Goal-Directed Design Agents: Integrating Visual Imitation With One-Step Lookahead Optimization for Generative Design

Ayush Raina
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: araina@andrew.cmu.edu

Lucas Puentes
Department of Mechanical Engineering,
The Pennsylvania State University,
University Park, PA 16802
e-mails: lucaspuentes22@gmail.com; lup93@psu.edu

Jonathan Cagan
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: cagan@cmu.edu

Christopher McComb
School of Engineering Design,
Technology, and Professional Programs,
The Pennsylvania State University,
University Park, PA 16802
e-mails: mccomb@psu.edu; uum209@psu.edu

Engineering design problems often involve large state and action spaces along with highly sparse rewards. Since an exhaustive search of those spaces is not feasible, humans utilize relevant domain knowledge to condense the search space. Deep learning agents (DLAgents) were previously introduced to use visual imitation learning to model design domain knowledge. This note builds on DLAgents and integrates them with one-step lookahead search to develop goal-directed agents capable of enhancing learned strategies for sequentially generating designs. Goal-directed DLAgents can employ human strategies learned from data along with optimizing an objective function. The visual imitation network from DLAgents is composed of a convolutional encoder–decoder network, acting as a rough planning step that is agnostic to feedback. Meanwhile, the lookahead search identifies the fine-tuned design action guided by an objective. These design agents are trained on an unconstrained truss design problem modeled as a sequential, action-based configuration design problem. The agents are then evaluated on two versions of the problem: the original version used for training and an unseen constrained version with an obstructed construction space. The goal-directed agents outperform the human designers used to train the network as well as the previous feedback-agnostic versions of the agent in both scenarios. This illustrates a design agent framework that can efficiently use feedback to not only enhance learned design strategies but also adapt to unseen design problems. [DOI: 10.1115/1.4051013]

Keywords: agent-based design, artificial intelligence, data-driven design, design automation

1 Introduction

Design is a complex, multi-step process that involves reasoning, creativity, planning, and efficient search, among other skills. Humans can visualize future states and goals using experience or domain knowledge, identifying promising search directions early in the design process [1–3]. Such abilities allow humans to shrink the design space radically [4]. Recently, data-driven learning has been an active part of research in engineering design [5–7]. Moreover, approaches that combine data-driven learning and lookahead search-based optimization methods have proven to be successful in a wide array of generic artificial intelligence problems [8–10]. These approaches are complementary. Data-driven learning, specifically deep learning, can efficiently represent complex multi-dimensional data and also learn highly non-linear relationships from it. Meanwhile, lookahead search provides fine-grained control by optimizing explicit objectives and more generalizability since it does not depend on observational data. This combined approach holds excellent potential for generative design, as design problems characteristically involve large state–action spaces and often have sparse or delayed feedback. The current work takes inspiration from human decision-making to enhance an existing design agent framework that leverages human visual intuition from observational data and optimizes low-level action strategies using lookahead search.

The agents in this existing framework are designed to solve parameterized configuration design problems [11] through decomposition as a sequential decision-making process [12]. The goal is to identify an optimal configuration of parametric components for a set of constraints and objectives using sequential decisions to add, remove, and change the parameters of components. In order for these agents to operate on such problems, there must be an associated set of design grammars and the design state must be representable as an n-dimensional matrix. Since a design begins as an empty set and is built iteratively to completion, there is often a period in which the design solution is incomplete, and it is often impossible to provide design quality feedback on incomplete designs. This challenge of delayed feedback is similar to most real-world design problems, where in the initial phase of a design process, the designers are guided by intuition. The proposed agent framework splits the design process into two parts based on feedback availability to address this challenge. When the feedback is unavailable, the agent simulates the intuition by visually imitating human designers using a deep learning framework defined in Ref. [13] and employs a heuristic guidance mechanism [14], both of which are feedback-agnostic. Alternatively, when feedback is available, the agent uses lookahead search to make designs, allowing better versatility and more focus on explicit goals. The one-step lookahead search is impossible when feedback is unavailable and hence using feedback-agnostic methods initially is essential. The feedback-based methods used later allow more explicit control on the objective function helping optimize it directly as compared to feedback-agnostic methods, making these two approaches complementary to each other.

The proposed agent framework models this goal-directed behavior of humans using a search-based approach. Making decisions that optimize for specific objectives constitutes a critical skill in problem solving [15]. This work investigates the effect of integrating data-driven strategies with explicit objective definitions on generative design performance. The agents employ a generic lookahead search that augments the learned strategies. These agents are tested on the original design problem and an unseen problem where an obstacle is introduced in the construction space. The performance results show the efficacy of the goal-directed agent framework and synergistic integration of different methodologies from deep learning, optimization, and heuristics with promising applications to design problem solving.

The rest of this note is organized as follows. Section 2 introduces the basic deep learning agent (DLAgent) architecture and the concept of heuristic guidance, along with other relevant background

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received October 26, 2020; final manuscript received April 16, 2021; published online June 9, 2021. Assoc. Editor: Jitesh H. Panchal.

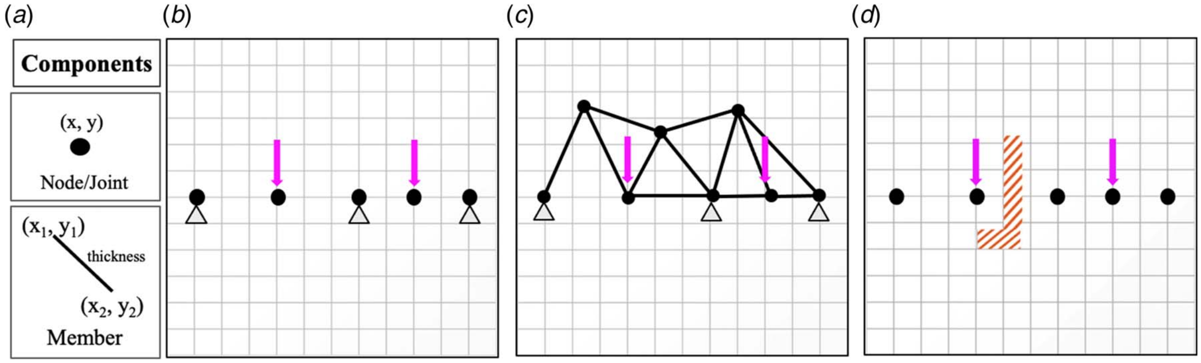

Fig. 1 Truss design problem. (*a*) The basic components: nodes and members that compose a truss design. (*b*) Initial state of the unconstrained truss design problem. (c) An example truss design. (*d*) Initial state of the constrained problem with hashed area representing the obstacle.

literature. Section 3 focuses on detailing the goal-directed agent framework. Section 4 explains the experimental setup and the human design dataset used as a baseline and training the visual suggestion network. Section 5 illustrates the performance results and relevant discussion. Lastly, Sec. 6 provides conclusions to this work.

2 Background

2.1 Deep Learning Agents and Heuristic-Guided Deep Learning Agents. DLAgents were introduced by Raina et al. [13] as a framework to implicitly capture design strategies from an image-based dataset of design state progression. These agents achieved human-level performance in truss design without any explicit information about the objective. There are two components of the agent framework: a deep learning network and an inference algorithm. The deep learning network creates a low-dimensional representation of prior design states and uses that to envision future states towards some arbitrarily defined objectives. This visualization of future design states is similar to how humans develop mental models [16] and visualize goal states to solve the required objectives [17,18]. Once the prediction is made, the inference algorithm identifies a particular action suggested by structural similarity [19] comparison. These DLAgents can generate meaningful truss designs and achieve performance similar to the humans used to train the network without ever performing an evaluation or receiving feedback. These agents are referred to as Vanilla DLAgents and are used as a comparative baseline in this note.

To expand the decision-making capabilities of Vanilla DLAgents, Puentes et al. [14] incorporated a guidance method to the framework that allows agents to follow multi-step design heuristics. These experience-derived strategies help focus on achieving specific design goals or subgoals [20–22]. Heuristics can improve the efficiency of design space exploration [23–27]. These new heuristic-guided agents referred to in this note as Temporal DLAgents select design actions to perform in a two-stage process. First, a set of candidate actions from the Vanilla DLAgent's inference algorithm gets classified as a heuristic based on similarity to a predefined set of heuristics. The set of candidate actions is ordered based on each candidate's visual similarity. By using this ordered set to identify a heuristic, an agent can use temporal information to follow said heuristic for multiple design iterations and execute its underlying strategy. For example, the execution of the heuristic *increase design scale* in a truss domain (as is done in both the current and previous works) may be done through the sequential application of multiple design actions to *increase truss member size*. Suppose a candidate list contains multiple high-ranking actions to *increase truss member size*. In that case, this could strongly indicate that the upcoming sequence of actions resembles the heuristic *increase design scale*. In the second stage, the candidate list gets filtered based on this heuristic classification, and the action that best continues the heuristic is selected. For example, if the heuristic *increase design scale* has been identified, then the agent will most likely continue to select the action to *increase truss member size* to continue that heuristic, as opposed to selecting an action such as *add new node*. The heuristic is enacted for a preset number of design actions, referred to as the burst length [28], after which a new classification is performed. This heuristic guidance enhances the performance of Vanilla DLAgents [14].

However, neither Vanilla DLAgents nor Temporal DLAgents utilize real-time objective feedback. The combination of data-driven models with real-time search systems has been successful in other domains [29–31]. The current work provides a mechanism to utilize real-time objective information on the design state using a one-step lookahead search to augment the design agents. The agent framework presented in Sec. 3 introduces a unique combination of data-driven strategies, pre-determined temporal heuristic relationships, and an objective-based lookahead search as a generic framework for design agents.

2.2 Truss Design Problem. The design of truss structures is a classical problem in structural mechanics and engineering. A truss can be geometrically defined as a spatial arrangement of nodes and members, these components and their associated spatial parameters as shown in Fig. 1(*a*). In this work, the problem is represented as a parameterized configuration design problem. It is further decomposed as a sequential decision-making process, as defined in previous works [9,26]. Designers sequentially select actions such as *add a node or member*, *delete a node or member*, and *increase or decrease thickness of members*. Every action also has associated spatial parameters that defines its location in the design space. Figure 1(*b*) shows the initial state of the design problem used in this work, with the nodes marked with arrows representing the loading points, while the other nodes with triangles represent the supporting points. Figures 1(*b*) and 1(*d*) show the unconstrained and constrained design scenario's initial boundary conditions, respectively.

3 Framework for Goal-Directed Design Agents

This section details the framework of the proposed goal-directed design agent (see Fig. 2). The framework is organized into three main parts. The first two parts represent the deep learning network and inference algorithm, identical to the previous Vanilla DLAgent framework [13]. The deep learning network is trained to visually predict the future state of the design, incrementally evolving the current design state towards an implicit goal. The future state, represented as a heatmap, contains differently shaded regions that correspond to a prediction of removing or adding

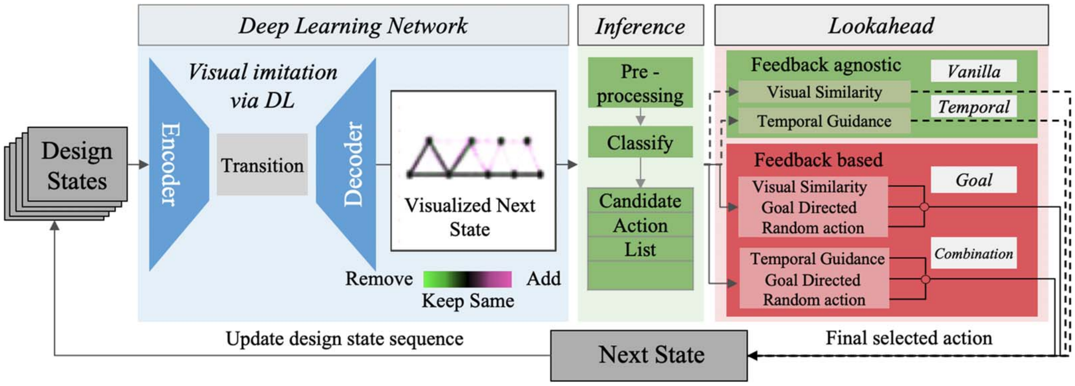

Fig. 2 Overall design agent framework

material in those regions. Generation of the heatmap is followed by the inference algorithm that uses rules based on image processing to identify a set of suggested candidate actions. This process maps the intuition suggested by the heatmap to a feasible set of actions in the given state. More details of these phases are provided in Ref. [13]. The final part of the framework uses lookahead evaluation, in which the agent makes the final action decision based on specific criteria and executes the selected actions to transition into a new state. This phase differentiates the new goal-directed agents from the prior Vanilla DLAgents and Temporal DLAgents, as they use different selection methods.

Vanilla DLAgents use a visual similarity-based metric that maintains high similarity between the heatmap and the next state, therefore mimicking human strategies from the dataset. Alternatively, Temporal DLAgents use the additional integrated heuristics along with visual similarity to filter candidate lists and maintain multi-step action relations; this combination of heuristic guidance and visual similarity to select an action is also referred to as temporal guidance. Both these previous methods are objective function feedback-agnostic (lookahahead component shown with dashed lines in Fig. 2) and have no explicit information about the design process's objectives. Instead, they are dependent on the design states' visual progression, and therefore are specific to the original design problem during training. In contrast, the new goal-directed agents (solid lines in lookahead section of Fig. 2) make their action selection based on feedback from an objective function. However, like most design problems, the objective value of a truss design is not defined for incomplete designs. As a result, only when a feasible design (i.e., a statically determinate design) is established, the feedback-based decision-making can be used. If the design is not feasible, the agent follows feedback-agnostic methods. All variants of DLAgents are shown in a tabular format in Table 1 to highlight their differences.

The two new versions of the goal-directed agents are defined and compared in this note:

(1) *Goal DLAgents*: These agents use the Vanilla DLAgent framework from the beginning of the process until a feasible state is reached. During this pre-feasible state, the agent will solely make design changes based on visual similarity. Once the objective feedback is available, the agent selects the best possible candidate action that leads to a state with the maximum objective value accomplishing a one-step lookahead search. This selection is based on a generic greedy method maximizing an arbitrary objective function and can be extended to an N-step lookahead evaluation. Extending to a higher number of steps should theoretically continually increase the performance at the cost of computation power, independent of the specific problem. In this work, the lookahead depth is limited to only one step. This simulates a greedy search based on a candidate list generated using a visual imitation-based prediction network. These agents isolate the effect of introducing goal-directedness when compared with the baseline DLAgents.

(2) Combination DLAgents: These agents use the Temporal DLAgents framework for the infeasible design phase. During this pre-feasible state, the agent will solely make design changes that follow temporal guidance (an identified heuristic and visual similarity). Once a feasible design is achieved, these agents use a combination of three methods to make the selection. The first option selects a candidate action based on temporal guidance, allowing it to behave similarly to a Temporal DLAgent. The second option chooses the action with the highest objective value, as Goal DLAgents utilize the available real-time feedback. The final option selects a random candidate action to introduce some additional stochasticity to the process. This tripartite selection procedure resembles an ε-greedy policy [32], where agents choose random action with a small probability of ε and select the greedy strategy with a probability of $1-\varepsilon$. Here, the agent selects one of these three methods based on a predefined weighting parameter tuned to balance exploration and exploitation. These combination agents illustrate that combinations of different methodologies can be integrated to develop variations within the goal-directed framework.

Table 1 DLAgent variants

	Feedback-agnostic	Feedback guided
No heuristic guidance	Vanilla DLAgents	Goal DLAgents
Heuristic guidance	Temporal DLAgents	Combination DLAgents

4 Experimental Setup and Dataset

The data used for training the DLAgents are derived from a human subjects truss design study in which teams of university engineering students completed a truss configuration problem [33]. The task was to create a truss design with minimal mass while meeting a factor of safety (FOS) requirement, with a value greater than 1.0 correlating to a feasible design that will not collapse under its load. The data from the human subjects were collected through a computer-based design interface where every action was recorded. Figure 1(*c*) shows an example of a truss design from the data. The dataset includes human trajectories from two different problems: an unconstrained construction space, as shown in Fig. 1(*b*), and a constrained construction space in which an obstacle was introduced, as shown in Fig. 1(*d*). Previously, these data have been successfully used to extract and represent design strategies as probabilistic models [34–36].

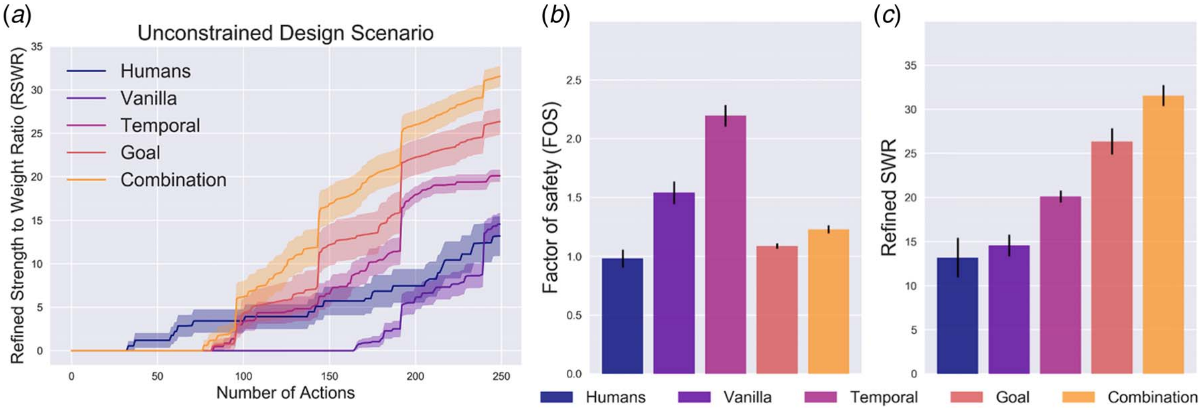

Fig. 3 Unconstrained problem. (*a*) RSWR comparison, bar plots for (*b*) FOS and (*c*) RSWR. ±1 standard error indicated.

In order to maintain a fair comparison with the human designers, certain modifications are made to the agent setup. The original design study was conducted in teams of three human designers. The team members could share their experimental designs through the interface and collectively reach a final design. The experiment included 16 teams, with each subject averaging 250 and 170 actions (or design iterations) for the unconstrained and constrained design scenarios, respectively. It should be noted that subjects average a lower number of actions in the constrained scenario, as they start from their previous design states. Furthermore, they interacted on average after every 48 iterations. In order to simulate a naive multi-agent setup, for every team, three identical instances of the design agent are initialized and collaborate with similar interaction frequencies over a similar run of iterations. During the interaction, every agent selects the highest performing feasible design at that instant from the team and then continues to work on them independently.

The prior feedback-agnostic agents are Vanilla DLAgents and Temporal DLAgents, which are compared with the new objective-driven agents Goal DLAgents and Combination DLAgents. These agents are evaluated on both the constrained and unconstrained versions of the problem. All agents begin with a random configuration of truss nodes and then iteratively act to reach their final design. The agents work on their design independently unless they interact, in which case, all agents always select and continue to iterate on the current highest quality design from the team's design pool. The whole process repeats until the agents reach 250 iterations for the unconstrained scenario or 700 iterations for the constrained scenario. A higher maximum iteration count is allowed for the constrained scenario since, unlike humans, agents start from scratch. However, agents are allowed to finish early in the case that no actions are suggested. As the data-driven part of these agents is only trained on the unconstrained version, the constrained design problem acts as the unseen version. Their performance is illustrative of the robustness of the learned design strategies and lookahead search. In order to incorporate this constraint in the agent decision-making, the candidate list generation is updated to filter out actions that violate the obstacle constraint. Implementing this filtering in the algorithm allows the design agents to respect this additional constraint without re-training the visual imitation algorithm, demonstrating the approach's potential adaptability.

5 Results and Discussion

5.1 Unconstrained Construction Space. Figure 3(*a*) shows the progression of the design agents in terms of refined strength-to-weight ratio (RSWR), which shows the SWR of only feasible designs (FOS ≥ 1.0). SWR is a standard single value metric that has been used to quantify the quality of a truss designs [13,33,37]. The ratio of strength (represented as FOS) to the mass of the design shows the efficiency of the design where a high SWR value corresponds to high strength with low mass implying a highly effective truss design. A high SWR value corresponds to high strength with low mass implying a highly effective truss design. Specifically, this plot shows the mean of the agents' best RSWR values up to a particular iteration, representing the average design progression trends. The Goal DLAgents perform better than humans and our previous baselines of Vanilla and Temporal DLAgents, but the Combination DLAgents ultimately perform the highest. This demonstrates that goal-directedness dramatically increases the design performance when compared with feedback-agnostic methods. Further integrating heuristic guidance enhances this performance even more. The greedy solution search of Goal Agents possibly limits their ability to search for diverse designs in the full solution space. By allowing agents to decide on actions using all three selection methods, it becomes possible for them to explore widely and optimize greedily. Figures 3(*b*) and 3(*c*) show the mean of the maximum value achieved for FOS and RSWR. These two metrics help in understanding the quality of the design produced. The FOS bars indicate that feedback-agnostic agents (Vanilla and Temporal DLAgents) reach very high strength values, showing that they continue to add more mass to the designs exceeding FOS values of 1.0. Alternatively, Goal DLAgents have minimal deviations, suggesting that once they reach the feasible threshold, they shift their focus to reducing mass and optimizing the design while maintaining a FOS around 1.0. The RSWR plots show similar trends as Fig. 3(*a*), with Combination DLAgents performing the best. All agents engage in a design sharing interaction step every 48th iteration to maintain the interaction frequency of humans. This leads to sudden jumps in the RSWR trajectory plots as all agent interact at the same iteration number unlike humans who interact at different iteration numbers but maintain the same average.

5.2 Constrained Construction Space. Figure 4(*a*) shows the RSWR progression of design states for the constrained problem with an obstacle (shown in Fig. 1(*d*)). These results demonstrate that humans achieve feasible and high-performing designs early on in the process, while the agents take longer to build feasible designs. This behavior is observed because humans begin with existing design states (a result of the original study's sequential nature) and modify them to adapt to the obstacle. Meanwhile, the agents start from scratch and determine feasible solutions after approximately 200 iterations. Although both feedback-agnostic agents perform similar to humans, objective-driven agents (Goal and Combined DLAgents) perform significantly better than them, illustrating the performance boost of utilizing the goal-directedness on a new problem. The agents can outperform humans by building

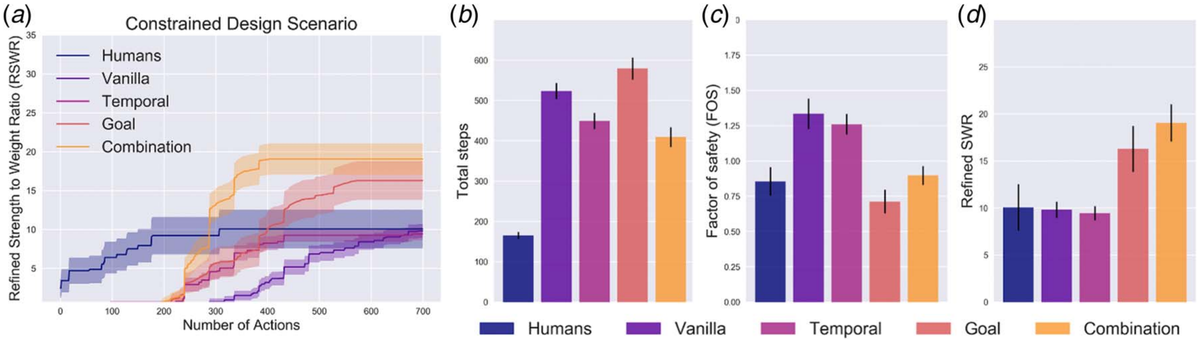

Fig. 4 Constrained problem. (*a*) RSWR comparison, (*b*) total steps taken, bar plots for (*c*) FOS and (*d*) RSWR. ±1 standard error indicated.

from their truss design strategies and further augmenting them with a lookahead search. Additionally, it can be observed from Figs. 4(*a*) and 4(*b*) that employing temporal guidance (Combination and Temporal DLAgents) allows the agents to reach higher performance faster. In terms of FOS, it can be observed in Fig. 4(*c*) that feedback-agnostic agents continue to improve strength, often having FOS greater than 1.0, rather than focusing on efficiency by reducing mass. Also, periodic jumps similar to Fig. 3 were observed due to the interaction step. Finally, the RSWR bar graph in Fig. 4(*d*) shows that humans, Vanilla, and DLAgents reach similar levels, demonstrating the effectiveness of the visual imitation strategies across problems. However, these agents are bested by Goal and Combination DLAgents. It shows that Goal DLAgents exhibit opportunistic behavior; they can select the best performing action from their candidate list, leading to high-performing designs. This indicates a practical methodology for utilizing real-time feedback information to improve design performance and adapt to unseen problem constraints.

6 Conclusion

Humans employ a combination of high-level visual intuition and low-level control strategies that help them to navigate the search space when engaged in design tasks efficiently. Our proposed agent framework models this behavior through an autoencoder-based visual imitation learning process that guides design actions and is further optimized by a one-step lookahead search. In this work, new variations of DLAgents are explored that follow design objective-driven actions. These goal-directed agent variations demonstrate, on an average, better performance than humans on two versions of a truss design problem. These results show that the agent can learn and then enhance design strategies from observational data. The experimental setup includes a new constrained problem on which the agents had not been trained. Currently, the agents use a rule-based inference algorithm to map the pixel-based visual guidance to actions. Future work may focus on learning to extract the feasible set of actions and developing an end-to-end trainable design agent framework. Additionally, further exploration of the proposed framework within other design applications is necessary to verify the transferability of the framework.

Acknowledgment

This material is based upon work supported by the Defense Advanced Research Projects Agency through cooperative agreement no. N66001-17-1-4064. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtained from the corresponding author upon reasonable request.

References

[1] Ferguson, E. S., 1994, *Engineering and the Mind's Eye*, MIT Press, Cambridge, MA.

[2] Kosslyn, S. M., Pascual-Leone, A., Felician, O., Camposano, S., Keenan, J. P., Thompson, W. L., Ganis, G., Sukel, K. E., and Alpert, N. M., 1999, "The Role of Area 17 in Visual Imagery: Convergent Evidence From PET and RTMS," Science, **284**(5411), pp. 167–170.

[3] Kosslyn, S. M., and Shwartz, S. P., 1977, "A Simulation of Visual Imagery," Cogn. Sci., **1**(3), pp. 265–295.

[4] Brown, D. C., and Chandrasekaran, B., 1989, *Design Problem Solving: Knowledge Structures and Control Strategies*, Morgan Kaufmann Publishers Inc., San Francisco, CA, pp. 35–64.

[5] Oh, S., Jung, Y., Kim, S., Lee, I., and Kang, N., 2019, "Deep Generative Design: Integration of Topology Optimization and Generative Models," ASME J. Mech. Des., **141**(11), p. 111405.

[6] Xiong, Y., Duong, P. L. T., Wang, D., Park, S.-I., Ge, Q., Raghavan, N., and Rosen, D. W., 2019, "Data-Driven Design Space Exploration and Exploitation for Design for Additive Manufacturing," ASME J. Mech. Des., **141**(10), p. 101101.

[7] Odonkor, P., and Lewis, K., 2019, "Data-Driven Design of Control Strategies for Distributed Energy Systems," ASME J. Mech. Des., **141**(11), p. 111404.

[8] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D., 2016, "Mastering the Game of Go With Deep Neural Networks and Tree Search," Nature, **529**(7587), pp. 484–489.

[9] Anthony, T., Tian, Z., and Barber, D., 2017, "Thinking Fast and Slow With Deep Learning and Tree Search," Adv. Neural Inf. Process. Syst., **2017**(Dec.(Il)), pp. 5361–5371.

[10] Lee, K., Kim, S.-A., Choi, J., and Lee, S.-W., 2018, "Deep Reinforcement Learning in Continuous Action Spaces: A Case Study in the Game of Simulated Curling," PMLR, Stockholm, Sweden, July 10–15, pp. 2937–2946.

[11] Wielinga, B., and Schreiber, G., 1997, "Configuration-Design Problem Solving," IEEE Exp. Syst. Appl., **12**(2), pp. 49–56.

[12] Yukish, M. A., Miller, S. W., and Simpson, T. W., 2015, "A Preliminary Model of Design as a Sequential Decision Process," Procedia Comput. Sci., **44**, pp. 174–183.

[13] Raina, A., McComb, C., and Cagan, J., 2019, "Learning to Design From Humans: Imitating Human Designers Through Deep Learning," ASME J. Mech. Des., **141**(11), p. 111102.

[14] Puentes, L., Raina, A., Cagan, J., and McComb, C., 2020, "Modeling a Strategic Human Engineering Design Process: Human-Inspired Heuristic Guidance Through Learned Visual Design Agents," Proc. Des. Soc. Des. Conf., **1**, pp. 355–364.

[15] Verschure, P. F. M. J., Pennartz, C. M. A., and Pezzulo, G., 2014, "The Why, What, Where, When and How of Goal-Directed Choice: Neuronal and Computational Principles," Philos. Trans. R. Soc. B: Biol. Sci., **369**(1655), p. 20130483.

[16] Thomas, N. J. T., 2021, *Mental Imagery – The Stanford Encyclopedia of Philosophy*, E. N. Zalta, ed., Spring 2021, Metaphysics Research Lab, Stanford University, Stanford, CA, pp. 1–45.

[17] Athavankar, U. A., 1997, "Mental Imagery as a Design Tool," Cybern. Syst., **28**(1), pp. 25–42.

[18] Goldschmidt, G., 1992, "Serial Sketching: Visual Problem Solving in Designing," Cybern. Syst., **23**(2), pp. 191–219.

[19] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., 2004, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Trans. Image Process., **13**(4), pp. 600–612.

[20] Nisbett, R. E., and Ross, L., 1980, *Human Inference: Strategies and Shortcomings of Social Judgment*, Prentice-Hall, Englewood Cliffs, NJ.

[21] Yilmaz, S., Seifert, C., Daly, S. R., and Gonzalez, R., 2016, "Design Heuristics in Innovative Products," ASME J. Mech. Des., **138**(7), p. 071102.

[22] Daly, S. R., Seifert, C. M., Yilmaz, S., and Gonzalez, R., 2016, "Comparing Ideation Techniques for Beginning Designers," ASME J. Mech. Des., **138**(10), p. 101108.

[23] Lenat, D. B., 1983, "EURISKO: A Program That Learns New Heuristics and Domain Concepts," Artif. Intell., **21**(1–2), pp. 61–98.

[24] Laird, J., Newell, A., and Rosenbloom, P. S., 1987, "SOAR : An Architecture for General Intelligence," Artif. Intell., **33**(1), pp. 1–64.

[25] Langley, P., McKusick, K. B., Allen, J. A., Iba, W. F., and Thompson, K., 1991, "A Design for the ICARUS Architecture," ACM SIGART Bull., **2**(4), pp. 104–109.

[26] Blösch-Paidosh, A., and Shea, K., 2019, "Design Heuristics for Additive Manufacturing Validated Through a User Study," ASME J. Mech. Des., **141**, p. 041101.

[27] Daly, S., and Christian, J. L., 2012, "Assessing Design Heuristics for Idea Generation in an Introductory Engineering Course," Int. J. Eng. Educ., **28**(2), pp. 463–473.

[28] Königseder, C., and Shea, K., 2016, "Visualizing Relations Between Grammar Rules, Objectives, and Search Space Exploration in Grammar-Based Computational Design Synthesis," ASME J. Mech. Des., **138**(10), p. 101101.

[29] Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., and Silver, D., 2019, "Mastering Atari, Go, Chess and Shogi by Planning With a Learned Model," arXiv:1911.08265v2, pp. 1–21.

[30] Brown, N., and Sandholm, T., 2018, "Superhuman AI for Heads-Up No-Limit Poker: Libratus Beats Top Professionals," Science, **359**(6374), pp. 418–424.

[31] Nair, A., Pong, V., Dalal, M., Bahl, S., Lin, S., and Levine, S., 2018, "Visual Reinforcement Learning With Imagined Goals," Adv. Neural Inf. Process. Syst., **2018**, pp. 9191–9200.

[32] Sutton, R. S., and Barto, A. G., 2018, *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, MA, pp. 73–94.

[33] McComb, C., Cagan, J., and Kotovsky, K., 2015, "Rolling With the Punches: An Examination of Team Performance in a Design Task Subject to Drastic Changes," Des. Stud., **36**, pp. 99–121.

[34] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Utilizing Markov Chains to Understand Operation Sequencing in Design Tasks," *Design Computing and Cognition '16*, J. S. Gero, ed., Springer International Publishing, Cham, pp. 401–418.

[35] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Mining Process Heuristics From Designer Action Data Via Hidden Markov Models," ASME J. Mech. Des., **139**(11), p. 111412.

[36] Raina, A., Cagan, J., and McComb, C., 2019, "Transferring Design Strategies From Human to Computer and Across Design Problems," ASME J. Mech. Des., **141**(11), p. 114501.

[37] Brownell, E., Cagan, J., and Kotovsky, K., 2021, "Only as Strong as the Strongest Link: The Relative Contribution of Individual Team Member Proficiency in Configuration Design," ASME J. Mech. Des., **143**(8), p. 081402.