

Empirical evidence and computational assessment on design knowledge transferability

Molla H. Rahman¹, Alparslan E. Bayrak ² and Zhenghui Sha ³

¹Department of Mechanical Engineering, University of Arkansas, Fayetteville, AR, USA

²Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, USA

³Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX, USA

Abstract

Developing an artificial design agent that mimics human design behaviors through the integration of heuristics is pivotal for various purposes, including advancing design automation, fostering human-AI collaboration, and enhancing design education. However, this endeavor necessitates abundant behavioral data from human designers, posing a challenge due to data scarcity for many design problems. One potential solution lies in transferring learned design knowledge from one problem domain to another. This article aims to gather empirical evidence and computationally evaluate the transferability of design knowledge represented at a high level of abstraction across different design problems. Initially, a design agent grounded in reinforcement learning (RL) is developed to emulate human design behaviors. A data-driven reward mechanism, informed by the Markov chain model, is introduced to reinforce prominent sequential design patterns. Subsequently, the design agent transfers the acquired knowledge from a source task to a target task using a problem-agnostic high-level representation. Through a case study involving two solar system designs, one dataset trains the design agent to mimic human behaviors, while another evaluates the transferability of these learned behaviors to a distinct problem. Results demonstrate that the RL-based agent outperforms a baseline model utilizing the first-order Markov chain model in both the source task without knowledge transfer and the target task with knowledge transfer. However, the model's performance is comparatively lower in predicting the decisions of low-performing designers, suggesting caution in its application, as it may yield unsatisfactory results when mimicking such behaviors.

Keywords: Sequential design decisions, Reinforcement learning, Transfer learning, Design behaviors, Design representation

1. Introduction

With the advent of powerful machine learning algorithms, a variety of intelligent agents have been developed that can enhance automation and relieve human labor. Studies in the design field report that artificial agents are capable of solving well-defined design problems, whereas human heuristics are more efficient in certain

Received 03 August 2022

Revised 13 March 2024

Accepted 19 March 2024

Corresponding author

Z. Sha

zsha@austin.utexas.edu

© The Author(s), 2024. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike licence (<http://creativecommons.org/licenses/by-nc-sa/4.0>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the same Creative Commons licence is used to distribute the re-used or adapted article and the original article is properly cited. The written permission of Cambridge University Press must be obtained prior to any commercial use.

Des. Sci., vol. 10, e10

journals.cambridge.org/dsj

DOI: [10.1017/dsj.2024.7](https://doi.org/10.1017/dsj.2024.7)

the **Design Society**
a worldwide community

 **CAMBRIDGE**
UNIVERSITY PRESS

tasks, such as abstract decision-making or finding intuitive explanations for design decisions (Sexton & Ren 2017; Raina, Cagan & McComb 2019). Therefore, there is a significant potential to complement human design thinking with state-of-the-art computational algorithms in a human-AI collaboration framework, particularly for novice designers who do not have much know-how. An AI that can mimic human behaviors in such a framework could provide several benefits due to its ability to predict human design decisions and intervene when necessary to recommend alternative design strategies.

Developing an intelligent design agent usually requires a large amount of data or a simulation environment that can capture key aspects of the design problem of interest. However, such data that can be used to train a capable AI are usually scarce. This is because, typically, there are not a large number of people working on the same design problem, let alone the time cost of collecting design data from each designer. Therefore, it is valuable to transfer the design knowledge acquired in one problem to another using intelligent agents. Here, design knowledge refers to the set of knowledge, skills, and expertise that designers possess, which guides their decision-making and actions throughout the design process (Cross 2023). Although different designers have different decision-making processes and associated design actions, there could be common patterns embedded in their actions. Design knowledge and design patterns are closely related as design patterns are one way of representing and communicating design knowledge. Design knowledge can be seen as knowledge of the patterns and relationships that exist between actions performed by different designers. However, design knowledge is generally tacit and embedded in design decisions and actions. Extracting and transferring such implicit and tacit knowledge and identifying beneficial design patterns is both scientifically and technically challenging.

Transfer learning, which has emerged as a research area in the machine learning domain, is a method where the goal is to obtain pre-trained values (knowledge) in a computational model and to use them for a new problem (Lazaric 2012). Transfer learning saves training time and resources in new problem-solving by reusing an already trained model from the source problem instead of training a new model and choosing hyperparameters from scratch. The assumption is that the knowledge required to solve the source and target problems is similar and transferable. However, the application of transfer learning to design problems is an understudied area where benefits and limitations are unknown. This motivates us to ask the following question: *To what extent is the design knowledge acquired from one problem computationally transferable to another problem in a different context?*

To answer this question, this study develops a computational design agent that can learn design knowledge and mimic the prominent behavioral patterns of human designers. To test the transferability of the knowledge learned by this agent in new design problems, we first introduce a process-level representation of the design action data inspired by the function-behavior-structure model (Gero & Kannengiesser 2014). While design behaviors at the design action level (referred to as the low level) can be significantly dependent on the problem context, the process-level representation introduced in this article is a high-level abstraction that has the potential to represent problem-agnostic behaviors. Therefore, we are particularly interested in transferring knowledge at the design process level for the

sake of model generality and transferability across domains. With such a process-level representation of the design action sequence, we adopt reinforcement learning (RL) and develop RL-based design agents that can mimic human behaviors and are capable of transferring the learned knowledge from one design problem to another.

In this approach, to acquire and incorporate human knowledge into the self-learning capability of RL, we introduce a reward mechanism that utilizes sequential design data from human designers through the first-order Markov chain model. More precisely, we use Temporal Difference (TD) based Q-Learning (Jang *et al.* 2019) to train the RL agent with this reward mechanism based on our previous study (Rahman, Bayrak & Sha 2022). We tested the generality and transferability of the design knowledge over two distinct solar design problems: an energy-plus home design problem and the solarize UARK campus design challenge. We train the agent on the design data collected from the energy-plus home design problem and transfer the design knowledge to the solarize UARK campus design problem. The transferability of the design knowledge is tested by the accuracy of the predicted design actions in the solarize UARK campus data set.

The remainder of the article is organized as follows. In [Section 2](#), we present the relevant research on agent-based design and transferability of design heuristics. [Section 3](#) introduces the technical background of RL (i.e., the Q-learning method) and an overview of the research approach. [Section 4](#) describes the experiments conducted for the collection of human behavioral data in two different design challenges. In that section, we also present the data processing methods and the model formulation for the RL-based design agent, including the model setup and the metrics for evaluating the model performance. In [Section 5](#), the results are presented first and then explained and discussed, from which the insights are generated and summarized. [Section 6](#) concludes the article with a summary of contributions and limitations, as well as potential directions for future work.

2. Literature review

2.1. Agent-based design

Agent-based modeling is a commonly adopted methodology for studying individual design strategies or team-based design. There exist different types of models to build artificial design agents. McComb, Cagan & Kotovsky (2015) developed a Cognitively-Inspired Simulated Annealing Teams (CISAT) modeling framework, an agent-based platform to simulate team-based engineering design. To mimic human search strategies from design crowdsourcing data, artificial agent-based inverse learning methods with Bayesian optimization (BO) have been developed (Sexton & Ren 2017). Chaudhari, Bilonis & Panchal (2020) used Bayesian inference to compare simple heuristic models and expected utility (EU)-based models to identify which model provided the best description of designers' information acquisition decisions. Sha, Kannan & Panchal (2015) developed a normative model that integrates a Weiner process BO with game theory to study designers' sequential decisions under competition. Later, Bayrak & Sha (2020) addressed the same problem by testing a data-driven approach integrating a long short-term memory (LSTM) network with non-cooperative games.

Additionally, there are studies that use deep learning techniques to build design agents. For example, Fuge, Peters & Agogino (2014) developed a collaborative filtering-based system to recommend appropriate design procedures to novice designers. The approach showed a significant improvement over the traditional text-based selection method. In our previous study, we developed a design agent using an LSTM network to predict future design actions based on historical design behavior data. The prediction accuracy was found to outperform traditional Markov models (Rahman, Xie & Sha 2021). Later, the authors further improved its prediction accuracy by combining static data characterizing human attributes and dynamic design action data in the LSTM framework (Rahman *et al.* 2020).

In addition to using text data, Raina *et al.* (2019) developed a two-step deep learning-based design agent based on image data from a truss design problem. In the first step, a convolutional neural network-based auto-encoder maps design images to a low-dimensional embedding to generate a sequence of truss design layouts. In the second step, a rule-based image processing inference algorithm outputs the design operations needed to construct the truss structures in the generated sequence and iteratively improves the design.

These existing studies mainly focus on using historical data to provide design feedback or extracting strategies from all designers to identify average design patterns. In this study, we leverage the power of RL to develop a design agent that mimics human design behaviors and test the transferability of learned knowledge to other design problems.

2.2. Transfer learning in RL and design

Transfer learning in RL can be categorized mainly into three major groups that include parameter transfer, instance transfer, and representation transfer (Lazaric 2012).

In parameter transfer, the target task can use the RL parameters (i.e., initial values or learning rate) according to the source tasks. Parameter transfer is suitable when the source and target tasks share a common state action space (Phillips 2006). Mehta *et al.* (2008) introduced a Variable Reward Hierarchical Reinforcement Learning (VRHRL), a parameter transfer method, which uses previously learned policies to speed up and improve the result. They assume that the reward function is a linear combination of reward weights throughout the Markov Decision Process (MDP). In another study, the Attend, Adapt, and Transfer (A2T) model, a deep RL model, was introduced by Rajendran *et al.* (2015), which can select and transfer the value function from multiple source tasks to the state space of the target task, but in the same problem domain.

In instance transfer, samples of states, actions, and corresponding rewards from different source tasks are used to learn the target task. For example, Sunmola & Wyatt (2006) transferred trajectory samples from the source task and used them in the model of new tasks to simplify the estimation of the model.

In representation transfer, the RL agent learns a representation of the source task and performs an abstraction process to fit it to the target task. In this process, studies have used neural networks for feature abstraction (Duan *et al.* 2016; Zhang, Satija & Pineau 2018), while other strategies, such as the reward-shaping method (Konidaris & Barto 2007), have also been explored. In the reward-shaping method, an intermediate reward function is introduced, which provides initial estimates of

the value function of the new task to the agent after a reasonable amount of training. This is different from using a neural network as a value function estimator to directly learn the values.

Although transfer learning has been used in many applications, it has rarely been studied in design behavior research. Raina *et al.* (2019) proposed an approach to transferring design strategies between similar design problems in the same context. This study used a hidden Markov model for problem representation and the CISAT framework as an agent to learn design strategies. They used this model to transfer the learned design strategy from a home cooling system design problem to a scaled-down and scaled-up version of it. The results indicate that transferring previous experience from the source problem improves the agent performance in the target problem, especially in the initial stages of the design process rather than in the later stages. Another example from Raina, Cagan & McComb (2022) integrated deep policy networks with a tree search algorithm to discover generalizable problem-solving behaviors with computational agents without prior data. Their work showed that their agents can learn high-performing design behaviors for truss and circuit design problems and those behaviors were transferable within the same problem context under different boundary conditions. Whalen & Mueller (2021) presented Graph-based Surrogate Models (GSMs) for trusses and explored transfer learning to enhance their adaptability across different design spaces, resulting in more flexible and data-efficient surrogate models with reduced prediction errors. Behzadi & Ilieş (2021) introduced a novel approach combining transfer learning and generative adversarial networks for topology optimization, enhancing generalization ability and reducing computational costs in design exploration.

Raina *et al.* (2019) presented evidence for design knowledge transferability for scaled problems and Raina *et al.* (2022) showed transferability for different boundary conditions within the same problem context. These studies focused on transferring design behaviors at the lower detailed action level. In this article, we extend the transferability question to significantly different design problems beyond scaled problems. We enable transferability to problems in different contexts by introducing a new high-level design process model to represent design knowledge that can be generalized across problem contexts.

3. Design problem and technical background

3.1. The design problem under investigation

In this study, the designers' behavioral data are collected from two design challenges: the energy-plus home design and the solarize UARK campus design (Figure 1). The reason for choosing these two specific problems is to demonstrate the feasibility of our approach in the controlled experimental setup where we could measure the transfer of design knowledge from one problem to another. Additionally, these two problems are chosen because they have different solutions and the design patterns used to solve them are not identical. We train our design agents using the energy-plus home design dataset. Therefore, it is treated as the source task, and the solarized UARK campus design is used as the target task. It is worth mentioning that this article focuses on design activities within a

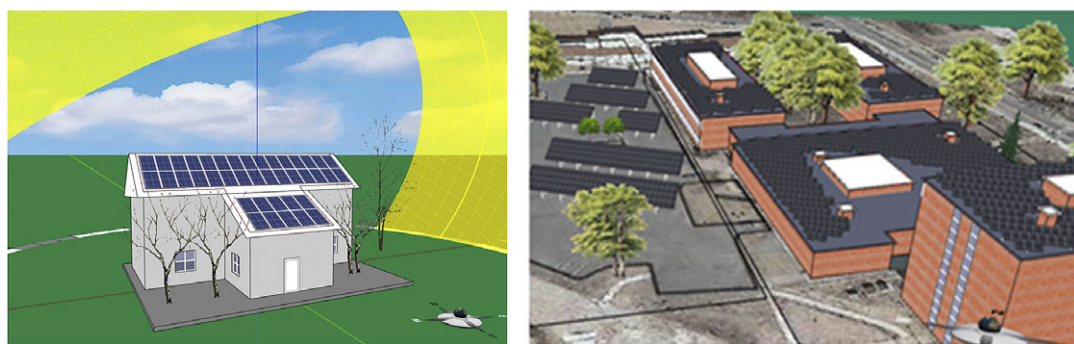


Figure 1. An example of the energy-plus home design problem (left) and an example of the solarize UARK campus design problem (right).

CAD process, but the approach is applicable to any design situation where designers' action data can be recorded and transcribed. For example, it will be very suitable for studying configuration design problems that often involve sequences of design actions. For conceptual design problems, since the decision processes in that stage could be sequential and parallel, our approach is more suitable for iterative and sequential design data. In the following, each of the tasks is described in detail.

In the energy-plus home design problem, participants were asked to build a solar-powered home in Dallas, Texas. The objective is to maximize the annual net energy (ANE) while minimizing the construction cost. The overall budget for this design problem is \$200,000. Furthermore, we set specific design constraints to confine the design space, as summarized in [Table 1](#). This system design problem involves many

Table 1. Design requirements of the design challenges

Design challenges	Design variables	Design constraints
Energy-plus home design	Story	1
	Number of windows	> 4
	Size of windows	>1.44 m ²
	Number of doors	≥ 1
	Size of doors (Width × Height)	> 1.2 m × 2m
	Height of wall	>2.5 m
	Distance between ridge to panel	>0
Solarize UARK campus design	Solar panel model	Choose 1 from 3 options
	Solar panel height	≥ 3.5 m
	Solar panel width	5.25 m–6 m
	Panel Placement (Overall)	Panel edges must not overlap
	Panel Placement (parking lot)	≥ 7 m from the nearest panel

design variables with complex coupling relationships among these variables (e.g., designers may want to add many solar panels for higher ANE, however, the distance between solar panels cannot be too small, so there is a limit for the number of solar panels to be placed). For this reason, the design space is large and different designers may have different strategies to explore and exploit the design space.

In contrast to the energy-plus home design problem, the solarize UARK campus design is more open-ended. In this problem, participants were provided with a computer-aided design (CAD) model of a student housing complex and its adjacent parking lot on a university campus and asked to use the open space on the roof of the buildings and the paved area on the parking lot to design the solar system. The goals of the design challenge are threefold. First, the annual energy output should be greater than 1,000,000 KWh. Second, the overall budget should not exceed \$1,900,000. Finally, the payback period should be less than 10 years. Participants are encouraged to work iteratively and record the performance of different solutions they explored, so that they can compare their own design iterations to continuously improve the performance of their designs. To achieve the design goals, designers needed to carefully control the design variables, including the location, length, tilt angle, and model of each solar panel considering the dependencies among them. Therefore, participants would benefit from the holistic perspective of systems thinking. For example, the optimal tilt angle of a solar panel depends on the height and where it will be placed. The degree to which designers could manipulate each variable is limited by a set of constraints, as shown in [Table 1](#).

Both design problems are carried out using Energy3D, a CAD software for renewable energy systems (Xie *et al.* 2018; Rahman *et al.* 2019). Energy3D collects design data in a non-intrusive way. The non-intrusive data collection process can reduce the cognitive bias during an experiment. Energy3D logs design data at a fine-grained level. In particular, it logs every design action performed and collects design files (including all artifacts) every 20 seconds. Therefore, the data collected from Energy3D fully capture what designers do (i.e., design actions) throughout the design process. Energy3D collects the design process data in JSON format, which records time stamps, design actions, design artifacts, and simulation results. On average, a participant has about 1500 lines of design process data. An example of two lines of the design actions log is presented in the text box below.

```
{“Timestamp”: “2020-05-23 08:17:38”, “File”: “Design-Contest.ng3”, “Edit Rack”:
{“Type”: “Rack”, “Building”: 2, “ID”: 485, “Coordinates”: [{“x”: 1.496, “y”: 47.053,
“z”: 54.7}]}}
```

```
{“Timestamp”: “2020-05-23 08:19:49”, “File”: “Design-Contest.ng3”,
“PvAnnualAnalysis”: {“Months”: 12, “Panel”: “All”, “Solar”: {“Monthly”:
[892.25,1060.33,1478.38,1544.75,1819.32,1950.18,2048.8,
1876.89,1423.77,1241.81,794.41,697.7], “Total”: 511869.84}}}
```

In this study, we extract only design actions related to design objectives, such as “Add wall,” “Edit wall,” “Edit roof,” “Show sun path,” and so on. We ignore design actions that have no effect on design outcomes, such as “Camera” and “Add tree.” This post-processing leads to 115 unique design actions in the energy-plus home design problem and 106 unique design actions in the solarize UARK campus design problem.

3.2. Technical preliminaries

Typical RL approaches rely on the formulation of a Markov Decision Process (MDP) to learn optimal behaviors in sequential decision-making problems. The goal of an MDP is to find the optimal policy for decision-making based on a pre-defined reward mechanism. Q-learning helps to find such a policy by generating a Q-value for each state-action pair that is used to determine the best decision for a given state of a problem environment. The Q-values of all state-action pairs are typically stored in a Q-table that is learned through a pre-defined number of iterations using the epsilon-greedy algorithm (Sutton & Barto 2018). Once the agent selects an action, it reaches a new state S' . In the new state, the agent selects the best possible action that yields the maximum Q-value and obtains the corresponding reward from the environment. Based on the reward values, the Q-table is updated using a temporal difference formulation (Jang *et al.* 2019) according to the following equation:

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha(R(s, a) + \gamma \max\{Q_t(s', a')\} - Q_t(s, a)), \quad (1)$$

where Q returns the expected cumulative reward of performing an action in a state (Sutton & Barto 2018). $Q_{t+1}(s, a)$ is the new Q-value for state s and action a in the next iteration $t + 1$. $Q_t(s, a)$ is the current Q-values. α is the learning rate, a hyperparameter that defines how much new information can be accepted in the current iteration versus the old information from previous iterations. When α is close to zero, Q-values are never updated, whereas an α value close to 1 means that the learning process occurs quickly. $R(s, a)$ is the value of the reward for taking action a in the state s . $\max\{Q_t(s', a')\}$ is an estimate of the maximum future reward value, and γ is the discount factor that controls how much these future rewards will be taken into account when updating the Q-values. This approach balances the importance of immediate and future rewards.

In this article, we adopt a probabilistic model for the action transition, known as the noisy rational decision model, where the RL agent chooses one of the actions with the following probability function based on the Q-values (Wu *et al.* 2021) to model noise in agent decisions,

$$\Pr(a|s) = \frac{\exp(\theta \cdot Q(s, a))}{\sum_{a_i \in A_i} \exp(\theta \cdot Q(s, a_i))}, \quad (2)$$

where A_i denotes the action space of an agent. The equation takes values from the Q-table and provides a probability of taking each possible action a in a given state s . The hyperparameter $\theta \in [0, \text{inf})$ determines the decision-making strategy of an agent. When θ is zero, the equation provides a uniform distribution (i.e., all design actions are equally likely to be selected with the probability of $\frac{1}{\text{dim}(A_i)}$). When θ goes to infinity, the probability of the action with the highest Q-value (e.g., the most frequently occurring design action at a given state) approaches 1. We use this θ to control how much to reinforce (or exploit) a commonly seen action pair in the data and how much to explore alternative action pairs. Note that this model is similar to the logit choice model commonly used in the design and marketing literature (Gensch & Recker 1979) where Q-values correspond to the utility of discrete choices.

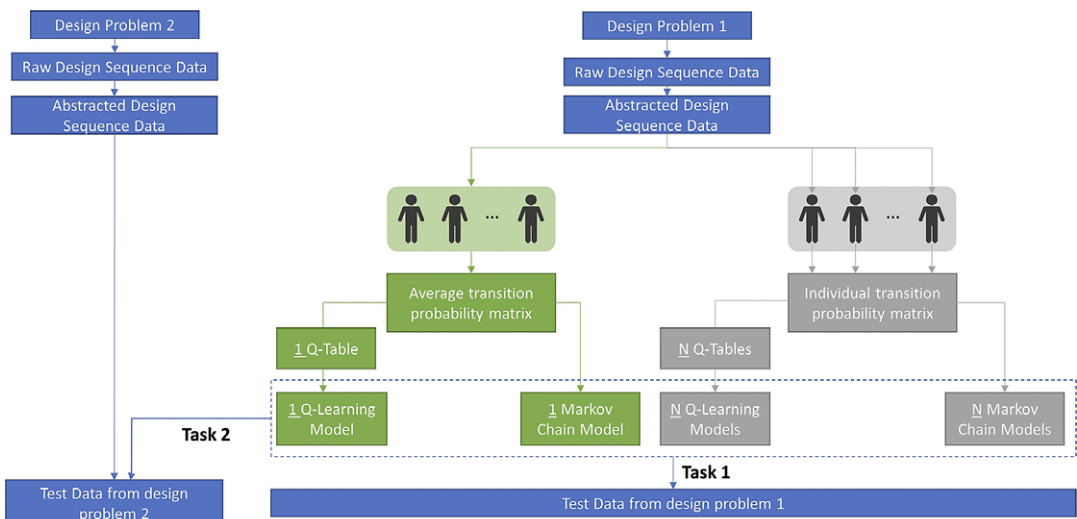
4. Research overview

4.1. Research approach

This study consists of two tasks, as shown in Figure 2. The first task is to develop an RL-based design agent to mimic the sequential decision-making behaviors of human designers. The data used to train the agent are obtained from a series of actions performed by the designers. The actions could be adding a component, deleting a component, or changing the parameters of a component. To evaluate the performance of the agent, we conducted a comparative study using the first-order Markov chain model as the baseline.

Once sequential design data are collected, a design process model is applied to convert each design action to its corresponding design process stage. A design process model at the ontological level captures the context-independent essence of design thinking regardless of the particular design action involved. Therefore, such a higher-level abstraction in problem representation helps generalize design knowledge and facilitate the transfer of design knowledge from one problem to another. Moreover, by applying the design process model to group similar design actions, the procedure turns out to be a dimension reduction that helps improve the computational efficiency. This is particularly useful in system design, where there could be a large number and a variety of actions involved.

After mapping the detailed sequential data into our high-level design process representation, we calculate transition probabilities for each state-action pair based on the first-order Markov chain model and use these transition probabilities for the reward table. There are two different ways to obtain the transition probability matrices when creating the reward table. One way is to use the average transition probability matrix that aggregates the sequential design data of N subjects (designers). In this situation, one Q-learning model will be



Task 1: The comparative study on Q-Learning and Markov Chain models
Task 2: The test on the transferability of design knowledge

Figure 2. The overview of the research tasks.

developed to predict the behaviors of N designers. The other way is to use each individual's transition probability matrix to construct N Q-learning models that can be used to predict the design actions of N designers separately. In both ways, we tune the hyperparameter θ and investigate how it influences the accuracy of the Q-learning model in predicting the sequential actions of each individual designer. Based on these configurations, we are interested in knowing which way is a better way to construct the reward table for the RL-based agent. To evaluate the performance of RL-based agents in prediction, we compare them with those without a reinforcement mechanism, that is, the models purely based on MC analysis. The results of these comparative studies are presented in [Section 5](#).

Our second task is to test the transferability of learned design knowledge between design problems. In particular, we apply the Q-tables learned from the source design problem (Design Problem 1) to predict the designers' behaviors in the target design problem (Design Problem 2).

4.2. Problem representation and RL model

We define the RL elements, that is, states, actions and rewards, in the context of the design problem below:

States describe the current situation in which the agent interacts with the environment. In this study, since our goal is to mimic human design behaviors, we define the state in RL as the state of a designer's thought process in design. Various ontological models have been proposed to represent design processes and interpret design thinking (Gero & Kannengiesser 2014). In this study, the proposed state representation model is inspired by the function-behavior-structure (FBS) design process model. The FBS model is a design ontology that has been widely used to represent a variety of design problems independent of the application context. The FBS model was later extended to design processes in CAD environments with additional sub-processes (Kannengiesser & Gero 2009).

Inspired by this CAD version of FBS, we define six states of design thinking in Energy3D: Formulation, Reformulation, Synthesis, Interpretation, Evaluation, and Analysis. These states are treated as the states in RL. In the design data collected from two challenges, we observe that designers can transition between any pair of states when using Energy3D. Therefore, we use a fully connected graph ([Figure 3](#)) as a state representation for the proposed RL model. Note that the primary focus of this article is to study whether problem-independent design knowledge can be transferred at a high-level abstraction using a representation rather than validating the capability of the FBS model. Alternative representations are possible, and we select the FBS model based on its applicability to CAD design problems. Finding the best representation for knowledge transfer is beyond the scope of this study.

Actions in our RL problem are the actions (e.g., CAD operations in Energy3D) performed by the designers. In this study, we combine similar design actions, such as adding a wall and adding solar panels, into one category (see [Table 2](#)). This mapping is done manually as problem representation is commonly a manual effort in data science. There are several benefits in doing the mapping. First, abstracting actions into higher-level categories captures the context-independent essence of design thinking, and thus improves generalizability. Second, these categories significantly decrease the number of possible state-

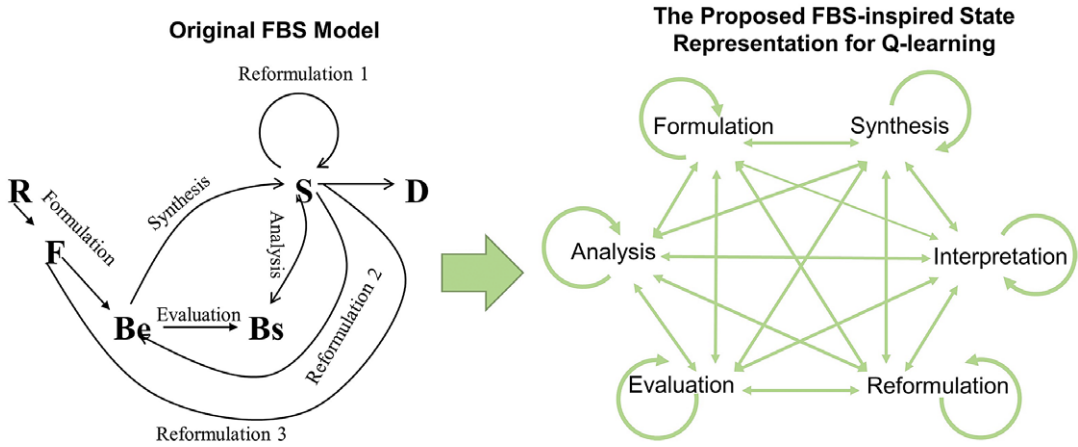


Figure 3. The FBS design process model (Gero 1990) and the design thinking states are defined in the proposed reinforcement learning model.

Table 2. Design action categories and their corresponding actions

Design action category	Action category	Example
Addition of any components	Add	Add wall, Add solar panel, etc.
Edit of any components	Edit	Edit door, Edit wall, etc.
Environmental check	Show	Show Helidon, Show sun path, etc.
Evaluation of cost	Cost	Cost
Removal of any components	Remove	Remove window, Remove the roof, etc.
Analysis of annual net energy	Analysis	Energy Annual Analysis

action pairs and reduce the computational burden during the training of RL agents. Finally, in a previous pilot study on human-AI interactions (Rahman *et al.* 2020), where a trained deep learning model was used to recommend design actions to designers, we observed that designers felt interrupted if they received specific CAD operations. Instead, they would prefer to receive more general guidance at a higher level of instruction. Therefore, grouping design actions into categories may better serve the development of RL-based design agents for the future of the human-AI partnership in design.

Reward is the feedback from the environment in response to a particular action. An RL agent aims to maximize the total reward calculated by summing all instantaneous rewards. However, this sum can potentially grow indefinitely. Therefore, a discount factor (γ) is included in the reward function to reduce the contribution of future rewards. A typical reward function is expressed as follows:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \tag{3}$$

where r_t is the instantaneous reward at time t . Typical RL models are self-learning methods that use a reward from the environment. As our target is to build an agent

that mimics human designers, we use the data containing the designers' actions (e.g., those shown in Table 2) to formulate a reward table. Specifically, we employ the first-order Markov chain model's transition probability matrix of the action sequences to construct the reward table. The transition probabilities can be the values of an individual participant's transition probability matrix or the average values of all the participants' transition probability matrices, characterizing the aggregated one-step-ahead sequential behaviors of all designers. Figure 4 shows the average transition probability matrix of participants in the source design task. This Markov chain-based reward mechanism reinforces the action pairs that occur more frequently in the training process. Combining this data-driven reward with the self-learning ability of RL is a uniqueness of our model.

4.3. Model setup and evaluation

Markov chain agent: We choose the first-order Markov chain model as our baseline model to compare with the Q-learning agent, as it has been widely used for sequential learning of design behaviors in the existing literature (Kan & Gero 2009; Yu *et al.* 2015; McComb, Cagan & Kotovsky 2017). First, we compute the transition probability matrix for each designer in the dataset and then apply the leave-one-out cross-validation method for the prediction. This means that we aggregate the matrices of the $(n - 1)$ designers and use the resulting aggregated transition probability matrix to predict the sequence of the n th designer. By iterating this process, we obtain the prediction accuracy for all designers and report the average as the final prediction accuracy. The average prediction accuracy for the baseline model of the source design problem is 41%.

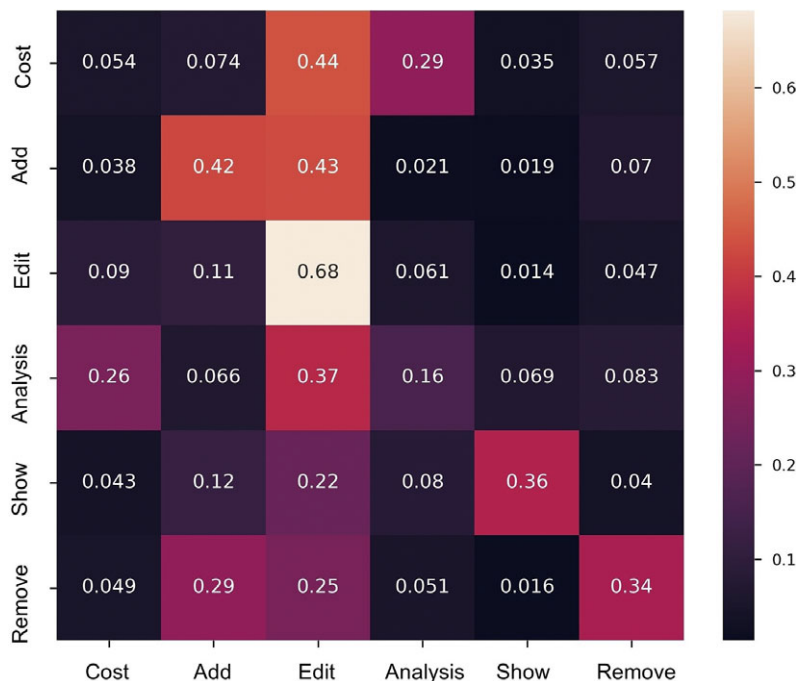


Figure 4. Average transition probability matrix of participants.

Q-learning agent: We use the k -fold cross-validation technique to evaluate the Q-learning agent. So, k rounds of training and testing are performed so that in each iteration, $(k - 1)$ partitions are used to obtain the reward table and train the Q-learning agent. The remaining partition is used to test the Q-learning agent. In this study, we chose $k = 11$ with a trade-off between maximizing the training dataset and minimizing total training cycles.

To initiate the training process, a 6×6 Q table is initialized, where the rows represent six states (see Figure 3) and the columns represent six actions (see Table 2). Initially, all values in the Q table are set to zero. As the iterations progress, the Q table undergoes updates. This entails that after taking a specific action, the designer transitions to a new state, and the corresponding value in the Q table is updated based on the reward table. In this way, the Q table dynamically evolves through the training iterations, capturing the learned knowledge and guiding future decision-making. We determine the optimal settings for the hyperparameters (α and γ) of the RL model using trial-and-error by testing values in $[0.1, 0.9]$ for both parameters, and train the Q-learning agent based on the best combination found. The final parameter values are learning rate $\alpha = 0.3$ and discount factor $\gamma = 0.6$ (see equation (1)). Similarly, the value of θ in equation (2) is tuned to achieve maximum accuracy. The Q-table is updated by every iteration and the training process takes 10,000 iterations.

Table 3 presents the learned Q table, which illustrates the transitions from state to action. The Q values in the table indicate that the transition to the *Edit* action generally has higher values compared to other actions. For instance, the transition from the *Formulation* state to the *Edit* action has the highest Q-value of 31.97. However, it is important to note that there are some cases where transitioning to *Edit* does not have the highest Q-value. For example, the transition from *Reformation* to *Add* achieves the highest Q-value, which is 24.19.

Metric for prediction accuracy: In this study, the agent only predicts the next action (i.e., the one-step look-ahead decision) based on the designer's action in the last step. As the agent chooses an action from a probability distribution, the prediction can vary from one iteration to another. Therefore, to account for the stochastic nature, we run a total of 50 realizations for each prediction. For each action sequence, we compare the predicted actions with the actual decisions and count the number of correctly predicted actions. The prediction accuracy is then obtained by dividing that number by the length of the action sequence. Finally, we

Table 3. The learned Q table in the reinforcement learning model

State/Action	Add	Analysis	Show	Remove	Cost	Edit
Formulation	29.69	9.96	7.77	10.98	11.34	31.97
Analysis	12.03	16.92	10.37	10.72	22.30	29.36
Interpretation	14.78	12.82	22.87	9.32	11.59	22.30
Reformation	24.19	11.27	7.70	22.54	12.16	23.84
Evaluation	13.01	20.74	8.75	10.93	12.55	32.72
Synthesis	14.84	11.87	7.47	9.61	13.34	44.57

take the average of the 50 predictions to report the final accuracy, as shown in equation (4).

$$\text{Prediction accuracy} = \frac{1}{50} \sum_{i=1}^{50} \left(\frac{n_i^c}{l} \right), \quad (4)$$

where n^c is the number of correctly predicted actions and l is the length of a designer's action sequence. We use this metric to evaluate the transferability of design knowledge between problems. The accuracy of a model trained with the source problem data using the proposed high-level problem representation to predict the actions in the target problem is used as the indicator of transferability. The benchmark for prediction assessment is set using a model trained with the target problem data.

5. Result and discussion

In this section, we first present a detailed description of the design experiments. Subsequently, we present the results of the two research tasks as illustrated in Figure 2. The first task aims to test the design knowledge transferability using a Q-learning model trained by energy-plus home design to predict the participants' sequential decisions in the solarized UARK campus design challenge. In the second task, we show the in-depth training procedure and present the capability of the model to learn useful design patterns and predict sequential design decisions using the energy-plus home design data. For that purpose, we discuss the results obtained from the Q-learning model trained using a reward table formulated by the average transition probability of participants in the energy-plus home design challenge. We also compare it with the Q-learning models trained using the reward table formulated by each individual transition probability matrix.

5.1. Experimental setup

Both of the design experiments are carried out as a form of design challenge, as it motivates the participants to explore the design space and improve their solution as much as possible. Additionally, designers are incentivized by monetary reward which relates to the quality of their designs. The energy-plus home design problem is conducted in a classroom setting and consists of three phases: pre-session, in-session and after-session. In the pre-session, the designers get familiar with the Energy3D environment, the design problem, and basic solar science concepts. The instruction and guidance provided in the pre-session help minimize the potential bias due to the differences in participants' pre-knowledge and learning curves. The pre-session lasts about 30 minutes while the in-session lasts about 90 minutes. During the in-session, participants perform the design task according to the design requirements. The after-session, which lasts about 10 minutes, is for participants to claim rewards and sign out of the challenge.

The Solarize UARK Campus design problem is conducted in a virtual setting due to COVID-19. At the beginning of the design challenge, participants receive all the necessary information and a tutorial session through an introductory presentation. The participants are then given seven full days to complete the design challenge at their own pace.

In the energy-plus home design problem, a total of 52 students from the University of Arkansas participated in the design challenge. Among them, 29 students were undergraduate students and 23 were graduate students. 48 of them were male while four were female. These designers are indexed according to their registered sessions and laptop numbers. Sessions are indexed by letters from A to G, and laptops are indexed with Arabic numerals. For example, A02 indicates that the participant joined session A and worked on laptop number 2. The total length of the design sequence for all designers in this design challenge is 17,510. The maximum length design sequence among these 52 participants is 629 and the minimum is 89.

In the Solarize UARK Campus design problem, we obtained design data from 45 participants, including 36 undergraduate and 9 graduate students. Among the participants, 40 of them were male and five were female. The participants were indexed by their corresponding flash drive number in which the instruction for design challenge was provided. The total length of the design sequence for all designers is 102,855. The maximum length of the design sequence among these designers is 7231 and the minimum is 414. At the end of the design, the participants return their flash drives, which recorded both their design behavior data and the design work files. For consistency, we simply refer to the student designers in both design problems as designers in the remainder of this article.

5.2. The results on the transferability of design knowledge

We present the capabilities and performance of the Q-learning model to capture high-level design patterns in the energy-plus home design dataset (source) in [Section 5.3](#). In this section, we focus on the transferability of the design knowledge learned in the source problem to the target problem of the solarize UARK campus design challenge. To this end, we use the Q-learning model trained in the source problem to predict the designers' design sequence in the target problem. Using these problems, we also test the design transferability using the Markov chain model for reference. In the Q-learning model, the "design knowledge" transferred is the Q-table learned from the source design problem; while in the Markov chain model, the "design knowledge" transferred is the aggregated transition probability matrix. Note that both models use the higher-level abstraction introduced in this article to improve model generalizability.

Both models are compared with the baseline Markov chain model trained from the solarize UARK campus design without transferring any "design knowledge." The baseline model is trained with all the designers' design sequence data except for the highest or lowest ten designers (i.e., the test dataset). Similarly, we choose the top ten highest and lowest performing designers to compare the models' performance.

[Figure 5](#) shows the prediction accuracy of the transferred Q-learning and Markov chain models along with the baseline Markov chain model for the high-performing group. The result shows that the transferred Q-learning model achieves the highest performance in predicting the sequential decisions of the ten designers among the three models. The highest prediction accuracy is 0.78, achieved when predicting designers 114 and 44. However, the prediction accuracy of the transferred Markov chain model is lower than the baseline Markov chain model for the ten designers. The average prediction accuracy of the baseline

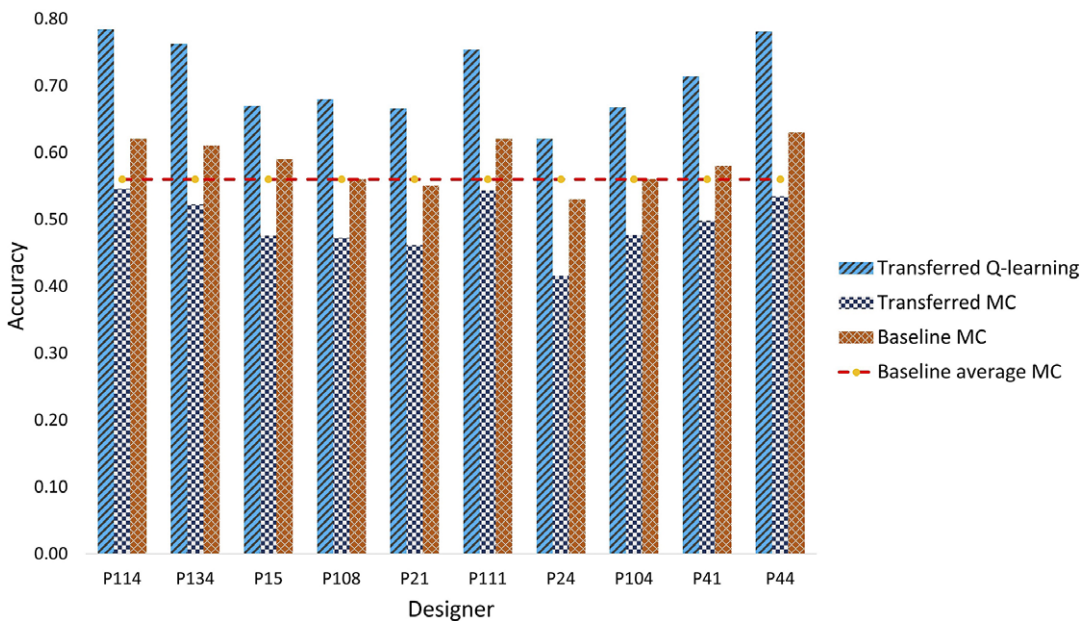


Figure 5. The prediction accuracy of the transferred Q-learning, Markov chain and the baseline Markov chain model for the high-performance design.

model is 0.56 illustrated by the dotted line in Figure 5. This result indicates that the frequently occurring design patterns in the source and target problems are similar, and reinforcing these patterns from the source problem also improves the prediction accuracy in the target problem for high-performing designers. This finding suggests that the Q-learning model can transfer the high-level problem-agnostic knowledge between the two datasets better than the Markov chain model.

Similar results are also observed in the low-performing group, as shown in Figure 6. However, the overall performance of the transferred Q-learning model in the low-performing group is lower than that in the high-performing group. For example, among the ten cases, only five instances have a prediction accuracy higher than the average prediction accuracy of the baseline MC model, much lower than that of the high-performing group. This result is expected since the accuracy of the Q-learning model with the average reward model on the source task is lower in the low-performing group than in the low-performing group on average, as discussed below in Figure 7. Since higher accuracy indicates a better capacity of the model in capturing design behavioral patterns and embedded knowledge, both Figures 5 and 6 show that the MC model is not an ideal model to test the transferability of design knowledge in CAD-based design activities.

Finally, we can use the Q-learning model to answer the research question: “To what extent is the design knowledge acquired from one problem computationally transferable to another problem in a different context?” In Figure 7, we show the correlation between the accuracy of the transferred Q-learning model and the baseline Markov chain model in the target task for both the high-performing and low-performing groups. The results show a strong correlation for both groups,

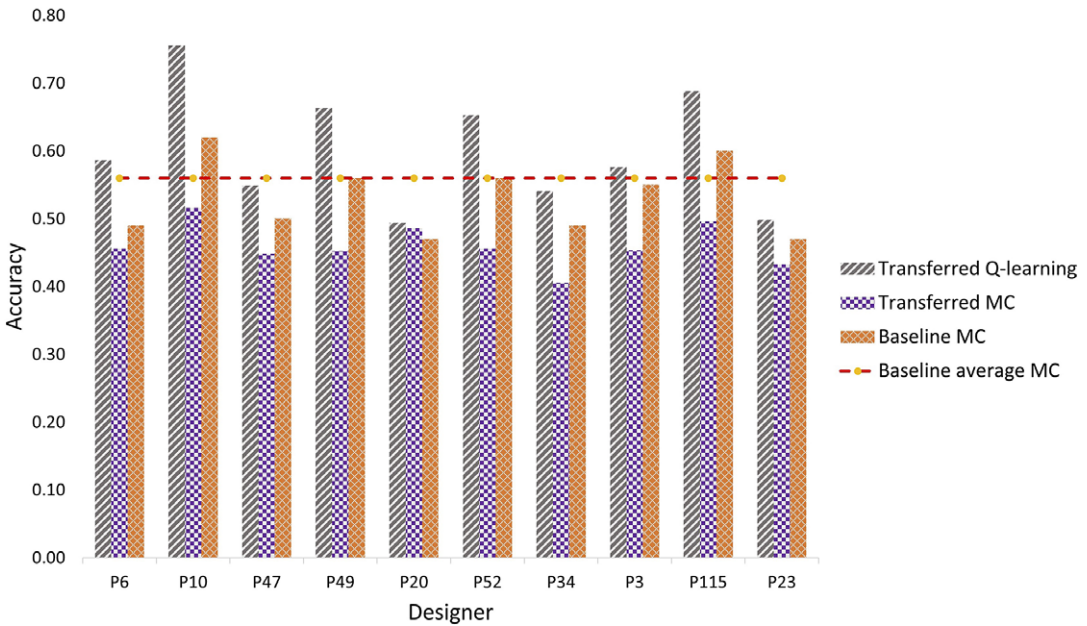


Figure 6. The prediction accuracy of the transferred Q-learning, Markov chain and the baseline Markov chain model for the low-performance design.

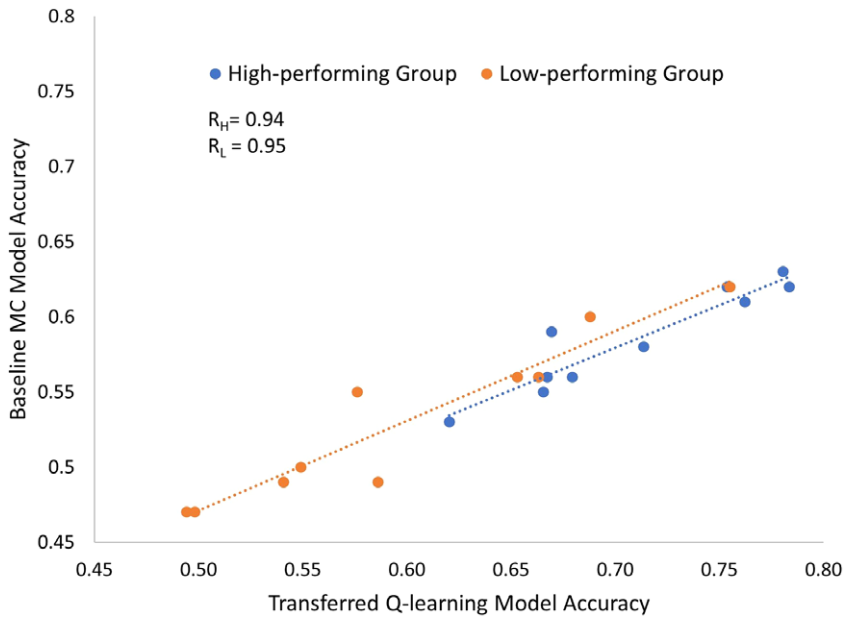


Figure 7. The correlation between the transferred Q-learning model and the baseline Markov chain model for the high-performing group (blue) and the low-performing group (orange).

where the correlation coefficients for the high-performing and the low-performing groups are 0.94 and 0.95, respectively. This high correlation in both groups suggests high transferability. This result indicates that the amount of design pattern that the MC model can capture from the baseline model in the target problem is directly proportional to the amount that the Q learning model can transfer from the source problem. In other words, if the ability of the baseline model to predict the behavior of a participant is high, then the ability of the transferred Q-learning model to predict their behavior is also high. If the accuracy of the baseline model to predict the behavior of a designer in the target task is low, it suggests that this designer weakly follows distinct patterns to be learned using a model. Therefore, the accuracy of the transferred Q-learning model is also proportionally low for those participants. This finding does not mean that the transferability of the Q-learning model is low, but rather the Q-learning model trained in the source task can only capture design patterns that can be captured with a model trained in the target task. This finding is reasonable since machine learning models can only work if there is a pattern to be learned.

5.3. The prediction results of the Q-learning model using the energy-plus home design dataset

5.3.1. Results

Figure 8 shows an example of the prediction accuracy for the individuals, F15, G05, G07, C03, C05, and C07. It shows that the prediction accuracy increases as θ increases from 0 to 0.25, after which the accuracy does not increase significantly and saturates to its final value for all the design sequences tested. Among all designers, G05 achieves the highest prediction accuracy of 73%, higher than the baseline of 41% achieved by the Markov chain model. We also observe that, for some designers, the prediction accuracy obtained at the maximum θ tested (i.e., when $\theta = 0.29$) is even lower than that of the baseline model, and this observation was found in the tests of other folds. Since θ controls the effect of reinforcement (as θ increases, the probability of the occurrence of reinforced action pairs increases), these results suggest that reinforcing certain behavioral patterns

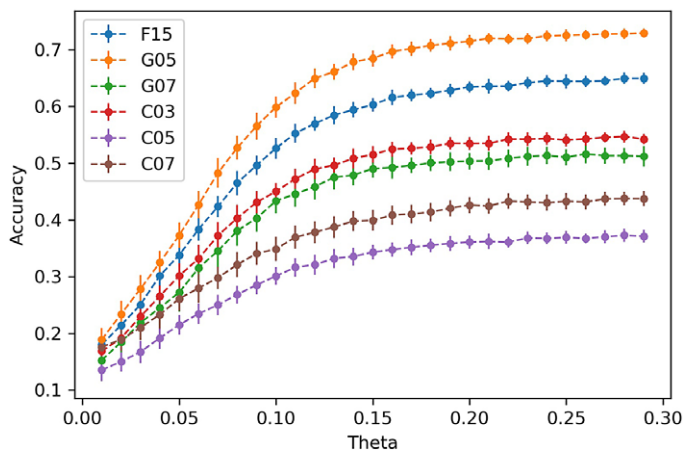


Figure 8. Prediction accuracy as a function of θ value in equation (2).

does not necessarily improve prediction accuracy. This is because if a designer's data do not (or weakly) show the patterns that are reinforced, using the RL agent will generate more wrong hits and result in a low predict accuracy.

These results also inspire us to investigate the relationship between prediction accuracy and the designers' performance. In this study, the design performance is measured in the following equation in the unit of KWh/dollar.

$$\text{Design performance} = \frac{\text{Budget} \times \text{ANE}}{\text{Cost}} \tag{5}$$

It should be noted that the Budget in this context is a predetermined constraint, while the cost can be adjusted according to the preferences of the designers. We choose the ten highest and ten lowest-performing designers and compare their prediction accuracy. In each group, the RL agent is trained based on data outside of the investigated group. For example, when studying the ten designers in the high-performing group, we train the RL agent using the remaining 42 designers and test the agent on these ten designers. Figure 9 shows the prediction accuracy for both the high- and low-performing groups. In the high-performing group, the model yields a high prediction accuracy (> 41% generated by the baseline model) in nine out of ten designers. In the low-performing group, the model only produces high accuracy in four designers. It is worth mentioning that the model produces the highest prediction accuracy for designer G05, who achieves the highest performance among all designers. Figure 9c shows the relationship between the Q-learning prediction accuracy and performance for all designers. While the regression

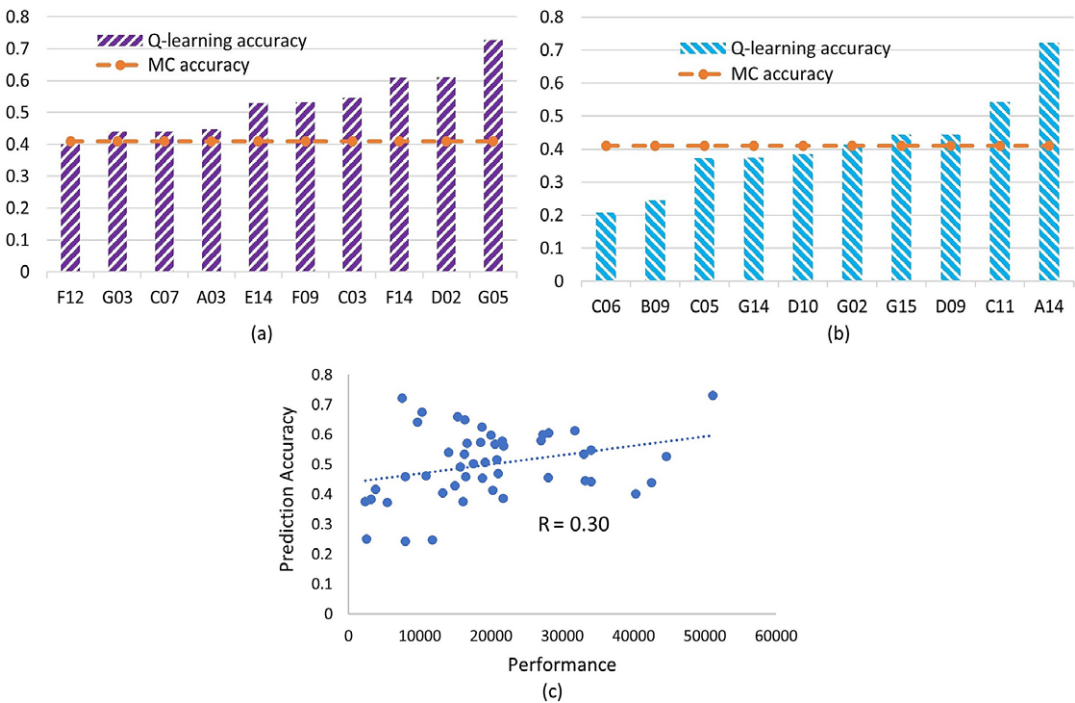


Figure 9. (a) Prediction accuracy of the high-performing design group. (b) Prediction accuracy of the low-performing design group. (c) Relationship between prediction accuracy and performance.

Table 4. The results of the one-sided paired *t*-test for the comparison of prediction accuracy between groups with three different configurations

High performance–Low performance	Average reward–Individual reward	Markov chain agent–Q-learning agent
High Q Avg–Low Q Avg <i>p</i>-value: 0.03	High Q Avg–High Q Ind <i>p</i>-value: 0.025	High Q Avg–High MC Avg <i>p</i>-value: 5.09E–05
High MC Avg–Low MC Avg <i>p</i>-value: 0.02	High MC Avg–High MC Ind <i>p</i> -value: 0.49	Low Q Avg–Low MC Avg <i>p</i> -value: 0.19
High Q Ind–Low Q Ind <i>p</i> -value: 0.25	Low Q Avg–Low Q Ind <i>p</i> -value: 0.29	High Q Ind–High MC Ind <i>p</i> -value: 0.09
High MC Ind–Low MC Ind <i>p</i>-value: 0.013	Low MC Avg–Low MC Ind <i>p</i>-value: 0.0018	Low Q Ind–Low MC Ind <i>p</i> -value: 0.98

Note: “High” and “low” indicate the high-performing and the low-performing groups. “Avg” and “Ind” indicate the average and individual reward formulations. “Q” and “MC” indicate the Q-learning agent and the Markov chain agent. So, “High Q Avg” indicates the prediction accuracy of the Q-learning model with an average reward formulation for the high-performing group.

analysis shows a correlation coefficient of 0.3, a weak correlation between design performance and prediction accuracy in trend, there is a noticeable (and statistically significant) difference in model accuracy between the low-performing and high-performing groups. The impact of participant performance is discussed later in Table 4 with *t*-tests. Including the participants in between clouds this pattern and hence results in a weak correlation.

5.3.2. Discussion

To understand the participants’ design strategies, we compare the transition probability matrix of the designers with the highest (G05) and lowest performance (F12) in the high-performing group. Figure 10 shows the transition probability matrices in the heat maps. The bigger circles indicate a higher transition probability, while the smaller circles indicate a lower transition probability.

In the Markov chain analysis of the high-performing group, it can be seen that the designer (G05) with the highest performance likes to repeat a few particular action pairs, while the designer (F12) with the lowest performance uses a more uniform distribution of the transition probabilities. For example, the top three highest transition probabilities of G05 are “Edit-Edit” (0.82), “Edit-Add” (0.71), and “Edit-Analysis” (0.58), while these probabilities for F12 are 0.57, 0.46, and 0.25, respectively. Meanwhile, G05 never used the action pairs of “Analysis-Analysis,” “Cost-Remove,” and “Remove-Show,” but these behavioral patterns were found in F12’s design sequence. Therefore, although both designers achieve satisfactory design performance (i.e., performance for G05 and F12 is 51137 and 40285, respectively) since they were both in the high-performing group, G05 finished the design task by focusing on exploiting a few specific action pairs, while F12 explored many different operations in Energy3D for the design objective. This also explains why the RL agent produces a higher prediction accuracy when predicting G05’s action sequence because the prominent design patterns were able to be learned and reinforced by the agent.

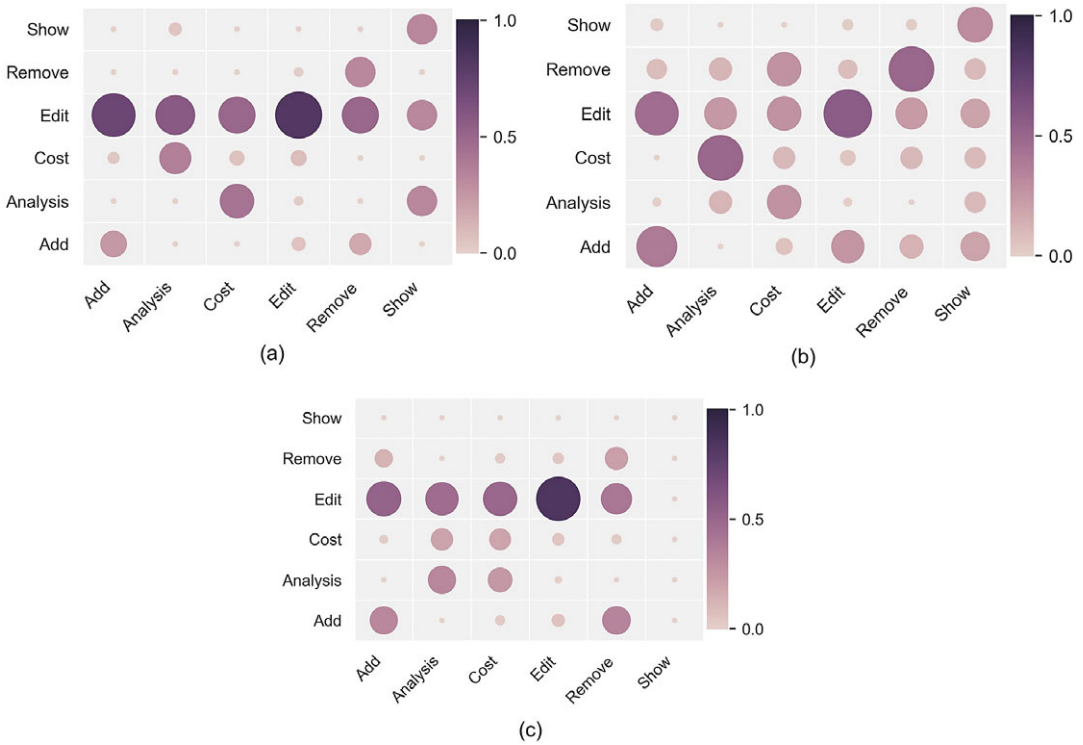


Figure 10. The heat maps of the transition probability matrix of (a) G05, (b) F12, and (c) A14.

In the low-performing group, most designers’ transition probabilities are more uniformly distributed, indicating that they had tried a variety of operations in Energy3D to explore the design space, yet those explorations may not necessarily be beneficial to the design objective. When using the RL agent to predict these designers’ action sequences, it produces the highest prediction accuracy for A14. [Figure 10c](#) shows the transition probability matrix of A14. Similarly to G05, specific design patterns were found in A14’s design process. However, the design performance achieved by A14 (7524 KWh/dollar) is way less than that of G05 (51137 KWh/dollar). This may be attributed to the fact that A14 used many redundant CAD operations. For example, the transition probability of using “Analysis-Analysis” for this designer is 0.33, higher than the average probability of the matrix 0.17, but it is redundant and not helpful. This is because once “Analysis” (analyzing the ANE of the current design) is performed, it should be unnecessary to analyze the design again, as the ANE information should have been acquired already.

Furthermore, A14 did not use the “Show” action at all, which indicates that this designer is not interacting with the CAD environment and is not active in learning the solar science concepts underpinning the design problem. This result indicates that even if the RL agent may predict well for both high-performing and low-performing designers, their design strategies could be different albeit consistent throughout the process and therefore lead to very different design performance. This is also in congruence with the findings in the literature. Burnap *et al.* (2015) showed that both experts and consistently wrong non-experts can present such

behaviors. However, it is worth noting that our models do not rely only on consistency but also on the previous design action data to predict future actions.

To systematically evaluate the performance of the Q-learning model, we compare it with a baseline model, that is, the Markov chain model. In this comparative study, we also tested two different formulations of the Q-table. One is obtained using the average transition probability matrix by aggregating all the designers' action sequences in the training dataset with the Markov chain analysis. The other one is obtained from each individual designer's transition probability matrix. Furthermore, this comparison was made between the high- and low-performing groups. Therefore, in total, 12 comparisons were conducted using the one-sided paired *t*-test. Table 4 shows the 12 tests and their corresponding *p*-values.

The first column shows the comparison of the prediction accuracy between high- and low-performing groups with different models and under different settings for the transition probability matrix. For example, to compare the prediction accuracy of the Q-learning agent using a reward formulation of the average transition probabilities in the high-performing group versus the low-performing group, the null hypothesis (H_0) is that the accuracy for both groups is not significantly different, while the alternative hypothesis (H_a) is that the former is significantly higher than the latter. With a level of significance of 0.05, the *p*-value (0.03) in the table indicates that the Q-learning agent trained from the reward formulation using average transition probabilities predicts more accurately in the high-performing group than in the low-performing group.

The *p*-values highlighted in bold in Table 4 indicate the tests that show statistical significance. The following conclusions in three categories are supported by the *t*-test results.

- *High-performing group versus low-performing group*: Using the Q-learning with the average reward formulation, the prediction accuracy achieved in the high-performing group is higher than in the low-performing group.
- *High-performing group versus low-performing group*: Using the MC model, the prediction accuracy achieved in the high-performing group is higher than that of the low-performing group, regardless of whether it is an average or individual reward formulation.
- *Rewards formulated by average versus individual transition probability matrix*: In the high-performing group, the predictive performance of the Q-learning model with the average reward formulation is better than with the individual reward formulation.
- *Rewards formulated by average versus individual transition probability matrix*: In the low-performing group, the predictive performance of the Markov chain model with the average reward formulation is better than that with the individual reward formulation.
- *Markov-chain agent versus Q-learning agent*: In the high-performing group, using the average reward formulation, the predictive performance of the Q-learning model is better than that of the Markov chain model.

These findings can be explained intuitively. For example, the accuracy of Q-learning using an average (aggregated) reward model is higher in the high-performing group compared to the low-performing group. Frequently occurring action pairs in the aggregated reward model may not necessarily be useful to improve accuracy, especially in the low-performing group, where designers may or

may not follow a particular pattern. Therefore, reinforcing these common patterns does not help with predicting the actions in that group. On the other hand, the high-performing group tends to follow a more consistent set of design strategies throughout.

In the same way, we can also explain why using an average transition probability matrix that aggregates designers' sequential design behaviors (equivalent to reinforcing the prominent patterns) improves a model's predictive performance in the high-performing group. In the average reward mechanism, all the prominent design strategies from the entire group are reinforced, while in the individual reward, those strategies may not be observed as frequently. This also explains why the Q-learning model is found to outperform the Markov chain model in the high-performing group, since the Markov chain model does not reinforce these patterns.

6. Conclusion

This article presented a study to test the extent to which the design knowledge acquired from a source problem using a Q-learning agent can be computationally transferable to a target problem in a different design context. We introduced a higher-level problem representation that allows capturing generalizable knowledge at the design process level. Using this representation, we developed a design agent based on Q-learning, a model-free RL algorithm to mimic human design strategies in the (source) system design context. A unique aspect of this model is that it does not require a pre-defined reward matrix to train the Q-table. Instead, it relies on a data-driven reward formulation using the first-order Markov chain model that ensures the trained agent mimics human designers' one-step look-ahead sequential behaviors. In the Q-learning model, we integrated a probabilistic model (known as the "noisy rational decision model") for the action transition, which allowed the control of reinforcement of the frequently occurring design action pairs in the training data.

There are three main findings from this study about human sequential decisions and the transfer of design knowledge. First, the performance of the Q-learning model in the high- and low-performance groups was sensitive to the setting of the reward matrix, that is, whether the reward matrix was formulated using the aggregated transition probability matrix or individual matrices. This again implies that the Q-learning agent performs effectively when the patterns in the sequential data are strong. Second, design behavioral patterns were more prominent in high-performing designers; therefore, their knowledge can be better captured and transferred. In other words, when transferring knowledge between different problems, the knowledge transferred from expert designers may be more reliable than that from novice designers. Third, compared to the commonly used Markov chain model, Q-learning agents were able to produce higher prediction accuracy in the high-performing group in both source and target problems. However, in the low-performing group (where the design patterns are weak in general), the p -value (i.e., 0.19) did not support the hypothesis that the Q learning model is better than the Markov chain model, and this has been found to be true regardless of the use of aggregated transition probability matrix or individual matrices.

There are a few limitations in the current work from which we identify opportunities for future research directions. First, the current RL model does not learn the optimal policy for good designs (designs that yield high objective values). Rather, it tries to maximally replicate the prominent design patterns found

from the first-order Markov chain model, so when the model is applied to learn expert designers' knowledge, it helps transfer the beneficial design patterns and transfer them to target design problems as needed. We acknowledge the potential to reinforce biases and undesirable behaviors if the agent is trained solely on data from a single designer. However, this effect can be mitigated by incorporating average knowledge and data from multiple designers or design experts. By assuming that experienced designers' action sequences are positively correlated with performance, our approach serves as a starting point for optimizing joint performance. The findings of this study encourage the future exploration of different sequential learning models in support of the formulation of the reward matrix beyond the Markov model. Second, some of the conclusions made are limited to the particular CAD-based design context and the two design problems that are both accomplished in the same CAD software.

The scope of this study is also limited to developing this generalizable process and providing evidence for knowledge transfer at a high-level problem abstraction. The extent of transferability may be influenced by participant characteristics such as experience level. A detailed analysis of the influence of such variables is left to a future study. In addition, the present study is limited to individual settings. Team-based decision-making adds other variables that could influence the findings, such as team structures, communication mechanisms, and interpersonal trust. A study of transferability in group settings is another interesting direction for future exploration.

In the next step, we plan to validate the RL-based agents in more design case studies to test the transferability of design knowledge across totally different design activities, for example, two different senior design projects where design topics and tools are all different. There is evidence in the literature showing that analogies from distant problems may stimulate novel design generation in human designers (Fu *et al.* 2013). Since we use a high-level abstraction of the design process, we expect transferrability even in distant problems considering that there are well-known problem-agnostic search strategies such as exploration-exploitation. However, transferability may be lower compared to similar problems, as the problem context may call for specific search strategies. Finally, we utilize the FBS framework for high-level design abstractions. Alternative high-level problem representations exist with varying capabilities for knowledge transfer across design problems. Finding the best representation for high-level knowledge transfer is a topic for a future study. One of the contributions of this article is that the proposed approach and procedure are general enough to support these identified future research explorations and many others as long as the data of design action sequence are obtainable.

Acknowledgment

The authors gratefully acknowledge the financial support from NSF-CMMI-1842588.

References

- Bayrak, A. E. & Sha, Z. 2020 Integrating sequence learning and game theory to predict design decisions under competition. *Journal of Mechanical Design* **143** (5), 1–14; doi: [10.1115/1.4048222](https://doi.org/10.1115/1.4048222).

- Behzadi, M. M. & Ilies, H. T.** 2021 GANTL: Toward practical and real-time topology optimization with conditional generative adversarial networks and transfer learning. *Journal of Mechanical Design* **144** (2), 021711; doi:[10.1115/1.4052757](https://doi.org/10.1115/1.4052757).
- Burnap, A., Ren, Y., Gerth, R., Papazoglou, G., Gonzalez, R. & Papalambros, P. Y.** 2015 When crowdsourcing fails: A study of expertise on crowdsourced design evaluation. *Journal of Mechanical Design* **137** (3), 031101; doi:[10.1115/1.4029065](https://doi.org/10.1115/1.4029065).
- Chaudhari, A. M., Bilonis, I. & Panchal, J. H.** 2020 Descriptive models of sequential decisions in engineering design: An experimental study. *Journal of Mechanical Design* **142** (8), 081704; doi:[10.1115/1.4045605](https://doi.org/10.1115/1.4045605).
- Cross, N.** 2023 *Design Thinking: Understanding how Designers Think and Work*. Bloomsbury Publishing.
- Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I. & Abbeel, P.** 2016 RL2: Fast reinforcement learning via slow reinforcement learning. <https://arxiv.org/abs/1611.02779>.
- Fu, K., Chan, J., Cagan, J., Kotovsky, K., Schunn, C. & Wood, K.** 2013 The meaning of “near” and “far”: The impact of structuring design databases and the effect of distance of analogy on design output. *Journal of Mechanical Design* **135** (2), 021007.
- Fuge, M., Peters, B. & Agogino, A.** 2014 Machine learning algorithms for recommending design methods. *Journal of Mechanical Design* **136** (10), 101103; doi:[10.1115/1.4028102](https://doi.org/10.1115/1.4028102).
- Gensch, D. H. & Recker, W. W.** 1979 The multinomial, multiattribute logit choice model. *Journal of Marketing Research* **16** (1), 124–132; doi:[10.1177/002224377901600117](https://doi.org/10.1177/002224377901600117).
- Gero, J. S.** 1990 Design prototypes: A knowledge representation schema for design. *AI Magazine* **11** (4), 26; doi:[10.1609/aimag.v11i4.854](https://doi.org/10.1609/aimag.v11i4.854).
- Gero, J. S. & Kannengiesser, U.** 2014 *The Function-Behaviour-Structure Ontology of Design BT – An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations*, pp. 263–283. Springer; doi:[10.1007/978-1-4471-6338-1](https://doi.org/10.1007/978-1-4471-6338-1).
- Jang, B., Kim, M., Harerimana, G. & Kim, J. W.** 2019 Q-learning algorithms: A comprehensive classification and applications. *IEEE Access* **7**, 133653–133667; doi:[10.1109/ACCESS.2019.2941229](https://doi.org/10.1109/ACCESS.2019.2941229).
- Kan, J. & Gero, J.** 2009 Using the FBS ontology to capture semantic design information in design protocol studies. In *About: Designing*, pp. 213–229. CRC Press; doi:[10.1201/9780429182433](https://doi.org/10.1201/9780429182433).
- Kannengiesser, U. & Gero, J. S.** 2009 An ontology of computer-aided design. In *Computer-Aided Design and Other Computing Research Developments* (ed. C. M. D. Smet & J. A. Peeters). Nova Science Publishers.
- Konidaris, G. & Barto, A.** 2007 Building portable options: Skill transfer in reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 895–900. Morgan Kaufmann.
- Lazaric, A.** 2012 *Transfer in Reinforcement Learning: A Framework and a Survey*, pp. 143–173. Springer; doi:[10.1007/978-3-642-27645-3_5](https://doi.org/10.1007/978-3-642-27645-3_5).
- McComb, C., Cagan, J. & Kotovsky, K.** 2015 Lifting the veil: Drawing insights about design teams from a cognitively-inspired computational model. *Design Studies* **40**, 119–142; doi:[10.1016/j.destud.2015.06.005](https://doi.org/10.1016/j.destud.2015.06.005).
- McComb, C., Cagan, J. & Kotovsky, K.** 2017 Capturing human sequence-learning abilities in configuration design tasks through markov chains. *Journal of Mechanical Design* **139** (9), 91101.
- Mehta, N., Natarajan, S., Tadepalli, P. & Fern, A.** 2008 Transfer in variable-reward hierarchical reinforcement learning. *Machine Learning* **73** (3), 289; doi:[10.1007/s10994-008-5061-y](https://doi.org/10.1007/s10994-008-5061-y).

- Phillips, C.** 2006 *Knowledge Transfer in Markov Decision Processes*. Technical Report, Citeseer.
- Rahman, M. H., Bayrak, A. E. & Sha, Z.** 2022 A reinforcement learning approach to predicting human design actions using a data-driven reward formulation. In *Proceedings of the Design Society: DESIGN Conference*, Vol. 1, pp. 1709–1718. Cambridge University Press; doi:[10.1017/pds.2022.173](https://doi.org/10.1017/pds.2022.173).
- Rahman, M. H., Schimpf, C., Xie, C. & Sha, Z.** 2019 A computer-aided design based research platform for design thinking studies. *Journal of Mechanical Design* **141** (12), 1–47; doi:[10.1115/1.4044395](https://doi.org/10.1115/1.4044395).
- Rahman, M. H., Xie, C. & Sha, Z.** 2021 Predicting sequential design decisions using the function-behavior-structure design process model and recurrent neural networks. *Journal of Mechanical Design* **143**, 1–46; doi:[10.1115/1.4049971](https://doi.org/10.1115/1.4049971).
- Rahman, M. H., Yuan, S., Xie, C. & Sha, Z.** 2020 Predicting human design decisions with deep recurrent neural network combining static and dynamic data. *Design Science* **6**, e15; doi:[10.1017/dsj.2020.12](https://doi.org/10.1017/dsj.2020.12).
- Raina, A., Cagan, J. & McComb, C.** 2019 Transferring design strategies from human to computer and across design problems. *Journal of Mechanical Design* **141** (11), 114501; doi:[10.1115/1.4044258](https://doi.org/10.1115/1.4044258).
- Raina, A., Cagan, J. & McComb, C.** 2022 Learning to design without prior data: Discovering generalizable design strategies using deep learning and tree search. *Journal of Mechanical Design* **145** (3), 031402; doi:[10.1115/1.4056221](https://doi.org/10.1115/1.4056221).
- Rajendran, J., Srinivas, A., Khapra, M. M., Prasanna, P. & Ravindran, B.** 2015 Attend, adapt and transfer: Attentive deep architecture for adaptive transfer from multiple sources in the same domain. <https://arxiv.org/abs/1510.02879>.
- Sexton, T. & Ren, M. Y.** 2017 Learning an optimization algorithm through human design iterations. *Journal of Mechanical Design* **139** (10), 101404; doi:[10.1115/1.4037344](https://doi.org/10.1115/1.4037344).
- Sha, Z., Kannan, K. N. & Panchal, J. H.** 2015 Behavioral experimentation and game theory in engineering systems design. *Journal of Mechanical Design* **137** (5), 51405; doi:[10.1115/1.4029767](https://doi.org/10.1115/1.4029767).
- Sunmola, F. T. & Wyatt, J. L.** 2006 Model transfer for Markov decision tasks via parameter matching. In *25th Workshop of the UK Planning and Scheduling Special Interest Group*. University of Nottingham.
- Sutton, R. S. & Barto, A. G.** 2018 *Reinforcement Learning: An Introduction*. MIT Press.
- Whalen, E. & Mueller, C.** 2021 Toward reusable surrogate models: Graph-based transfer learning on trusses. *Journal of Mechanical Design* **144** (2), 021704; doi:[10.1115/1.4052298](https://doi.org/10.1115/1.4052298).
- Wu, H., Ghadami, A., Bayrak, A. E., Smereka, J. M. & Epureanu, B. I.** 2021 Impact of heterogeneity and risk aversion on task allocation in multi-agent teams. *IEEE Robotics and Automation Letters* **6** (4), 7065–7072; doi:[10.1109/LRA.2021.3097259](https://doi.org/10.1109/LRA.2021.3097259).
- Xie, C., Schimpf, C., Chao, J., Nourian, S. & Massicotte, J.** 2018 Learning and teaching engineering design through modeling and simulation on a CAD platform. *Computer Applications in Engineering Education* **26** (4), 824–840; doi:[10.1002/cae.21920](https://doi.org/10.1002/cae.21920).
- Yu, R. O. N. G. R. O. N. G., Gero, J. O. H. N. S., Ikeda, Y., Herr, C. M., Holzer, D., Kaijima, S., Kim, M. J., Schnabel, A. & Citeseer** 2015 An empirical foundation for design patterns in parametric design. In *20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA)*, Daegu, South Korea, pp. 20–23. CAADRIA.
- Zhang, A., Satija, H. & Pineau, J.** 2018 Decoupling dynamics and reward for transfer learning. arXiv preprint [arXiv:1804.10689](https://arxiv.org/abs/1804.10689); doi:[10.48550/arXiv.1804.10689](https://doi.org/10.48550/arXiv.1804.10689).