the agent as it moves from state to state. Here, the thickness of an edge encodes the time of the step, so that later transitions can again be distinguished from earlier transitions.

#### 4.6 Policy Explanations

Once end-users have selected one or more agents of interest, they may inspect the agent's behavior using a surrogate rule set. The surrogate rules are learnt using observed state-action pairs of the policy behaviour in order to classify the action taken by the agent based on the state [37]. The resulting surrogate is a set of boolean logic rules that attempt to encompass and translate an agent's behavior in the most compact way possible. Rules are hierarchical, the first covers the largest population in the data, then any unexplained data points are covered by the second rule, followed by the third etc. In order to approximate an agent's behavior in the form of rules, users may choose a set of evaluations over a predefined interval of episodes, ranging from 0 to 20. The rule extraction method performs more accurately with more inputs but early episodes may not be as representative of the final agent as the final episodes. Once this

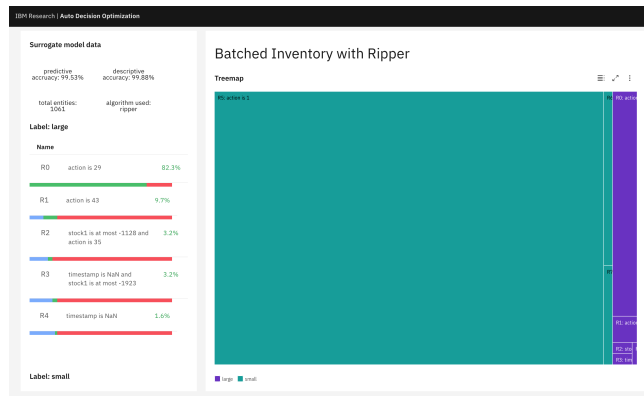


Figure 14: Extracted surrogate rule set

interval is selected, all evaluations are concatenated into one, then users are required to select and batch a desired column into buckets as rules require a categorical prediction label. Once this batching process is completed, the selected column is discarded and rules are generated for each available prediction label. These are displayed as a list of boolean rules with their respective coverage chart as well as a global coverage tree map, displaying the weight of each rule. Boolean rules enable end-users to quickly understand an agent's behavior and compare it to other similar agents, providing better comparative insights than simple accuracy scores.

## 5 EVALUATION

### 5.1 User Study with Data Scientists

Building on results of prior expert interviews potential target end-users for AutoDOViz were identified to be data scientists. To validate this assumption and the platform's development with regard to design requirements a user study was conducted with 13 data scientists<sup>3</sup> employed internally in the company from across 8 different countries, with varying levels of ML experience. Participants were recruited both from pools of self-selecting volunteers via an open call on the company's internal messaging software, and others whom were contacted directly. The participants were not additionally compensated for their time. Participants were given no prior knowledge of study content apart from that it was to test a novel UI aimed towards data scientists. The aim of the study was to broadly evaluate whether the system successfully affords the user the ability to employ AutoDOViz to create and manage gyms and configure agents, as set out by the design requirements in section 3.3 and design choices made in section 4. We also aimed to evaluate the UI to identify any usability concerns and to solicit feedback about potential improvements to the tool. The study took the form of a 45 minute online session for each participant comprising a set of tasks and a semi-structured interview, which was recorded for later analysis. Detail about the participants is given in Table 5. Each participant was guided by an interviewer to complete a pre-study

<sup>3</sup>University collaborators were not directly involved in conduction of the user study with human subjects so this study is not subject to federal IRB oversight. IBM Business Conduct Guidelines have been followed at all times <https://www.ibm.com/investor/governance/business-conduct-guidelines>.

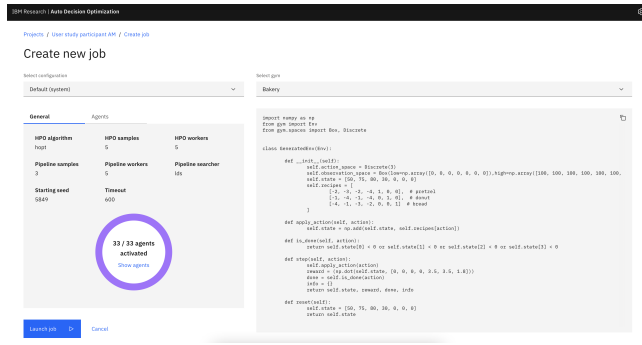
questionnaire, complete a list of tasks using an online environment and then complete a post-study questionnaire to reflect upon the experience.

**Pre-study Questionnaire.** This questionnaire contained 14 questions covering demographic information to learn more about the participants as well as questions to determine their level of prior knowledge and trust in Auto ML, reinforcement learning, and decision optimization.

**Tasks.** Participants were encouraged to "think aloud" as they followed through a set of tasks. In order to account for differing knowledge levels, an explanation of the concept of OpenAI gyms was given to some of the participants. When incorrect basic assumptions were made, help was provided to allow them to proceed with the tasks if necessary. Participants were asked to answer questions designed to check their ability to gather information from different parts of the UI and to assess their understanding of the scenario. After the interviews, participants' task responses were categorized as correct/successful or incorrect/fail by the interviewers, taking into account the level of additional help required to get through the tasks.

As earlier interviews with DOs highlighted the importance of a thorough understanding of the business problem and processes rather than simply understanding through the lens of data, scenarios were prepared using real-world examples to allow assessment of both how well the UI assisted in data scientists' understanding of the business problem, as well as their success in creating, editing and configuring OpenAI gyms and jobs using the composer interface. In the first scenario, participants were asked to use the online environment and a configuration file to create and modify an OpenAI gym representing a bakery (ref. figure 15), which included details of certain recipes and how these affected inventory levels. Participants were provided with a link to an online tool and a pre-configured JSON file containing details of the bakery to use to complete the task. The questions were designed to guide them through the process of creating a gym and creating and modifying jobs and configurations, and assess their understanding of the optimization problems posed in the bakery. Participants were asked to use the catalog to load another scenario based on a produce arrangement problem of the chemical reactions between different fruits (described in section 3.2) and asked to create an additional metric which determines the distance between specified fruits.

**Post-study reflection.** The post-study questionnaire took the form of 14 questions, including 11 5-point Likert agreement scale questions attempting to assess trust in the system, ease of use and usefulness of the UI and visualizations, attitude to the catalog and any change in their self-reported level of understanding of reinforcement learning. Free-text sections were provided for the participants to record what they liked and disliked about the tool. These were used as a basis for informal discussions with the participants to further determine their attitudes to the tool and how it might fit into their current workflows.



**Figure 15: Job preview screen in AutoDOViz: Execution of a job to evaluate agents on *Bakery*-gym during user study.**

## 5.2 Results

In the following subsections we report results from our user study leading up to a discussion of our findings and limitations in section 6.

**5.2.1 Pre-study questionnaire.** Table 5 shows the demographic data on user study participants. While gender is balanced (A1), age distribution leans towards younger professionals (A2). Yet, 23.1 % of participants were above the age of 51, with 1 participant above the age of 61. All participants had a university degree, with the majority of participants having advanced degrees (A3), also in fields *outside* of computer science, data science or artificial intelligence (A4). All users had prior experience in machine learning or data science, with the majority between 2 and 10 years, and 1 outlier with up to more than 30 years of experience in machine learning (A5). Many (46.2 %) users stated that they had no prior experience with decision optimization (A6).

In self-assessment, 84.61 % of users reported to have worked or frequently work with automated machine learning (A7). Furthermore, 61.5 % of participants state to have worked with RL before (A10), however, only 30.8 % claimed skills beyond beginner level (A8). All participants with prior RL experience have seen or directly experimented with OpenAI Gym implementations before. Less than half of the users seemed interested in applying AutoDO in real-world problems with clients (A9). However, 69.2 % reported they would trust automated decision optimization (A12). Regarding the execution environment for RL agent optimization workers, only 1 user felt comfortable to use automated decision optimization in a shared environment, 38.5 % preferring to exclusively work in custom environments, and 53.9 % with the desire for both (A13).

**5.2.2 Tasks.** Task B1 was correctly executed by all but 2 users, who were unable to find the button to open their project after creation. Loading from file (B2) or catalog (B3) was mostly successful, however, with few templates in our catalog, 3 participants were confused as to why the template counter in the category tiles did not change and needed little guidance drilling down the hierarchy. Reading simple information about the gym (B4-6) was easy enough for most users, however 3 users gave the right answer based on reading from the wrong location: when asked about the length of the state vector they instead read the length from the observation

space. The advanced task of adding a custom reward metric to a gym was successfully completed by 84.6 % of the users. All users correctly explained the transition function, with 8 (5) people giving a very precise, problem-specific (more general) explanation. Further, 9 (5) participants provided precise, problem-specific (more general) explanations on the termination criteria, with only 1 person failing to provide a meaningful description in the given scenario. Regarding tasks to modify the configuration of the AutoDO core engine (B10-12), almost all users correctly solved all tasks, with only 1 person changing the number of *pipeline* workers rather than the number of *optimization* workers. Tasks concerned with the interpretation and launch of automation jobs (B13-17) were successfully executed by all participants. A single user misread the number of active agents despite looking at the right information in the tool. In B18-19 users were presented transition matrix and transition graph visualizations and asked to answer a total of 6 questions associated with the interpretation of the visualizations: number of clusters, behavioral changes before and after training, tour of the agent. All participants gave the correct answers, with 2 different participants answering only partially correctly in some of the 6 questions. Overall, 93.1 % of the tasks were executed correctly.

**5.2.3 Post-study questionnaire.** Table 8 shows results of the post-study questionnaire with a Likert scale from 1 (strongly disagree) to 5 (strongly agree). 92.3 % of the participants found the gym composer to be helpful or very helpful (C1). 84.6 % strongly agreed that the visualizations provided useful insights on RL agents (C2). All users found it very easy to launch jobs once a gym was created (C3). 11 of 13 participants strongly agreed that knowing the behavior of individual metrics of reward functions made them more comfortable using an agent in production (C4). While not all participants felt a strong desire to *contribute* to the gym template catalog (C5), 92.3 % could see themselves consulting the template catalog before coding a gym themselves (C6). 10 participants expressed strong preference to be able to fine-tune the AutoDO engine with configurations (C7). 69.2 % agreed to trust or strongly trust AutoDO, with only 4 participants being neutral. No participant reported distrust (C8). 9 participants expressed strong or advanced understanding of RL after our study, 2 were neutral and only 2 participants reported less than neutral understanding of RL (C9). 92.3 % of users would like to use AutoDO in their own projects with only 1 user being neutral (C10). Lastly, 3 participants did not necessarily prefer to execute AutoDO jobs in a custom environment rather than a shared one, 5 participants were neutral, 3 would prefer to and 2 users would strongly prefer to execute AutoDO in their custom environment/private cluster (C11).

**5.2.4 Post-survey likes, dislikes and reflection.** Participants made comments in the post-survey questionnaire relating to specific UX improvements (5 comments), and need for additional on-screen explanation for less familiar users (2 comments). Areas which they felt could be improved included user experience for small screen-size users, to reduce the amount of scrolling necessary in the composer screen while trying to remember relevant information. The agent listing screen was also something users felt could be refined. One user felt that interpretability of visualizations should be improved. Three participants reported that there was nothing in particular they disliked about the tool. Other suggestions included time sliders

**Table 5: Participant demographic data gathered from pre-study questionnaire**

<b>A1</b>	Gender	<b>Female</b> 6 (46.15 %)	<b>Male</b> 7 (53.85 %)			
<b>A2</b>	Age	<b>20 - 30</b> 8 (61.54 %)	<b>31 - 40</b> 2 (15.39 %)	<b>41 - 50</b> 0 (0 %)	<b>51 - 60</b> 2 (15.38 %)	<b>61 - 70</b> 1 (7.69 %)
<b>A3</b>	Degree	<b>Bachelor</b> 4 (30.77 %)	<b>Master</b> 6 (46.15 %)	<b>PhD</b> 3 (23.08 %)		
<b>A4</b>	Field of study	Data Science & Artificial Intelligence (3), Computer Science (3), Physics (2), Design Engineering (1), Electrical Engineering (1), Geospatial Science (1), Computer Applications (1), Tourism & Communication (1)				
<b>A5</b>	Experience with ML/DS (years)	<b>0-1</b> 1 (7.69 %)	<b>2-4</b> 6 (46.15 %)	<b>5-10</b> 3 (23.08 %)	<b>11-30</b> 2 (15.38 %)	<b>30+</b> 1 (7.69 %)
<b>A6</b>	Experience with DO (years)	<b>0</b> 6 (46.15 %)	<b>1</b> 4 (30.77 %)	<b>2-3</b> 3 (23.08 %)		

**Table 6: Results pre-study questionnaire**

	(Scale 1 to 5)	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>A7</b>	Experience AutoML	1 (7.69 %)	1 (7.69 %)	5 (38.47 %)	2 (15.38 %)	4 (30.77 %)
<b>A8</b>	Experience RL	1 (7.69 %)	3 (23.08 %)	5 (38.47 %)	4 (30.77 %)	0 (0 %)
<b>A9</b>	Interest in AutoDO	1 (7.69 %)	1 (7.69 %)	5 (38.47 %)	2 (15.38 %)	4 (30.77 %)
		<b>Yes</b>	<b>No</b>			
<b>A10</b>	Previously worked with RL	8	5			
<b>A11</b>	Previously used Simplex algorithm	3 (23.08 %)	10 (76.92 %)			
<b>A12</b>	Trust in AutoDO	9 (69.23 %)	4 (30.77 %)			
<b>A13</b>	Preferred execution environment.	<b>Shared</b>	<b>Custom</b>	<b>Both</b>		
		1 (7.69 %)	5 (38.47 %)	7 (53.85 %)		

to replay real-time feedback on the visualizations of the agents' progress. The participants reported things they liked which can be summarized as the overall experience was reported as intuitive (5 comments), easy to work with (6 comments), customizable (4 comments) and transparent (1 comment). Participants called out the visualizations as an element that they found useful (4 comments). It was appreciated that the code and the builder were presented side-by-side and that the UI was generally fun to use. The interviewer discussed these reflections further with the participants to ensure that their feedback was being interpreted as intended. In general, the written feedback comments suggest that the tool is perceived as both useful and highly usable.

## 6 DISCUSSION

### 6.1 Trust, control & transparency

A surprisingly high level of trust in automated decision-making was already reported in the pre-study questionnaire (A12), with 69.23% of participants in agreement that they had confidence in automated DO and 30.77% against it. Several participants expressed reservations about the trust question as they wished to qualify that it could be circumstance-dependent. Still, it could be that as people involved daily in ML, they had some idea of the potential problems and rewards. *"Perfectly comfortable with it. I know the pros and cons.*

*I know what can go wrong. Uh, yeah, so that's the qualified yes."* After the study, we can see that the same amount of participants remains positive, but the remainder has moved to a neutral position (R6). *"I mean, I couldn't even see what happens in the back end, but it does seem to inspire trust"*.

Participants further appreciated that the UI allowed them to customize gyms and job configuration quickly; this appeared to positively affect perception of control of the tool and its output (R5). Although they have a level of trust in AI, they wish to remain able to verify and control the system's output. *"Definitely having that control to a certain extent... that's definitely preferable."* *"Yeah, I mean, ... especially if you're an expert user... you know how machine learning works, what kind of things you want to leave to the machine, and which type of things you want to control yourself."*

Transparency was an aspect participants felt was imperative, particularly on metrics. *"knowing how it's measured, that makes one more comfortable."* The majority appeared to scroll through the generated gym code output to verify and understand. *"Just having insights into what is doing, maybe what are some of the trade-off that it is making... to get a little bit more of that,..."*

### 6.2 Accessibility for novice and expert users

The initial interviews suggested that the tool should support users of multiple optimization problems (R4). The tool aimed to balance

**Table 7: Results tasks**

		Correct/success	Incorrect/fail
<b>Project</b>			
<b>B1</b>	Create	11	2
<b>Gym</b>			
<b>B2</b>	Load from file	12	1
<b>B3</b>	Load from catalog	10	3
<b>B4</b>	Read size of action space	13	0
<b>B5</b>	Read length of state vector	10	3
<b>B6</b>	Read reward	13	0
<b>B7</b>	Add reward metric	11	2
<b>B8</b>	Explain transition function	8 + 5	0
<b>B9</b>	Explain termination criteria	9 + 3	1
<b>Configuration</b>			
<b>B10</b>	Change number of workers	12	1
<b>B11</b>	Enable/disable agents	13	0
<b>B12</b>	Change agent hyperparameter	13	0
<b>Job</b>			
<b>B13</b>	Read number of workers	13	0
<b>B14</b>	Read number of agents	12	1
<b>B15</b>	List 3 agents	13	0
<b>B16</b>	Read agent hyperparameter	13	0
<b>B17</b>	Launch	13	0
<b>Visualization</b>			
<b>B18</b>	Interpret matrix	12	1
<b>B19</b>	Interpret graph	12	1
<b>TOTAL</b>		<b>230 (93.1 %)</b>	<b>17 (6.9 %)</b>

**Table 8: Results post-study questionnaire**

		1	2	3	4	5
<b>C1</b>	Gym Composer	0	0	1	6	6
<b>C2</b>	Visualizations	0	0	2	5	6
<b>C3</b>	Launching jobs	0	0	0	0	13
<b>C4</b>	Reward metrics	0	0	0	2	11
<b>C5</b>	Catalog contrib.	0	0	4	3	6
<b>C6</b>	Catalog consult.	0	0	1	0	12
<b>C7</b>	Engine config	0	0	1	2	10
<b>C8</b>	Trust AutoDO	0	0	4	5	4
<b>C9</b>	Understand RL	0	2	2	3	6
<b>C10</b>	Apply AutoDO	0	0	1	2	10
<b>C11</b>	Custom env.	0	3	5	3	2

wizard-style tasks for entry-level and direct code access to gyms for more experienced users. This appears to have been successful in that both low-code and code-first configurations are supported, and the participants recognized this. Some advocated for low-code solutions with even more adaptations for novice users, such as UI labels for the state. *“Well, compared to code, it makes it much more accessible, that’s for sure. If that’s the alternative, I’ll take this tool any time of the day for these kinds of things. So that’s, I think, a big plus. It makes the code a little bit less daunting and makes it easier to use”* Others were more comfortable having a flexible approach but thought that the UI would increase their efficiency and

save time when creating a gym. *“I like the flexibility of having code there.”* Overall, the feedback would suggest that the UI successfully allows data scientists to learn to automate DO tasks with RL quickly. *“Kudos to you guys! I liked it as a beginner and someone who’s not used these things in my day-to-day project. I think it was it was very user-friendly.”*

### 6.3 Leveraging the UI as an educational tool

We observe that participants’ previous level of RL knowledge did not correlate directly to their success in completing the required tasks, or the fluency in which they could navigate the interface. The tool and the experience of completing the tasks in the user study attempt to make the complex area of RL more palatable to the novice user, as directed by requirement R7. This is supported by the increase in self-reported understanding of RL (C9) and comments from participants. *“having had almost no experience with this, it was intuitive enough for me to follow and, you know, just get a sense of what was happening.”* (R7)

### 6.4 Collaborative potential & resource sharing

With respect to the gym template catalog (R1, R8, R9), participants seemed well-disposed towards finding and reusing code samples from the catalog for different domains. *“As a new user, you probably want to have a look for some examples before you go in and try on yourself”*. However, they were less sure that they would contribute



to the catalog due to concerns about client confidentiality for their work.

One user in particular seemed eager to collaborate in AutoDOViz with their clients and external stakeholders (R3). *“With this tool, it’s something that I would consider playing around with, coming up with use cases, you know, doing some testing. That’s definitely something. Even opening this conversation with clients... and thinking of use cases in the industry could be beneficial. But definitely, this is this could be a conversation starter for sure.”*

Several participants asked when AutoDOViz would be integrated into their existing toolkits for them to start to experiment with it for clients. *“I’m a data scientist within client engineering, so we work on... 6 to 8-week intervals with their clients and create a product within that, an MVP or proof of concept. So definitely, something like this, I can see, is very helpful, rather than me focusing on, like, the little itty-bitty like the parameters. I think you should have a Kaggle mindset to see how you can make the best model, but to have something that works and the client’s happy with it. I think this is a good option.”*

When questioned about attitudes to working in a shared versus a custom environment (C11), it emerged that this is highly use-case dependent (R1, R8), on customer need driven by security, privacy, and cost factors. Participants working in pre-sales on customer demos would have a different perspective than those working on government projects.

## 6.5 Design decisions & mental model

The UI was designed and developed using a UI framework that was familiar to the data scientists from other software that they used daily (R2). This consistency was appreciated by some of the participants and they suggested that it helped them navigate through the tasks more smoothly. *“Right and the placement also, right? I mean, yeah, it helps to... it’s sort of like muscle memory that okay I know where to find.”*

*“As a whole, it kind of fits my mental model. I think of where you would click and where you would find things. So, that fits well as well...it’s kind of logical that way.”*

## 6.6 Limitations

Although the population was diverse in age and experience level, all participants were selected from the same company, which might yield familiarity with similar tools. This could potentially affect the generalization of results if the study was replicated with external participants (external validity). Overall, we could criticize differing ML/RL knowledge levels amongst participants, such as using a task script augmented by verbal instructions and discussion, which differed slightly for each participant. Also, the participants are subjectively self-reporting their skill level (internal validity). The tasks and examples chosen could be simpler than those encountered in real-world scenarios. However, we needed to balance ease of access for a beginner and an experienced user (construct validity). Another limitation could be the question of how to measure trust in AutoDOViz. Yes-No questions may not provide the nuance that participants expressed verbally since there were a few “Yes, but...” responses.

## 7 CONCLUSION AND FUTURE WORK

We presented AutoDOViz, an interactive user experience for automated DO with RL. Findings from semi-structured expert interviews with DO practitioners and business consultants generated design requirements for human-centered automation for DO. We implemented a system design and evaluated our software in a rigorous user study with data scientists. We found that they are significantly more open to engaging in DO after using our proposed solution. AutoDOViz promotes trust and confidence in RL agent models via reward metrics and makes the automated training and evaluation process more accessible and understandable. This is further expanded by leveraging the benefits of AutoDO algorithms for RL pipeline search to generate policy insights and advanced visualizations to enable communication between DO experts and domain experts. The novel gym composer and proposed gym template catalog lower the barrier of entry for data scientists in problem specification for RL problems. The provided streaming architecture enables Human-within-the-Loop for enhanced interaction, with real-time *manipulation* subject to future work. The user study further gave rise to helpful design improvements in the overall user interface. Future work will be conducted in the direction of real-world applications for our proposed system to test and demonstrate the interface’s efficiency amongst various personas. In this regard, we would also like to iterate on the application to become role-based and add collaborative features directly to the interface. Lastly, more problem-specific visualizations will make the agents’ performance more interpretable and explainable.

## REFERENCES

- [1] Dan Amir and Ofra Amir. 2018. Highlights: Summarizing agent behavior to people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 1168–1176.
- [2] M Emin Aydin and Ercan Öztemel. 2000. Dynamic job-shop scheduling using reinforcement learning agents. *Robotics and Autonomous Systems* 33, 2-3 (2000), 169–178.
- [3] Maria Barbati, Giuseppe Bruno, and Andrea Genovese. 2012. Applications of agent-based models for optimization problems: A literature review. *Expert Systems with Applications* 39, 5 (2012), 6020–6028.
- [4] Andrew G Barto, Satinder Singh, Nittapong Chentanez, et al. 2004. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*. Piscataway, NJ, 112–119.
- [5] Ekaba Bisong. 2019. Google AutoML: cloud vision. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Springer, 581–598.
- [6] Ulrik Brandes and Christian Pich. 2006. Eigensolver methods for progressive multidimensional scaling of large data. In *International Symposium on Graph Drawing*. Springer, 42–53.
- [7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
- [8] Angelos Chatzimpampas, Rafael Messias Martins, Ilir Jusufi, Kostiantyn Kucher, Fabrice Rossi, and Andreas Kerren. 2020. The state of the art in enhancing trust in machine learning models with the use of visualizations. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 713–756.
- [9] Po-Wen Chiu and Christina Bloebaum. 2008. Hyper-Radial Visualization (HRV) for decision-making in multi-objective optimization. In *46th AIAA Aerospace Sciences Meeting and Exhibit*. 907.
- [10] Piali Das, Nikita Ivkin, Tanya Bansal, Laurence Rouesnel, Philip Gautier, Zohar Karnin, Leo Dirac, Lakshmi Ramakrishnan, Andre Perunicic, Iaroslav Shcherbatyi, et al. 2020. Amazon SageMaker Autopilot: a white box AutoML solution at scale. In *Proceedings of the fourth international workshop on data management for end-to-end machine learning*. 1–7.
- [11] Pieter Desmet and Paul Hekkert. 2007. Framework of product experience. *International journal of design* 1, 1 (2007), 57–66.
- [12] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).



- [13] Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. *Commun. ACM* 63, 1 (2019), 68–77.
- [14] Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. 2003. The role of trust in automation reliance. *International journal of human-computer studies* 58, 6 (2003), 697–718.
- [15] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and robust automated machine learning. *Advances in neural information processing systems* 28 (2015).
- [16] Jasmina Gajcin, Rahul Nair, Tejaswini Pedapati, Radu Marinescu, Elizabeth Daly, and Ivana Dusparic. 2022. Contrastive Explanations for Comparing Preferences of Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*.
- [17] Claudio Gambella, Bissan Ghaddar, and Joe Naoum-Sawaya. 2021. Optimization problems for machine learning: A survey. *European Journal of Operational Research* 290, 3 (2021), 807–828.
- [18] Yolanda Gil, James Honaker, Shikhar Gupta, Yibo Ma, Vito D’Orazio, Daniel Garijo, Shruti Gadewar, Qifan Yang, and Neda Jahanshad. 2019. Towards human-guided machine learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 614–624.
- [19] Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. 2018. Visualizing and understanding atari agents. In *International conference on machine learning*. PMLR, 1792–1801.
- [20] Marc Hassenzahl and Noam Tractinsky. 2006. User experience—a research agenda. *Behaviour & information technology* 25, 2 (2006), 91–97.
- [21] Wenbin He, Teng-Yok Lee, Jeroen van Baar, Kent Wittenburg, and Han-Wei Shen. 2020. DynamicsExplorer: Visual analytics for robot control tasks involving dynamics and LSTM-based control policies. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 36–45.
- [22] Martin Hirzel, Kiran Kate, Parikshit Ram, Avraham Shinnar, and Jason Tsay. 2022. Gradual AutoML using Lale. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4794–4795.
- [23] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. 2019. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Transactions on Visualization and Computer Graphics* 25, 8 (2019), 2674–2693. <https://doi.org/10.1109/TVCG.2018.2843369>
- [24] Theo Jaunet, Romain Vuilleminot, and Christian Wolf. 2020. DRLViz: Understanding decisions and memory in deep reinforcement learning. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 49–61.
- [25] Haifeng Jin, Qingquan Song, and Xia Hu. 2019. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1946–1956.
- [26] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research* 4 (1996), 237–285.
- [27] James Max Kanter and Kalyan Veeramachaneni. 2015. Deep feature synthesis: Towards automating data science endeavors. In *2015 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE, 1–10.
- [28] Udayan Khurana, Deepak Turaga, Horst Samulowitz, and Srinivasan Parthasarathy. 2016. Cognito: Automated feature engineering for supervised learning. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 1304–1307.
- [29] A. Kishimoto, D. Bouneffouf, R. Marinescu, P. Ram, A. Rawat, M. Wistuba, P. Palmes, and A Botea. 2022. Bandit Limited Discrepancy Search and Application to Machine Learning Pipeline Optimization. In *AAAI Conference on Artificial Intelligence (AAAI)*. 10228–10237.
- [30] Hoang Thanh Lam, Johann-Michael Thiebaud, Mathieu Sinn, Bei Chen, Tiep Mai, and Ozgur Alkan. 2017. One button machine for automating feature engineering in relational databases. *arXiv preprint arXiv:1706.00327* (2017).
- [31] Effie Lai-Chong Law, Arnold POS Vermeeren, Marc Hassenzahl, and Mark Blythe. 2007. Towards a UX manifesto. In *Proceedings of HCI 2007 The 21st British HCI Group Annual Conference University of Lancaster, UK* 21, 1–2.
- [32] Erin LeDell and Sebastien Poirier. 2020. H2o automl: Scalable automatic machine learning. In *Proceedings of the AutoML Workshop at ICML*, Vol. 2020.
- [33] Doris Jung-Lin Lee and Stephen Macke. 2020. A Human-in-the-loop Perspective on AutoML: Milestones and the Road Ahead. *IEEE Data Engineering Bulletin* (2020).
- [34] Yuxi Li. 2017. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274* (2017).
- [35] Michele Lombardi, Michela Milano, and Andrea Bartolini. 2017. Empirical decision model learning. *Artificial Intelligence* 244 (2017), 343–367.
- [36] Radu Marinescu, Tejaswini Pedapati, Long Vu, Paulito Palmes, Todd Mummert, Peter Kirchner, Dharmashankar Subramanian, Parikshit Ram, and Djallel Bounieffouf. 2022. Automated Decision Optimization with Reinforcement Learning. (2022).
- [37] James McCarthy, Rahul Nair, Elizabeth Daly, Radu Marinescu, and Ivana Dusparic. 2022. Boolean Decision Rules for Reinforcement Learning Policy Summarisation. *International Workshop on Safe Reinforcement Learning, IJCAI* (2022).
- [38] David Meignan, Sigrid Knust, Jean-Marc Frayret, Gilles Pesant, and Nicolas Gaud. 2015. A review and taxonomy of interactive optimization methods in operations research. *ACM Transactions on Interactive Intelligent Systems (TiIS)* 5, 3 (2015), 1–43.
- [39] Aditi Mishra, Utkarsh Soni, Jinbin Huang, and Chris Bryan. 2022. Why? Why not? When? Visual Explanations of Agent Behaviour in Reinforcement Learning. In *2022 IEEE 15th Pacific Visualization Symposium (PacificVis)*. IEEE, 111–120.
- [40] 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), 529–533.
- [41] Sina Mohseni, Niloofar Zarei, and Eric D Ragan. 2021. A multidisciplinary survey and framework for design and evaluation of explainable AI systems. *ACM Transactions on Interactive Intelligent Systems (TiIS)* 11, 3–4 (2021), 1–45.
- [42] Deepak Mukunthu, Parashar Shah, and Wee Hyong Tok. 2019. *Practical Automated Machine Learning on Azure: Using Azure Machine Learning to Quickly Build AI Solutions*. O’Reilly Media.
- [43] John A Nelder and Roger Mead. 1965. A simplex method for function minimization. *The computer journal* 7, 4 (1965), 308–313.
- [44] Randal S Olson, Nathan Bartley, Ryan J Urbanowicz, and Jason H Moore. 2016. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the genetic and evolutionary computation conference 2016*. 485–492.
- [45] Mark Ortmann, Mirza Klimenta, and Ulrik Brandes. 2016. A sparse stress model. In *International Symposium on Graph Drawing and Network Visualization*. Springer, 18–32.
- [46] Erika Puiutta and Eric Veith. 2020. Explainable reinforcement learning: A survey. In *International cross-domain conference for machine learning and knowledge extraction*. Springer, 77–95.
- [47] Emily Saldanha, Brenda Praggastis, Todd Billow, and Dustin Lockhart Arendt. 2019. ReLVis: Visual Analytics for Situational Awareness During Reinforcement Learning Experimentation.. In *EuroVis (Short Papers)*. 43–47.
- [48] Pedro Sequeira, Eric Yeh, and Melinda T Gervasio. 2019. Interestingness Elements for Explainable Reinforcement Learning through Introspection.. In *IUI workshops*, Vol. 1.
- [49] Syed Yousaf Shah, Dhaval Patel, Long Vu, Xuan-Hong Dang, Bei Chen, Peter Kirchner, Horst Samulowitz, David Wood, Gregory Bramble, Wesley M Gifford, et al. 2021. AutoAI-TS: AutoAI for time series forecasting. In *Proceedings of the 2021 International Conference on Management of Data*. 2584–2596.
- [50] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- [51] Alfredo Vellido, José David Martín-Guerrero, and Paulo JG Lisboa. 2012. Making machine learning models interpretable. In *ESANN*, Vol. 12. Citeseer, 163–172.
- [52] Dakuo Wang, Justin D Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI collaboration in data science: Exploring data scientists’ perceptions of automated AI. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–24.
- [53] Junpeng Wang, Liang Gou, Han-Wei Shen, and Hao Yang. 2018. Dqnviz: A visual analytics approach to understand deep q-networks. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 288–298.
- [54] Junpeng Wang, Wei Zhang, Hao Yang, Chin-Chia Michael Yeh, and Liang Wang. 2021. Visual analytics for rnn-based deep reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics* (2021).
- [55] Qianwen Wang, Zhutian Chen, Yong Wang, and Huamin Qu. 2021. A Survey on ML4VIS: Applying Machine Learning Advances to Data Visualization. *IEEE Transactions on Visualization and Computer Graphics* (2021).
- [56] Qianwen Wang, Yao Ming, Zhihua Jin, Qiaomu Shen, Dongyu Liu, Micah J Smith, Kalyan Veeramachaneni, and Huamin Qu. 2019. Atmsee: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [57] Tianshu Wei, Yanzhi Wang, and Qi Zhu. 2017. Deep reinforcement learning for building HVAC control. In *Proceedings of the 54th annual design automation conference 2017*. 1–6.
- [58] Daniel Karl I Weidele. 2019. Conditional parallel coordinates. In *2019 IEEE Visualization Conference (VIS)*. IEEE, 221–225.
- [59] Daniel Karl I Weidele, Justin D Weisz, Erick Oduor, Michael Muller, Josh Andres, Alexander Gray, and Dakuo Wang. 2020. AutoAIViz: opening the blackbox of automated artificial intelligence with conditional parallel coordinates. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 308–312.
- [60] Jun Yuan, Changjian Chen, Weikai Yang, Mengchen Liu, Jiazhi Xia, and Shixia Liu. 2021. A survey of visual analytics techniques for machine learning. *Computational Visual Media* 7, 1 (2021), 3–36.
- [61] Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. 2016. Graying the black box: Understanding dqns. In *International conference on machine learning*. PMLR, 1899–1908.
- [62] Jonathan X Zheng, Samraat Pawar, and Dan FM Goodman. 2018. Graph drawing by stochastic gradient descent. *IEEE transactions on visualization and computer graphics* 25, 9 (2018), 2738–2748.

- [63] Marc-André Zöller and Marco F Huber. 2021. Benchmark and survey of automated machine learning frameworks. *Journal of artificial intelligence research* 70 (2021), 409–472.

Received 15 October 2022; revised 03 February 2023; accepted 10 February 2023