De Peuter, Sebastiaan; Oulasvirta, Antti; Kaski, Samuel

Toward AI assistants that let designers design

Published in:
AI Magazine

DOI:
10.1002/aaai.12077

Published: 01/03/2023

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

DOI: 10.1002/aaai.12077

ARTICLE

Toward AI assistants that let designers design

Sebastiaan De Peuter[1] | **Antti Oulasvirta**[2] | **Samuel Kaski**[1,3]

[1]Department of Computer Science, Aalto University, Espoo, Finland

[2]Department of Information and Communications Engineering, Aalto University, Aalto, Finland

[3]Department of Computer Science, University of Manchester, Manchester, UK

Abstract

We need to rethink how we assist designers with artificial intelligence (AI). AI should aim to cooperate, not automate, by supporting and leveraging the creativity and problem-solving capabilities of designers. The challenge for such AI is how to infer designers' goals and then help them without being needlessly disruptive. We introduce AI-assisted design, a new framework for creating such AI, built around generative user models, which allow it to infer and adapt to designers' goals, reasoning, and capabilities.

INTRODUCTION

Over the last decades, artificial intelligence (AI) has been adopted across areas of design and engineering practice. There has been an eager uptake of AI-based methods for *decision-making in design* (Allison et al. 2022), the part of the design process where a design is produced and optimized according to chosen design goals. For example, methods have been developed to help structural engineers create structural elements with minimal materials usage (Fairclough et al. 2019). In computational drug design, there is an ongoing effort to speed up and automate the identification of promising drug molecules using AI (Sliwoski et al. 2014). And in graphic design, generative deep learning methods are helping designers rapidly create appealing posters (Guo et al. 2021).

AI has already transformed the way designers work. However, we argue that it remains hamstrung by inefficient human–AI collaboration. We illustrate this using a simple design problem: planning an enjoyable day trip to a city abroad by selecting a set of points of interest (POIs) to visit (Figure 1). Clearly there will be many enjoyable day trips, but we would like to find one that we will find as enjoyable as possible. Because there may be thousands of POIs to choose from, we would like to enlist the help of an AI, here taking the form of a combinatorial optimization algorithm. But, to do this, we need to give it an explicit objective function, which captures what an enjoyable day trip is for us. This would clearly be challenging. What makes a day trip enjoyable? Is it determined by how many museums we visit? Should we break up the museum visits with some sightseeing? Or shopping? This ambiguity and complexity make it difficult to explain to an AI assistant what makes a day trip good.

This day trip planning problem captures two defining characteristics of problem-solving in design. First, it is usually not clear from the outset what our goals are. Due to its open-ended nature, at the outset, the goals for a design problem are often under-specified. It is up to designers to determine – through exploration and trial and error – which among the many feasible solutions constitute good solutions to the problem and to elucidate a precise set of goals, which captures this. This happens through an iterative process, which involves continuously creating solution candidates and updating the design goals based on insights learned from these candidates. Through this process, design goals that were initially tacit, evolve and concretize as designers work (Bradner, Iorio, and Davis et al. 2014; Dorst and Cross 2001).

The second defining characteristic is that turning these partially tacit, complex sketches of goals into hard terms which AI can operate on – which we will refer to as *goal communication* – is burdensome and prone to error (Amodei et al. 2016), especially when there has been

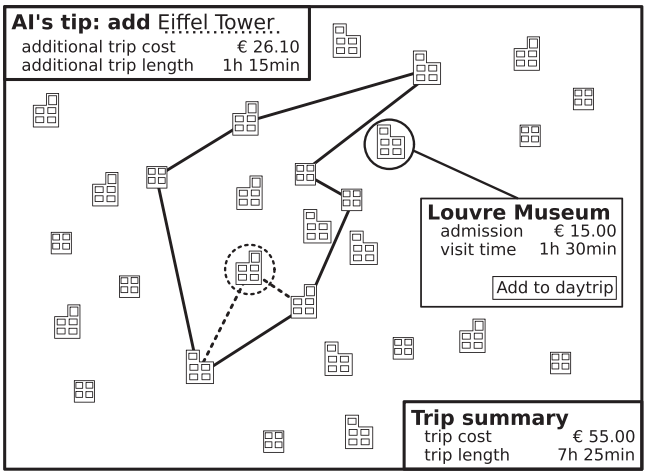

FIGURE 1 AI-assisted day trip design. Many decision-making problems in design are combinatorial by nature and involve underspecified objectives that are refined during the course of design. We illustrate artificial intelligence (AI) assistance for such problems with a trip planning task, where a designer must select a number of points of interest (POI) to visit. This figure shows an example user interface for this problem. A trip is constructed by iteratively selecting POIs (solid circle) to add or remove. We use our proposed AI-assisted design framework to implement an assistant that can infer the designer's goal and then help the designer by recommending POIs to add or remove (dashed circle).

limited opportunity to explore and ideate. A recent study of architects' use of computational design tools notes that "[w]hen architects abstract designs into rules and make observations such as 'you need to write the rules correctly' they begin to sound more like software programmers than architects" (Bradner, Iorio, and Davis 2014, p. 6).

Goal communication is the main barrier to organizing effective collaboration between designers and their AI tools. In many design fields, there is now a plethora of design tools available, but most solve the design process autonomously, without human intervention. The problem is that these rely completely on having correctly communicated goals. When goals are not communicated absolutely correctly, they will solve for the wrong objective. Some tools have addressed this by not requiring a specified goal at all. These function much like recommendation systems, applying previous design solutions to new problems (Guo et al. 2021; Zhang et al. 2021). Though this avoids the goal communication problem, like recommendation systems these approaches require a lot of data and have trouble generalizing beyond the training distribution. Instead, the most common solution has been to substitute the explicit communication of goals by more indirect forms. For example, by asking designers to indicate preferences over a range of designs (Brochu, Brochu, and De Freitas 2010; Koyama, Sato, and Goto 2020; Mikkola et al. 2020), or to sketch partial solutions (Dayama et al. 2020; Igarashi et al. 2007).

Another barrier to effective collaboration is the lack of agency currently given to designers by some AI tools. With the goal of minimizing human labor, AI tools have generally been designed to work autonomously (Dafoe et al. 2021). Even tools ostensibly labeled as interactive do not usually give designers an active role in the design process; and instead consider the designer as a passive source of feedback. When put in these kinds of roles, designers have previously raised concerns over limited ownership, creativity, and expressiveness (Chan et al. 2022), and above all a lack of control (Guo et al., 2021). Yet it is clear that designers are able to play a greater role in the design process. Beyond understanding of the goal, designers bring creativity, intuition, and tacit knowledge that may not be available to the AI system.

Solving both the goal communication problem and the loss of designer agency requires a paradigm shift in AI tools for design. We argue that these tools should be collaborative, as opposed to autonomous (Akata et al. 2020; Dafoe et al. 2021). Design tools should operate like *assistants*: working alongside designers, in support of designers, but in such a way that designers can continue to actively participate in the design process and retain control over the design decisions made. Active participation and control will clearly benefit designer agency, but will also improve goal communication. With the designer actively involved, there is more opportunity for them to explore and develop their goals. And because they are an active participant, the AI can infer their goals from their actions, instead of relying on explicit communication. Further, keeping the designer in the loop mediates potentially inaccurate goal inferences, as any erroneous assistance due to a miscommunicated goal can be corrected immediately by the designer. These corrections not only give the designer more control over the AI, but also help the AI because they are highly informative of the designer's goals.

In this paper, we propose *AI-assisted design* (AIAD) (Figure 2), a general framework for creating collaborative AI assistants. The framework introduces an AI assistant that assists users throughout the design process, without requiring an explicit description of the design goal upfront. The types of assistance available to the assistant will depend on the design domain, but could include recommendations for incremental design changes, or proposals for fundamentally different designs. The designer can choose to accept these recommendations, or ignore them and modify the design themselves. The designer's (current) goals are inferred from their behavior, that is, their interactions with the assistant, and their direct changes to the design. For this purpose, we equip the AI with a *user model*, a generative model of how goals result in the behavior we observe. Beyond goal inference, this model is also used by the assistant to plan how it will interact with the designer. But a designer's behavior is determined by more

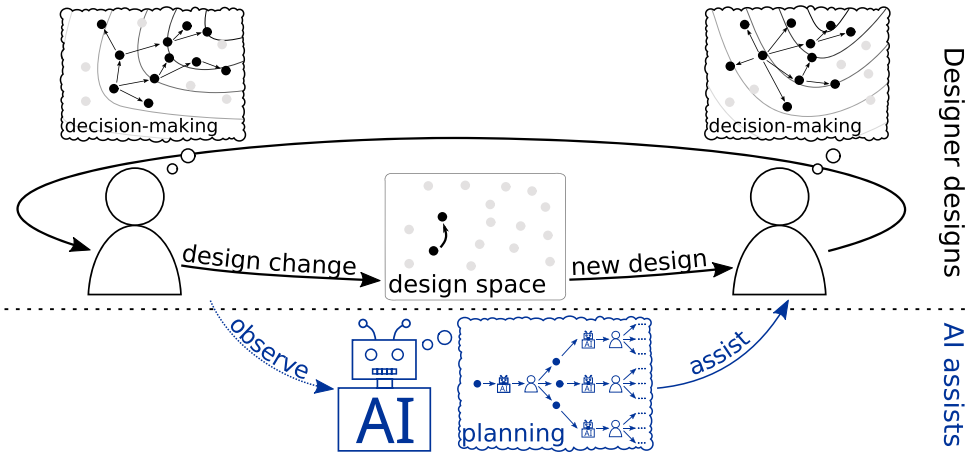

FIGURE 2 Overview of the AI-assisted design framework. Designers work by exploring design changes they think could improve a design. Which change to try next is planned based on a utility function (contours in thought bubbles), which evolves as the designer develops their goals. Artificial intelligence (AI) can help designers solve complex design problems, but should do so in a way that is collaborative and cognizant of this evolving character, and should ensure that designers retain agency and control over the process. To collaborate, it must understand the designer, specifically their goals and behavioral factors that influence their behavior. We propose to create AI assistants (shown in blue) that can infer a designer's goals from observations and can use them to assist them in a collaborative manner.

than just their goals. Biases, cognitive bounds, problem knowledge, and more influence how a designer behaves. We will call these *behavioral factors*. It is important that these are accounted for and inferred jointly with the designer's goals (De Peuter and Kaski 2023; Elmalech et al. 2015). For example, an anchoring bias can cause a designer to fixate on their current design and refuse large changes proposed by an assistant. An assistant that can infer this fixation can focus on proposing small localized design changes or could nudge the designer to explore more by highlighting promising regions in the design space. The framework is extensible: it is not limited to any specific type of user model or assistance and can incorporate more explicit forms of goal communication if desired.

FORMALIZING DECISION-MAKING IN DESIGN

Our framework specifically targets decision-making in design; that is, the cognitive process of deciding features, functions, and properties of a design in the best possible way. We target design problems where the full design space and the set of goals can be defined in advance, but the designer's true goals within this space must be inferred online. A prominent class of such design problems is engineering design problems. These span a wide variety of domains ranging from the design of electro-mechanical systems such as vehicles over civil engineering applications to the design of optimal control policies for complex systems such as inventory management and production planning (Fairclough et al. 2019; Rao 2019). Further, a large number of design problems beyond engineering design match these assumptions. For example, the design of biochemical structures such as drug design (Sliwoski et al. 2014), or user interface design (Dayama et al. 2020).

Two dominant formalizations for these types of design problems exist; they can be formalized as either an optimization problem or a sequential decision problem. When formalized as an optimization problem, we think of it as finding an optimal point under a utility function within a feasible set (Rao 2019). The feasible set is determined by the constraints. The utility function (sometimes called an objective function) encodes the design goals. When formalized as a sequential decision problem, the state space is the space of feasible designs (according to the constraints known a priori) and the actions are *design changes* that take us from one design to another (McComb, Cagan, and Kotovsky 2017; Raina, Cagan, and McComb 2021). The goal is again encoded as a utility function, which the decision process seeks to maximize. For example, in the day trip design problem, every day trip is a state, and every addition or removal of a POI is a design change that results in a new state (a new day trip).

For AIAD, we will formalize design as a sequential decision process. Designers are decision-makers: they reason about the changes they can make to a design and choose one (Figure 2). Say we were planning a trip and needed to choose between visiting the Louvre Museum or the Eiffel Tower. We would consider both options and their implications on future changes – if we add the Eiffel Tower, it will be easy to visit the Musée du Quai Branly Jacques Chirac next as it is close by – and choose what we like most. Integrating a human decision-maker into an

optimization process is difficult because humans do not think or work like optimizers do. However, if design is formalized as a decision process, there is a clear opportunity to create cooperation around the design decisions that have to be made.

We will assume that the design space can be represented as a set, and that actions can be defined such that they allow us to move between elements in this set. These assumptions are necessary to encode a design problem as a sequential decision process. We will also assume that some space of utility functions is given and that the designer's goals and constraints can be encoded as a utility function from this space. This will allow the AI to maintain beliefs about the utility function and reason about the designers' goals. In the trip planning problem, the design space is the collection of all sets of POIs that can be visited in a day. The actions, which add or remove POIs to these sets, allow us to move within this design space.

PRIOR WORK ON COMPUTATIONAL METHODS FOR DESIGN ASSISTANCE

There is a vast literature on AI tools – or computational tools more generally – for assisting designers in the decision-making stage of design. Much of this work is siloed within individual design domains, with many tools designed for specific domains. We provide a high-level overview of the prominent approaches here, with representative applications from various design domains. Some of the approaches covered fall under the umbrella term interactive optimization. For an in-depth survey of interactive optimization, we refer readers to Meignan et al. (2015).

Trial-and-error optimization is the simplest and most common form of design support (Meignan et al. 2015). Here the design tool is a solver, which takes some description of the design goal and constraints as input, and produces an optimal design for that description. There is no goal communication support. Instead, the designer must iteratively refine their goal description, based on what designs the solver is producing, to rectify errors in their goal description.

Leveraging prior solutions: A popular approach for helping designers is to make design recommendations based on previous solutions to similar design problems. These approaches are similar to recommendation systems, and use patterns in previous solutions to find design changes they can recommend for the current task (Jiang et al. 2021). An early approach by Lee et al. (2010), for example, involved a web design tool, which showed designers examples from a corpus of existing websites and allowed them to copy elements to their own website design. Guo et al. (2021) introduced a deep learning system that recommends advertising posters based on a user-provided product picture and taglines. The generated posters were found to be as good as human-designed ones. Raina, McComb, and Cagan (2019b) proposed a system that learns a policy, which imitates human design strategies in a bridge design setting. This policy could be simulated on new problem instances to create new bridge designs. The system was found to produce qualitatively comparable designs to human designers.

The advantage of these approaches is that because they apply patterns, they do not require explicitly given design goals. This eliminates the need for explicit goal communication. The downside, however, is twofold. First, relying on prior solutions makes it difficult to generalize beyond the training distribution. This is an especially pertinent issue for design problems, where the aim is to tackle a novel problem or to find novel solutions. Second, learning to apply or recommend human design patterns can match human performance but not improve upon it. Neither Guo et al. (2021) nor Raina, McComb, and Cagan (2019b) produced designs that were unequivocally better than human-generated designs.

Preference-based approaches use an explicit model of a designer's preferences – formulated as a utility function. The preferences are inferred iteratively through some form of interaction with the designer. Once inferred, they can be passed to a solver to generate an optimal design.

In interactive multi-objective optimization (Deb 2014; Meignan et al. 2015), these preferences are formalized as an aggregation of predefined (and often conflicting) subobjectives into a single utility function. What must be inferred from the designer is how to trade-off these subobjectives within the aggregation. This is learnt through a sequence of interactions. Multiple modes of interaction exist, such as asking the designer to provide reference points, which achieve a reasonable trade-off, or to rate the desirability of system-proposed trade-offs (Deb 2014). Interactive multi-objective optimization has been applied to many real-world design problems, including the design of district heating networks (Laukkanen et al. 2012) and energy systems of large buildings (Pour et al. 2022).

Preferential Bayesian optimization (PBO) (Brochu, Brochu, and De Freitas 2010; González et al. 2017) is similar but infers a complete utility function. This is advantageous in cases where defining subobjectives would be burdensome. PBO infers designers' utility function by iteratively asking them to express their preference over pairs of designs. Brochu, Brochu, and De Freitas (2010) used this approach to find maximally appealing smoke animations. One issue is that pairwise comparisons are not particularly informative, meaning that many queries to the designer are needed to infer their utility function. One solution to this is to ask designers to select their most preferred

point value on one (or more) sliders (Koyama et al. 2017; Koyama, Sato, and Goto 2020; Mikkola et al. 2020). This allows designers to express their preference over an infinite number of designs located on a line (or hyperplane in the case of multiple sliders) within the design space.

Although they can produce optimal designs, preference-based methods have some notable downsides. Interactive multi-objective optimization can only infer trade-offs. PBO is more flexible but as a result needs many iterations of feedback to infer designers' preferences. An issue that both of these approaches share is that they give designers very little agency. Designers have no part in the optimization process beyond providing preference information. Further, they do not account for how biases or other behavioral factors may shape the feedback the designer gives, and do not consider that a designer's goals may evolve as they interact with the system.

Sketch-based approaches: An interesting body of work that has received comparatively little attention so far is based around sketches: designs that are coarse or incomplete. Sketches are easy to produce for humans, are highly informative of designers' intents, and can be refined iteratively with the help of interactive design tools. Igarashi et al. (2007) proposed a tool that took human-drawn sketches and vectorized them as they were drawn. This included inferring geometrical properties like collinearity, and asking the user to resolve ambiguous inferences. More recently, Dayama et al. (2020) introduced a tool that could complete partial website layouts. Designers could either place elements on a canvas, fixing their size and location, or leave them in a workspace. The system would then propose a set of complete designs that incorporated the elements from the workspace on the canvas, taking into account any additional constraints placed on the elements by the designer. Though these approaches show promise, a limitation is that sketches are highly problem-specific. It is, therefore, difficult to transfer approaches from one design domain to another.

General decision-making support: It is clear that there is an opportunity in design for smarter and more capable tools to support designers. Specifically, design tools, which more effectively infer designers' (evolving) goals, preserve designers' agency, and account for behavioral factors. By thinking of design as a decision problem, we are able to leverage existing work on creating collaborative AI assistants that can support agents (human or otherwise) in general decision tasks. These frameworks could in principle be applied to design problems. We detail these frameworks below and explain why they are not immediately suitable.

Cooperative inverse reinforcement learning (CIRL) (Hadfield-Menell et al. 2016) is a cooperative assistance method, which introduces an assistant that works side-by-side with a human. CIRL is similar to the AIAD framework we propose, with the main difference that in CIRL, the assistant has complete autonomy with no human control over it. This means that the designer would have to keep careful watch of the assistant to correct any mistakes it makes. *Shared autonomy* (Javdani, Srinivasa, and Andrew Bagnell 2015) is another cooperative assistance method. Here the human directs an assistant as it solves a decision problem. However, the assistant is given significant freedom in how it acts based on these directions, making the human's control over the assistant limited. Fern et al. (2014) introduced a framework in which a human and an AI assistant act in turns. Both were allowed to skip their turn. This framework is simple but highly flexible: no specific type of interaction was prescribed, and no limits were placed on the action space of the assistant. Like the other frameworks covered here, though, it is not able to infer behavioral factors from interaction and does not support assistance under an evolving goal.

THE AI-ASSISTED DESIGN FRAMEWORK

We propose AI-assisted design (AIAD), a flexible, domain-agnostic framework for implementing collaborative AI assistants for design problems. These assistants primarily help designers by interacting with them. We model the problem of assisting a designer from the assistant's point of view as a decision problem, where every possible interaction with the designer is an action the AI assistant can take. A generative user model allows the assistant to plan these interactions by simulating their effect on the designer, and to infer the designer's utility function and behavioral factors. The framework is flexible in the sense that it is not tied to any specific form of assistance. Appropriate forms of assistance can be implemented for individual applications. Any interaction with the designer that can be represented as one or more actions is, therefore, compatible with the framework.

By defining assistance as a decision problem, AIAD can flexibly accommodate a wide variety of interactions, and can interlace them in any way that is beneficial. Interactions that directly assist the designer include recommending design changes to make, recommending alternative designs, or recommending a continuous subspace of designs, which the designer can explore through sliders, similar to Koyama et al. (2017), Koyama, Sato, and Goto (2020), and Mikkola et al. (2020). These recommendations can be constructed to include justifications. But interactions do not always have to involve the design. One could add an interaction that tries to learn more about the user's goal. Taking inspiration from multi-objective optimization, the assistant could ask the designer for

reference points, ask them to rate objective trade-offs achieved by hypothetical designs, or ask them to compare pairs of designs. Contrary to preference-based approaches, the designer can choose to ignore these questions without compromising the functioning of the assistant.

For the remainder of this paper, we will focus on design change recommendations. We can think of any set of recommendations presented to the user as an interaction, and therefore, an action that is available to the assistant. Unlike in regular recommendation engines, recommendations here are *what-if* scenarios. For every recommended design change, the resulting design is shown together with additional relevant information such as justifications and the design's outcomes – domain-specific measures of its functioning or performance. In Figure 1, a recommended design change – additionally visiting the Eiffel Tower – is shown in the dotted circle, together with information on how the itinerary (dotted lines) and outcomes (top left) would change. This supports the counterfactual thinking which designers engage in when designing (Oulasvirta and Hornbæk 2022). Counterfactual thinking involves hypothesizing about the effects of design changes on the design and its outcomes. For example, when considering whether we want to visit the Eiffel Tower, we must gauge the effect of this on the length and cost of our trip. But counterfactual thinking is difficult: designers must contend with a vast number of counterfactuals that are usually hard to reason about. There are generally many potential design changes, and outcomes are complex functions of the designs these changes produce. Recommendations help the designer filter the set of potential counterfactuals and focus on those that look promising. They also simplify the counterfactual reasoning itself by including information about the outcomes.

CREATING A COLLABORATIVE ASSISTANT

Successful cooperation requires two capabilities: that the assistant (i) can estimate the designer's utility and – given that utility – (ii) can predict how the designer will behave in future situations (Dafoe et al. 2021). The first ensures that the assistant can act in the interest of the designer, for example, by making recommendations that improve the design. The second allows the assistant to plan its assistance based on how the designer will react to it. Designers may have bounds and biases, and may not possess complete knowledge. These factors can lead them to behave in ways that appear irrational to the assistant (Elmalech et al. 2015). To mitigate the effect of this, the assistant must adapt its choice of interactions to these factors (De Peuter and Kaski 2023).

Modeling designers

To allow the assistant to act collaboratively, we propose to equip it with a *generative user model*, that is, a generative model of how human reasoning translates an internal utility function into behavior. It models the designer's behavior on every type of interaction the assistant could have with the designer. This allows the assistant to evaluate possible actions by forward-simulating their effect on the user. This is very different from earlier work on user modeling, which has focused on modeling knowledge and preferences for personalization (Kobsa 2001). There is a similarity to recent work on modeling human decision-making and strategies in design problems, which has tried to imitate and analyze human behavior using Markov chains (McComb, Cagan, and Kotovsky 2017) and neural networks (Raina, McComb, and Cagan 2019b; Raina, Cagan, and McComb 2021) and has sought to use human strategies to tune solution methods (Raina, Cagan, and McComb 2019a; Sexton and Ren 2017). An important difference, however, is that we model designers who are interacting with an assistant, with a special focus on how that interaction influences how the designer changes the design.

Human designers almost never exhibit optimal behavior because of incomplete problem knowledge, and cognitive limits exposed in research on design cognition (Cross 2001). We call these factors *behavioral factors*. Designers rely on "fast and frugal" decision heuristics, which may create biases in their thinking. For example, when designers fixate on minor variations of a certain known solution to a design problem, we call that an anchoring bias. Also, because of limits on human long-term memory, they may also get stuck, or fixated, on ideas, showing inability to generate novel candidate solutions. Such factors are highly idiosyncratic. For example, though all designers exhibit anchoring, some designers will fixate more strongly on the neighborhood of a certain design than others (Cross 2001, 2004). Equally, the design they anchor to may differ. In order to capture the personal variations within these factors, we can parameterize them within the user model. These parameters, which we will call behavioral parameters, then allow us to infer the personal cognitive bounds of a designer.

The behavioral parameters can be inferred from observations, together with the utility function, up to limitations on identifiability (Mindermann and Armstrong 2018). If the user model correctly captures these factors, their effect on the designer's behavior can be separated from that of the utility. This generative user model allows the assistant to perform the two types of reasoning necessary for cooperation. (i) Inference of the user model parameters and the utility function allows the assistant to act in the

designer's interest. (ii) By simulating the user model, once the utility has been inferred, the assistant can predict its effect on the designer's behavior and plan its assistance accordingly.

Though AIAD is agnostic to any specific user modeling approach, we see computational rationality (Lewis, Howes, and Singh 2014; Oulasvirta, Jokinen, and Howes 2022) as a promising theory on which to build user models. Computational rationality is a cognitive theory that postulates that apparently irrational human behavior corresponds to rational utility-maximizing behavior under a set of bounds and a subjective model of the problem. The utility-maximizing behavior can be found using standard reinforcement learning methods. In the case of design, the utility being maximized is the internal utility of the designer, and the subjective model of the task corresponds to the designer's own understanding of the design problem. As a theory, computational rationality has proven able to accurately predict human behavior in various tasks (Callaway et al. 2022; Chen, Chang, and Howes 2021; Chen et al. 2015; Jokinen et al. 2021). By parameterizing the bounds and a subjective task model, a computationally rational model can generate a wide variety of behaviors. Prior rule-based cognitive models and general machine learning models (neural networks, Markov chains, …) can also generate a variety of behaviors. However, the former are less robust and require a lot of handcrafting to adapt them to new domains and applications (Oulasvirta, Jokinen, and Howes 2022), while training the latter would require large amounts of data because it lacks the prior knowledge encoded in a computationally rational model.

Do we need perfect models? Like any statistical method, mis-specification in the models on which our framework is built could negatively affect the performance of the assistant. Mis-specification in the user model will reduce the accuracy of the predicted behavior, and – as inferences about the utility are based on it – will also harm the correctness of the assistant's beliefs about the designer's utility. Faithfully modeling the complexities of human behavior may seem like a lofty goal, but experience shows that the user model need not be perfect to be useful. Prior work on interactive decision-making has shown success using only relatively simple models of human behavior (Hadfield-Menell et al. 2016; Reddy, Dragan, and Levine 2018). Mis-specification in the utility function space, on the other hand, could mean that no utility function can accurately represent the designer's goals. This would create an irreducible discrepancy between the designer's goals and the assistant's beliefs about those goals, and limit the assistant's ability to help in the mis-specified aspects. The risk that we mis-specify the utility space will be larger for more complex and open-ended design problems. However, preliminary results show that the proposed framework is robust to some mis-specification in the utility space (De Peuter and Kaski 2023).

Planning interactions

For the assistant, the interactions with the designer form a decision process. This process has as actions the interactions and as state the current design and the assistant's belief about the designer's utility function and behavioral factors. For the assistant to be useful, this decision process must be *aligned* (Amodei et al. 2016) with the designer's decision problem – the problem of creating a design and interacting with the assistant – meaning that it should seek to maximize the designer's own utility function. As the utility function is unknown, the assistant must correctly infer it to achieve alignment.

The assistant plans over this decision process to find a policy for interacting with the designer. The user model supports this planning by allowing the assistant to simulate how the designer will change the design in response to interactions, allowing it to predict the value of every interaction and its effect on future interactions. The exact form of this decision process depends on the design problem and the supported interactions. In general, it will fall under one of three formalisms, depending on the assumptions made about the designers' utility function and behavioral factors. If they are unknown but fixed, the decision problem can be formalized as a *generalized hidden-parameter Markov decision process* (GHP-MDP) (De Peuter and Kaski 2023; Perez, Such, and Karaletsos 2020). If we assume that they change, but we know how they change, we can formalize the decision problem as a *partially observable Markov decision process* (POMDP) (Shah et al. 2020; Spaan 2012). If, however, the way they change is unknown, it can be formalized as a *Bayes-adaptive POMDP* (Çelikok, Oliehoek, and Kaski 2022; Ross et al. 2011).

When planning, the assistant must take into account how useful potential interactions will be to the designer, and how much the resulting observations will improve future assistance. Interactions have a direct and indirect effect on the design process. The direct effect is the increase in utility, which results from interactions – for example, from good design change suggestions. The indirect effect comes from the information the interaction reveals about the designer's utility function or behavioral factors. This information allows the assistant to better estimate these variables and thereby provide better assistance in the future. To be optimal, the assistant's planning must consider both of these effects.

Should assistant be able to act autonomously?

The proposed framework can easily be extended to additionally support interactions that directly change the design, without the involvement of the designer. This allows the assistant to make improvements to the design, without requiring any effort from the designer, when it is certain they are in their interests. However, such an assistant will face similar issues to tools that simply automate. The increased agency available to the assistant will inherently decrease the control of the designer. And when uncertainty about the goal is high, the assistant may follow the wrong goals to improve the design. It would then be up to the designer to undo any undesirable changes the assistant made. Fortunately, the interactive nature of the assistant makes it straightforward to mitigate and curtail these potentially negative effects.

When we equip an assistant with interactions that give it direct access to the design, it is essential that we also equip it with interactions where the designer is in control, such as recommendations. Because the proposed framework maintains Bayesian beliefs about the designer's utility function, it is able to judge the risk involved in any interaction. Thus, it can determine whether it is certain enough about the utility to act autonomously, or whether it would be better to influence change by interacting with the designer. Recommendations are an excellent choice here because bad recommendations have minimal impact on the design process; the designer can reject them without further effect.

To ensure that designers maintain their desired level of control, they should be able to choose how much agency the assistant has. At minimum, the designer should be asked at the start of the design process whether the assistant should be able to act autonomously. More fine-grained controls could include curtailing the assistant's direct access to certain parts of the design, or restricting how much it can change the design at once. These restrictions could be set either when implementing the assistant, or during the design process by designers themselves. In the latter case, it is important that we give the designer clear intuition about what degree of agency they are selecting.

PRELIMINARY RESULTS

Preliminary results show that AIAD helps designers produce significantly better designs than alternative forms of design assistance (De Peuter and Kaski 2023). We report here on earlier work that applied AIAD to the day trip design problem introduced in Figure 1. We made a number of simplifying assumptions in this earlier work, namely that the user model was correct and that the utility function and behavioral factors were unknown but fixed. These imply that the designer has no memory and cannot learn about its own goals. The AIAD implementation was, therefore, much simpler than what we have introduced in this paper.

In the day trip design problem, the designer has to choose a subset of POIs, which can be visited within 1 day (12 h) from a set of 100 POIs that make up a city. The duration of a day trip includes the time needed to visit the POIs but also the time it takes to walk between them. The designer does not have to choose the order in which to visit the POIs; the system automatically generates the order that minimizes travel time. This simplifies the routing aspect of the design problem, which involves solving a traveling salesperson problem (TSP). The utility of a design is a product of two scores, one scoring the monetary cost of the day trip (lower is better), and another measuring the time-weighted enjoyment of visiting the POIs themselves. The enjoyment of visiting a POI is determined by the designer's interests.

We chose this design problem specifically because it is easy to understand for a general audience, and replicates many core difficulties of design problems. At any point in time, there is a large number of changes that could be made to the design (any one of the 100 POIs can be added or removed), while only a few of them will significantly improve the design. Further, counterfactual reasoning about how changes will affect the design is hard: predicting how the duration will change involves solving a TSP. The vast size of the design space and the need for complex counterfactual reasoning add to the generalizability of the results presented here. However, due to its constrained design space and relatively well-defined goal space, this problem is most typical of engineering design, rather than more open-ended design problems.

We created an AI assistant with two types of interactions at its disposal. It could choose to recommend a design change to the designer, that is, it could recommend a specific POI to be included or removed, or it could choose to edit the design directly. Because we assumed that the utility function and behavioral factors were fixed, we modeled the decision problem as a GHP-MDP. The hidden parameters were the designer's utility function, defined by a parameter representing their cost tolerance and parameters representing their interests, and a single behavioral factor, namely whether an anchoring bias was present or not. Designers with an anchoring bias were assumed to refuse to consider adding any POI that was located more than 500 m from the itinerary (solid line between the POIs in Figure 1). The user model was strictly speaking not a computationally rational model, but a model built on a

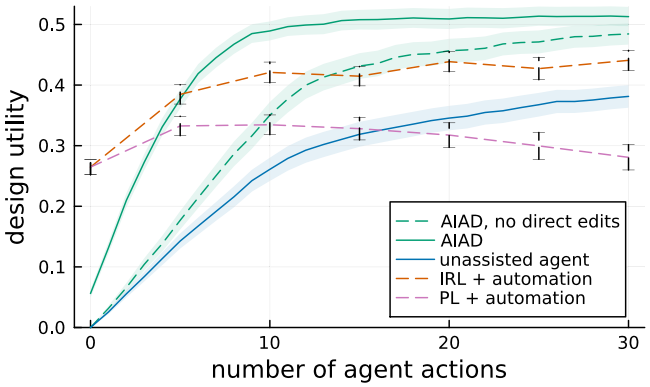

FIGURE 3 Effectiveness of assistance. We compare the effectiveness of various types of assistance, including our proposed framework, in a day trip design process. This graph shows the utility of the design produced as a function of the designer's effort. The shading shows the standard error around the mean over multiple experiments.

bounded rational theory of human decision-making based on information processing costs (Genewein et al. 2015). Further, to reason about how the addition of a POI would alter the duration of a trip, our user model used a visual heuristic that is often used by humans when solving TSPs (MacGregor, Ormerod, and Chronicle 2000).

We compared our implemented AI assistant to three baselines in a simulation study. The designer in these studies was a simulated instance of our user model with a randomly sampled utility function. Half of the simulated designers had an anchoring bias. In every experiment, we tested five types of design processes, all consisting of the same designer assisted through different methods, and measured the quality of the design by the designer's utility function throughout the process.

Figure 3 shows the design's utility produced in these design processes as a function of the effort of the designer, measured here by the number of actions taken. As actions, we count both changes to the design and interactions with the relevant assistance method. The first baseline considered (preference learning [PL] + automation) was a PL implementation, which used pairwise comparisons to elicit the designer's utility function, and then produced a design by optimizing the inferred utility function. The second baseline (inverse reinforcement learning [IRL] + automation) is similar, but used IRL (Abbeel and Ng 2004) to infer the utility function from observations of the designer working unassisted. The third baseline was a designer without any assistance. These were compared to our AIAD implementation. We considered two versions: one that was allowed to edit the design directly (AIAD), and one that was not (AIAD, no direct edits). The latter variant can only make recommendations and thus gives the designer complete control over the design process.

Our results show that AIAD helped simulated designers produce significantly better designs than the baselines. Figure 3 shows the utility of produced designs as a function of designer effort, measured here by the number of actions taken. For PL + automation, every preference over a pair of designs was counted as one action. We see that after eight actions taken, designers assisted by AIAD produced better designs than designers assisted by the baseline methods for the same amount of effort. Comparing the two versions of AIAD, we find that allowing the assistant to directly edit the design generally had a positive effect on the design process.

Giving the assistant the ability to both directly change the design and give recommendations was essential here. When the assistant was uncertain about the designer's utility function, it could use recommendations. But when it was certain enough about the value of a design change, it could reduce the effort required from the designer by applying it automatically. Direct edits also allowed it to carry out changes that would benefit the designer but that would have been rejected solely due to the designer's anchoring bias. If the designer had refused this level of agency (this corresponds to the "AIAD, no direct edits" baseline), they would eventually have reached a similar quality of design, though it would have taken more effort.

SUMMARY AND FUTURE WORK

In this paper, we have introduced AI-assisted design (AIAD), a framework for developing collaborative assistants for design problems. We reported on preliminary results that provided evidence that this type of assistance has practical benefits. However, a number of challenges remain before the framework can be applied to arbitrarily complex design settings. We highlight three prominent open challenges here.

The assistant should have a multitude of interactions available to it. This paper has primarily focused on recommendations as a general type of interaction. More specialized forms of interaction will likely be more suitable for specific design problems. To offer the best assistance it can, the assistant should have multiple types of interaction at its disposal. These interactions should be developed to be maximally useful to designers while also being informative to the assistant.

Cooperation with the designer requires a sufficiently accurate user model. To cooperate, the assistant needs a good model of the designer's behavior. AIAD does not commit to a single modeling paradigm, but we have argued that computational rationality is an especially promising option for producing scientifically grounded models.

New machine learning methods are needed to support planning and inference with the user model. Inferring the utility and behavioral factors that parameterize the user model needs to be done in a data-efficient way to minimize unnecessary interactions. This will require a combination of effective interaction planning and novel inference methods. The multi-agent nature of cooperation requires complex nested reasoning within the user model, making forward simulation computationally expensive. Creating assistants that can swiftly provide helpful assistance, therefore, requires new machine learning methods and approximations to ensure that interaction planning can be done in real time.

ACKNOWLEDGMENTS

We would like to thank Julien Gori, Andrew Howes, Jussi Jokinen, Pierre-Alexandre Murena, and Shibei Zhu for their valuable feedback and suggestions. This work was supported by the Academy of Finland (flagship program: Finnish Center for Artificial Intelligence, FCAI, grants 328400, 345604, 341763; BAD, grant 318559; Human Automata, grant 328813) and UKRI Turing AI World-Leading Researcher Fellowship, EP/W002973/1.

CONFLICT OF INTEREST

The authors declare that there is no conflict.

ORCID

Sebastiaan De Peuter https://orcid.org/0000-0002-0684-0110
Antti Oulasvirta https://orcid.org/0000-0002-2498-7837
Samuel Kaski https://orcid.org/0000-0003-1925-9154

REFERENCES

Abbeel, Pieter, and Andrew Y. Ng. 2004. "Apprenticeship Learning via Inverse Reinforcement Learning." In *Proceedings of the Twenty-First International Conference on Machine Learning*, New York, New York, USA: Association for Computing Machinery. https://doi.org/10.1145/1015330.1015430.

Pour, Pouya Aghaei, Tobias Rodemann, Jussi Hakanen, and Kaisa Miettinen. 2022. "Surrogate Assisted Interactive Multiobjective Optimization in Energy System Design of Buildings." *Optimization and Engineering* 23: 303–27. https://doi.org/10.1007/s11081-020-09587-8.

Akata, Zeynep, Dan Balliet, Maarten De Rijke, Frank Dignum, Virginia Dignum, Guszti Eiben, Antske Fokkens, et al. 2020. "A Research Agenda for Hybrid Intelligence: Augmenting Human Intellect with Collaborative, Adaptive, Responsible, and Explainable Artificial Intelligence." *Computer* 53(08): 18. https://doi.org/10.1109/MC.2020.2996587.

Allison, James T., Michel-Alexandre Cardin, Chris McComb, Max Yi Ren, Daniel Selva, Conrad Tucker, Paul Witherell, and Yaoyao Fiona Zhao. 2022. "Special Issue: Artificial Intelligence and Engineering Design." *Journal of Mechanical Design* 144(2): 020301. https://doi.org/10.1115/1.4053111.

Amodei, Dario, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. arXiv preprint arXiv:1606.06565.

Bradner, Erin, Francesco Iorio, and Mark Davis. 2014. "Parameters Tell the Design Story: Ideation and Abstraction in Design Optimization." In *Proceedings of the Symposium on Simulation for Architecture & Urban Design*, volume 26, 1–8, San Diego, California, USA: Society for Computer Simulation International.

Brochu, Eric, 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*, 103–12, Goslar, Germany: Eurographics Association.

Callaway, Frederick, Bas van Opheusden, Sayan Gul, Priyam Das, Paul M Krueger, Falk Lieder, and Thomas L Griffiths. 2022. "Rational Use of Cognitive Resources in Human Planning." *Nature Human Behaviour* : 1112–25. https://doi.org/10.1038/s41562-022-01332-8.

Çelikok, Mustafa Mert, Frans A Oliehoek, and Samuel Kaski. 2022. "Best-response Bayesian Reinforcement Learning with Bayes-adaptive POMDPs for Centaurs." In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, 235–43, Richland, South Carolina, USA: International Foundation for Autonomous Agents and Multiagent Systems.

Chan, Liwei, Yi-Chi Liao, George B Mo, John J Dudley, Chun-Lien Cheng, Per Ola Kristensson, and Antti Oulasvirta. 2022. "Investigating Positive and Negative Qualities of Human-in-the-loop Optimization for Designing Interaction Techniques." In *CHI Conference on Human Factors in Computing Systems*, 1–14. New York, New York, USA: Association for Computing Machinery. https://doi.org/10.1145/3491102.3501850.11.

Chen, Haiyang, Hyung Jin Chang, and Andrew Howes. 2021. "Apparently Irrational Choice as Optimal Sequential Decision Making." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 792–800. AAAI Press.

Chen, Xiuli, Gilles Bailly, Duncan P Brumby, Antti Oulasvirta, and Andrew Howes. 2015. "The Emergence of Interactive Behavior: A Model of Rational Menu Search." In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 4217–26. New York, New York, USA: Association for Computing Machinery. https://doi.org/10.1145/2702123.2702483.

Cross, Nigel. 2001. "Design Cognition: Results from Protocol and Other Empirical Studies of Design Activity." In *Design Knowing and Learning: Cognition in Design Education*, 79–103. Oxford: Elsevier Science.

Cross, Nigel. 2004. "Expertise in Design: An Overview." *Design Studies* 25(5): 427–41. https://doi.org/10.1016/j.destud.2004.06.002.

Dafoe, Allan, Yoram Bachrach, Gillian Hadfield, Eric Horvitz, Kate Larson, and Thore Graepel. 2021. "Cooperative AI: Machines Must Learn to Find Common Ground." *Nature* 593: 33–6. https://doi.org/10.1038/d41586-021-01170-0.

Dayama, Niraj, Kashyap Todi, Taru Saarelainen, and Antti Oulasvirta. 2020. "GRIDS: Interactive Layout Design with Integer Programming." In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI'20)*, 1–13. New York, New York, USA: Association for Computing Machinery. https://doi.org/10.1145/3313831.3376553.

De Peuter, Sebastiaan, and Samuel Kaski. 2023. "Zero-shot Assistance in Sequential Decision Problems." In *Thirty-Seventh AAAI*

Conference on Artificial Intelligence, Austin, Texas, USA: AAAI Press. To appear. arXiv:2202.07364.

Deb, Kalyanmoy. 2014. "Multi-objective Optimization." In *Search Methodologies*, 403–49. Boston, Massachusetts, USA: Springer. https://doi.org/10.1007/978-1-4614-6940-7.

Dorst, Kees, and Nigel Cross. 2001. "Creativity in the Design Process: Co-evolution of Problem-solution." *Design Studies* 22(5): 425–37. https://doi.org/10.1016/S0142-694X(01)00009-6.

Elmalech, Avshalom, David Sarne, Avi Rosenfeld, and Eden Shalom Erez. 2015. "When Suboptimal Rules." In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 1313–9. Austin, Texas, USA: AAAI Press.

Fairclough, H., M. Gilbert, C. Thirion, A. Tyas, and P. Winslow. 2019. "Optimisation-driven Conceptual Design: Case Study of a Large Transfer Truss." *The Structural Engineer* 97(10): 20–6.

Fern, Alan, Sriraam Natarajan, Kshitij Judah, and Prasad Tadepalli. 2014. "A Decision-theoretic Model of Assistance." *Journal of Artificial Intelligence Research* 50: 71–104. https://doi.org/10.1613/jair.4213.

Genewein, Tim, Felix Leibfried, Jordi Grau-Moya, and Daniel Alexander Braun. 2015. "Bounded Rationality, Abstraction, and Hierarchical Decision-Making: An Information-Theoretic Optimality Principle." *Frontiers in Robotics and AI* 2. https://doi.org/10.3389/frobt.2015.00027.

González, Javier, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. 2017. "Preferential Bayesian Optimization." In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 1282–91. PMLR.

Guo, Shunan, Zhuochen Jin, Fuling Sun, Jingwen Li, Zhaorui Li, Yang Shi, and Nan Cao. 2021. "Vinci: An Intelligent Graphic Design System for Generating Advertising Posters." In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–17, New York, New York, USA: Association for Computing Machinery. https://doi.org/10.1145/3411764.3445117.

Hadfield-Menell, Dylan, Stuart J Russell, Pieter Abbeel, and Anca Dragan. 2016. "Cooperative Inverse Reinforcement Learning." In *Advances in Neural Information Processing Systems*, volume 29, 3909–17.

Igarashi, Takeo, Satoshi Matsuoka, Sachiko Kawachiya, and Hidehiko Tanaka. 2007. "Interactive Beautification: A Technique for Rapid Geometric Design." In *ACM SIGGRAPH 2007 Courses*, 18. New York, New York, USA: Association for Computing Machinery. https://doi.org/10.1145/1281500.1281529.

Javdani, Shervin, Siddhartha S Srinivasa, and J Andrew Bagnell. 2015. "Shared Autonomy via Hindsight Optimization." In *Proceedings of Robotics: Science and Systems*. https://doi.org/10.15607/RSS.2015.XI.032.

Jiang, Shuo, Jie Hu, Kristin L. Wood, and Jianxi Luo. 2021. "Data-Driven Design-by-Analogy: State-of-the-Art and Future Directions." *Journal of Mechanical Design* 144(2): 020801. https://doi.org/10.1115/1.4051681.

Jokinen, Jussi, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. 2021. "Touchscreen Typing as Optimal Supervisory Control." In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–14. New York, New York, USA: Association for Computing Machinery. https://doi.org/10.1145/3411764.3445483.

Kobsa, Alfred. 2001. "Generic User Modeling Systems." *User Modeling and User-Adapted Interaction* 11(1): 49–63. https://doi.org/10.1023/A:1011187500863.

Koyama, Yuki, Issei Sato, Daisuke Sakamoto, and Takeo Igarashi. 2017. "Sequential Line Search for Efficient Visual Design Optimization by Crowds." *ACM Transactions on Graphics (TOG)* 36(4): 1–11. https://doi.org/10.1145/3072959.3073598.

Koyama, Yuki, Issei Sato, and Masataka Goto. 2020. "Sequential Gallery for Interactive Visual Design Optimization." *ACM Transactions on Graphics* 39(4). https://doi.org/10.1145/3386569.3392444.

Laukkanen, Timo, Tor-Martin Tveit, Vesa Ojalehto, Kaisa Miettinen, and Carl-Johan Fogelholm. 2012. "Bilevel Heat Exchanger Network Synthesis with an Interactive Multi-Objective Optimization Method." *Applied Thermal Engineering* 48: 301–16. https://doi.org/10.1016/j.applthermaleng.2012.04.058.

Lee, Brian, Savil Srivastava, Ranjitha Kumar, Ronen Brafman, and Scott R Klemmer. 2010. "Designing with Interactive Example Galleries." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2257–66. New York, New York, USA: Association for Computing Machinery. https://doi.org/10.1145/1753326.1753667.

Lewis, Richard L, Andrew Howes, and Satinder Singh. 2014. "Computational Rationality: Linking Mechanism and Behavior through Bounded Utility Maximization." *Topics in Cognitive Science* 6(2): 279–311. https://doi.org/10.1111/tops.12086.

MacGregor, James N., Thomas C Ormerod, and EP Chronicle. 2000. "A Model of Human Performance on the Traveling Salesperson Problem." *Memory & Cognition* 28(7): 1183–90. https://doi.org/10.3758/BF03211819.

McComb, Christopher, Jonathan Cagan, and Kenneth Kotovsky. 2017. "Capturing Human Sequence-Learning Abilities in Configuration Design Tasks through Markov Chains." *Journal of Mechanical Design* 139(9): 091101. https://doi.org/10.1115/1.4037185.

Meignan, David, 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): 1–43. https://doi.org/10.1145/2808234.

Mikkola, Petrus, Milica Todorović, Jari Jarvi, Patrick Rinke, and Samuel Kaski. 2020. "Projective Preferential Bayesian Optimization." In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, 6884–92, PMLR.

Mindermann, Soren, and Stuart Armstrong. 2018. "Occam's Razor is Insufficient to Infer the Preferences of Irrational Agents." In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 5603–14.

Oulasvirta, Antti, and Kasper Hornbæk. 2022. "Counterfactual Thinking: What Theories Do in Design." *International Journal of Human–Computer Interaction* 38(1): 78–92. https://doi.org/10.1080/10447318.2021.1925436.

Oulasvirta, Antti, Jussi PP Jokinen, and Andrew Howes. 2022. "Computational Rationality As a Theory of Interaction." In *CHI Conference on Human Factors in Computing Systems*, 1–14, New York, New York, USA: Association for Computing Machinery. https://doi.org/10.1145/3491102.3517739.

Perez, Christian, Felipe Petroski Such, and Theofanis Karaletsos. 2020. "Generalized Hidden Parameter MDPs: Transferable Model-Based RL in a Handful of Trials." In *Proceedings of the AAAI*

Conference on Artificial Intelligence, volume 34, 5403–11, AAAI Press.

Raina, Ayush, Jonathan Cagan, and Christopher McComb. 2019a. "Transferring Design Strategies from Human to Computer and across Design Problems." *Journal of Mechanical Design* 141(11): 114501. https://doi.org/10.1115/1.4044258.

Raina, Ayush, Christopher McComb, and Jonathan Cagan. 2019b. "Learning to Design from Humans: Imitating Human Designers through Deep Learning." *Journal of Mechanical Design* 141(11): 111102. https://doi.org/10.1115/1.4044256.

Raina, Ayush, Jonathan Cagan, and Christopher Mc Comb. 2021. "Design Strategy Network: A Deep Hierarchical Framework to Represent Generative Design Strategies in Complex Action Spaces." *Journal of Mechanical Design* 144(2): 1–36. https://doi.org/10.1115/1.4052566.

Rao, Singiresu S. 2019. *Engineering Optimization: Theory and Practice*. Hoboken, New Jersey, USA: John Wiley & Sons. https://doi.org/10.1002/9781119454816.

Reddy, Siddharth, Anca D. Dragan, and Sergey Levine. 2018. "Where Do You Think You're Going?: Inferring Beliefs about Dynamics from Behavior." In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 1461–72.

Ross, Stéphane, Joelle Pineau, Brahim Chaib-draa, and Pierre Kreitmann. 2011. "A Bayesian Approach for Learning and Planning in Partially Observable Markov Decision Processes." *Journal of Machine Learning Research* 12(48): 1729–70.

Sexton, Thurston, and Max Yi Ren. 2017. "Learning an Optimization Algorithm through Human Design Iterations." *Journal of Mechanical Design* 139(10): 101404. https://doi.org/10.1115/1.4037344.

Shah, Rohin, Pedro Freire, Neel Alex, Rachel Freedman, Dmitrii Krasheninnikov, Lawrence Chan, Michael Dennis, Pieter Abbeel, Anca Dragan, and Stuart Russell. 2020. "Benefits of Assistance over Reward Learning." In Workshop on Cooperative AI (Cooperative AI @ NeurIPS 2020).

Sliwoski, Gregory, Sandeepkumar Kothiwale, Jens Meiler, and Edward W Lowe. 2014. "Computational Methods in Drug Discovery." *Pharmacological Reviews* 66(1): 334–95. https://doi.org/10.1124/pr.112.007336.

Spaan, Matthijs TJ. 2012. "Partially Observable Markov Decision Processes." In *Reinforcement Learning*, 387–414. Berlin, Germany: Springer. https://doi.org/10.1007/978-3-642-27645-3.

Zhang, Guanglu, Ayush Raina, Jonathan Cagan, and Christopher McComb. 2021. "A Cautionary Tale about the Impact of AI on Human Design Teams." *Design Studies* 100990, 72. https://doi.org/10.1016/j.destud.2021.100990.

How to cite this article: De Peuter, Sebastiaan, Antti Oulasvirta, and Samuel Kaski. 2023. "Toward AI assistants that let designers design." *AI Magazine* 44: 85–96. https://doi.org/10.1002/aaai.12077

AUTHOR BIOGRAPHIES

Sebastiaan De Peuter is a Doctoral Candidate at Aalto University, Finland. Contact him at sebastiaan.depeuter@aalto.fi.

Antti Oulasvirta is a computational cognitive Scientist and Associate Professor at Aalto University, Finland. He leads the interactive AI research program at the Finnish Center for Artificial Intelligence FCAI. Contact him at antti.oulasvirta@aalto.fi.

Samuel Kaski is a Professor of Computer Science at Aalto University and Professor of AI at the University of Manchester. He leads the Finnish Center for Artificial Intelligence FCAI and ELLIS unit Helsinki. Contact him at samuel.kaski@aalto.fi.