



# MODELING A STRATEGIC HUMAN ENGINEERING DESIGN PROCESS: HUMAN-INSPIRED HEURISTIC GUIDANCE THROUGH LEARNED VISUAL DESIGN AGENTS

L. Puentes<sup>2</sup>, A. Raina<sup>1</sup>, J. Cagan<sup>1</sup>, and C. McComb<sup>2</sup>

<sup>1</sup> Carnegie Mellon University, United States of America, <sup>2</sup> The Pennsylvania State University, United States of America

 [cagan@cmu.edu](mailto:cagan@cmu.edu)

## Abstract

Human designers often work in a visual design space, projecting step-by-step design progression through evolving mental images. The strategic evolution of that design leverages heuristics based on experience and domain knowledge. The methodology presented in this paper brings together the visual nature of design problem solving and design heuristics in a deep learning computational agent framework that emulates and enables human-mirrored design. When applied to a truss design task, results demonstrate superior results to those of human designers who provided the initial data.

*Keywords: artificial intelligence (AI), engineering design, design automation, deep learning, heuristics*

## 1. Introduction

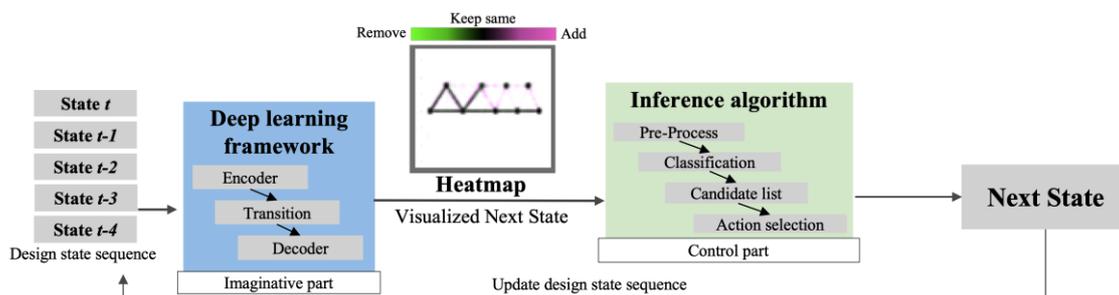
Humans are strategic planners. We can envision future scenarios and mentally lay out the steps needed to bring these visions to reality. This foresight ability is especially useful in engineering design. Designers can cognitively picture future iterations of their work and establish a step-by-step process to reach that point. To an outside viewer, the singular changes a designer makes during their design process may seem insignificant on their own. In reality, the designer is enacting a deliberate sequence of precise changes that will help them strategically achieve their design goal. This research paper works towards employing these design behaviours in an automated computational setting by fusing two existing frameworks that directly reflect the cognitive tendencies of designers: visual design thinking (Raina et al., 2019) and strategy-driven design search (Puentes et al., 2019). More specifically, this paper looks to enhance the design decision-making process of a visual deep learning agent by providing it with a method of heuristic-guidance, resulting in solutions superior to any of those found in the human data used to train the deep learning network. Through this methodology, human designers can employ the design strategies that they themselves would manually use for problem solving within an automated framework. The heuristic-imbued design agents can explore design solutions in a manner similar to the human designer but at a much more rapid pace, thus allowing the human to shift their focus and effort to other stages in the design process. The following sub-sections further establish the two previous frameworks and detail the process for their combined implementation.

### 1.1. Visual design agents

Physical design is usually a visual process where designers visually interact with their design environments; they observe the current design state, make an action decision, and then receive the

updated design state as visual feedback. This can be simplified and modelled as a sequential decision-making process (McComb et al., 2017a; Miller et al., 2015; Yukish et al., 2015), allowing the decomposition of the overall process into sequential design states. Designers analyse these states and make one change at a time to help progress the current design state towards a goal. Although meta design processes can be iterative, the execution of actions follow one another over a period of time. Moreover, humans are shown to think and solve problems visually (Brooks, 1991), create mental models of the problem (Athavankar, 1997) and also mentally envision changes required for the current designs so that an objective is reached (Goldschmidt, 1992). Humans look for patterns and spatial relations during designing that often have essential implicit information related to the problem. For example, symmetry is sometimes preferable and triangular structures might provide more strength. A design agent that can replicate the human designer behaviour should function in a similar manner by visually interacting with a design interface or environment and making temporally sequential actions. Representing the design state as an image (or an N-dimensional vector) is applicable if it can capture the entire design state information (otherwise alternative quantifiable representations can be used). Using an N-dimensional vector to represent design states allows the agent formulation to generalize across similar visual design processes.

Previous work with visual deep learning agents, or DLAges, (Raina et al., 2019) illustrates learning design behaviour from human data, and the ability to generate designs in a similar manner to humans. The agents were able to learn implicit human strategies by observing state progression and mimic them to reach human-level performance without any explicit performance feedback during training. Figure 1 represents the components from the original DLAges algorithm. The algorithm has two functionally different parts: the imaginative part that uses data-driven imitation learning based methods for visualising solutions, and the control part that is responsible for selecting the design action. Since these two parts are separate, heuristic-based strategies can be independently added to the control part to observe the combined effect of these two methods.



**Figure 1.** Schematic representation of the original DLAges framework (Raina et al., 2019)

The first part of the agent algorithm is the deep learning framework, where the task for the network is to predict or visualize the future state. The framework is composed of a deep convolutional autoencoder (Masci et al., 2011) and an artificial neural network (transition network). Autoencoders, being a highly efficient method of dimensionality reduction (Hinton and Salakhutdinov, 2006) convert the design state image to a low dimensional representation and then back to the original image dimensions. A transition network transforms the low-dimensional representation of the current state to the representation of the future state and in this process, it implicitly applies the learnt knowledge to predict the future state. Since the transition network is the only part that predicts the future state, it is solely responsible for solving the design problem. The framework produces a visualization of the next step which is then represented as a colour coded heatmap showing the highlighted pixel corresponding to addition or removal of material from particular regions. The training of this framework is similar in essence to showing the model multiple videos of humans designing a truss structure without any explicit action or objective information and then asking the framework to replicate the process.

The second part is the control part which infers the final action parameters using an image-processing based inference algorithm. The algorithm determines the action and its parameters that are suggested in the heatmap. The inference algorithm, however, does not account for any temporal (multi-time-step) relationships for action selection. This paper extends this functionality by adding heuristic

knowledge to introduce temporal relations in design actions for better guiding the agent algorithm. These relations allow agents to follow multi-step heuristic strategies during action selection that are adopted from a two-tier formulation (Puentes et al., 2019) as explained in the next sub-section.

## 1.2. Leveraging heuristics for computational design

Through experience, human designers develop their own heuristics, or personal strategies, to assist with problem-solving. These heuristics provide designers with guidelines to smoothly and efficiently navigate through familiar design domains. An observational design study by Yilmaz and Seifert (2011) noted that heuristic-based strategies led to a varied pool of solutions to a given problem. Their research further acknowledged significant heuristic use by experts, which led those designers to search for more creative and novel design concepts. However, the lack of domain experience for novice designers inhibits their ability to have fully developed their own personal heuristics for completing design tasks. Ahmed et al. (2003) observed that a novice designer's lack of heuristic strategies results in non-expansive, localized solution search. In an unfamiliar design domain, novices are likely to search experimentally in a way which resembles a trial-and-error approach. Slowly, a novice may navigate a solution space towards refined designs by way of incremental design changes. Yet, an expert can translate past experiences into a strategic sequence of planned design changes. This type of design methodology often results in vast solution space exploration which can quickly locate effective designs.

Past studies have demonstrated that sets of generalized heuristics may be beneficial to designers who may not yet have developed their own personal set. These high-level, abstract strategies can allow those individuals with limited experience in a specific design domain to explore that design space more effectively. Daly and Christian (2012) provided novices with a set of design heuristics to assist with concept generation. They noticed direct positive correlation between designs which showed clear use of these heuristics and a high level of creativity, novelty, and expansive solution search similar to expert designers. Kramer et al. (2014) further explored the research of Daly and Christian, showing that these generalized design heuristics are applicable across multiple design domains. This finding confirms that although the strategic concepts contained within generalized heuristics may take alternative forms depending on the specific domain, the strategies themselves may still prove helpful for inspiring expansive and effective solution search.

One such tool that designers may use to represent a design and navigate through a solution space are graph grammars. Graph grammars are used to visually represent design states and iteration-by-iteration changes. Through these grammars, the layout of a physical system is represented as a series of nodes and connections which contain information on the parameters of corresponding objects (Chakrabarti et al., 2011; Königseder and Shea, 2014; Schmidt and Cagan, 1997; Stöckli and Shea, 2017). A ruleset contains a list of all available actions, called rules, which can be performed to alter the layout of the system. These actions include the addition, removal, and manipulations of nodes and connections. Design problems are solved through iterative selection and application of rules. Computational problem-solving methods can greatly increase the speed and span of grammar-based design exploration, but such methods typically lack the heuristic-driven search tendencies of human designers. Many existing optimization techniques, such as those explored in by several other researchers (Campbell et al., 2012; Kirkpatrick et al., 1983; Reddy and Cagan, 1995; Schmidt and Cagan, 1997; Shea and Cagan, 1997; Tai and Akhtar, 2005) can follow a multi-step solution search, however, they typically do not do so in a design goal-driven manner similar to humans. These algorithms are very effective for rapid and intensive design exploration, but risk not fully exploring the valid solutions to a design problem.

Prior work by Puentes et al. (2019) aimed to enhance computational grammar-based design methods through the incorporation of generalized heuristics. A two-tiered grammar approach was presented which splits a design grammar into a "lower-tier" and a "higher-tier". The lower-tier contains the traditional single-step grammar rules often used in grammar-based design. Alternatively, the higher-tier contains multi-step macro rules based on the strategic concepts of generalized design heuristics. The set of heuristics chosen for this tier may be derived from a designer's personal experience in a domain or from a wider pool of abstract design principles. Application of a higher-tier rule is done through executing a guided sequence of lower-tier rules which will accomplish the design principle of that heuristic. This approach can be seen in the Figure 2. As it is likely that a single heuristic may be

applicable to a domain in numerous ways, interpretation of the heuristic and its translation to a sequence of lower-tier rules is up to the discretion of the designer who establishes the grammar. Across different domains, the abstract design principle conveyed by a heuristic is likely to have different translations. However, the underlying principle is fundamentally the same. This two-tiered grammar approach was implemented within a computational agent team framework and applied to the design generation of trusses and wave energy converters. These applications demonstrated that the inclusion of heuristics to guide grammar-based design allowed for much more rapid solution improvement during early design iterations than traditional single-step methods. Additionally, this two-tiered approach exhibited similar performance benefits across two different design domains, suggesting that the generalizable nature of the design heuristics used may translate to comparable success in other domains. To enhance the design capability of the DLAgents, this work will incorporate the heuristic grammars to improve move selection resulting from the inference algorithm. In this way, the agents will not only visualize, reason and act on the design, but do so in a strategic way, using design heuristics.

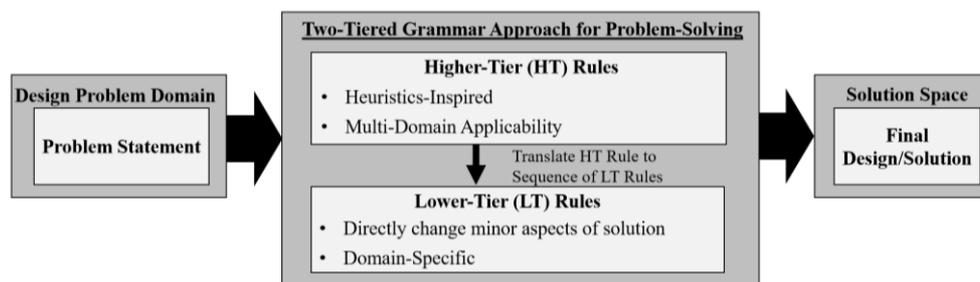


Figure 2. Two-tiered approach to grammar-based design problem solving (Puentes et al., 2019)

### 1.3. Prior research gaps and proposed methodology

Previous work has shown that the DLAgents introduced by Raina et al. (2019), although they are very effective for predicting immediate design changes based on prior design states, cannot accurately resolve temporal relationships between action selection across iterations. Their iteration-by-iteration action selection method requires agents to choose a design action to achieve greatest resemblance between the updated design state and the visualized next design state. By using some degree of selection guidance based on human-derived heuristics, agents can strategically navigate the design space in a manner similar to experienced human designers. This may lead to a more efficient and thorough exploration of design space. The two-tiered grammar-based approach to design introduced by Puentes et al. (2019) demonstrated that heuristic-guidance can be effective for performing multi-step design changes aimed at accomplishing a conceptual goal.

The methodology proposed in this paper enhances the action selection capabilities of the data-driven DLAgent algorithm through the guidance of generalized human-inspired heuristics. Whereas a human designer usually must select between a manual, heuristic strategy-driven solution search method or an automated, heuristic strategy-lacking method, this methodology aims to capture the advantages of both. The problem solving behaviours of design agents empowered with heuristics is intended to mirror that of human designers, and thus the need for manual solution search from a human is reduced. Figure 3 illustrates the location of the additional heuristic guidance component (shown in red) with respect to the overall agent algorithm. While design actions in the prior DLAgents were selected on a best-at-current-iteration basis, this new heuristic-guided process classifies the agent's heatmap as a heuristic and then guide the following actions to follow that heuristic strategy. The agent follows the heuristic for multiple design iterations, selecting design actions which continue to best follow the classified heuristic and hence streamline its action selection process by maintaining temporal relations. This process provides guidance in action selection for DLAgents and enables them to perform sequences of design actions that resemble certain generalized design heuristics.

The remainder of this paper is organized as follows. Section 2 presents the framework for the DLAgents and the heuristic-guidance process. Section 3 explains the truss design problem applied to

the heuristically-guided DLAgent framework. In Section 4, the results of this application are conveyed and discussed. Lastly, Section 5 concludes this paper and details limitations and future work.

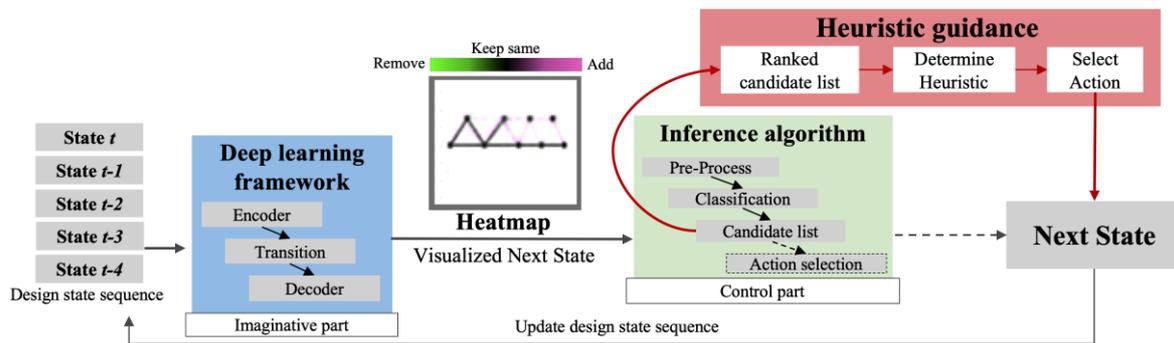


Figure 3. Schematic representation of the proposed Heuristic-Guided Deep Learning Agents (HG-DLAgents). Dotted line components are replaced by the red block to add heuristic guidance.

## 2. Methodology

This section presents the methodology for integrating a heuristic-guidance method with the DLAgent architecture. Section 2.1 describes the deep learning framework which DLAgents use to generate predictive suggestion heatmaps. The inference algorithm that generates a list of candidate actions is explained in Section 2.2. Lastly, Section 2.3 details the heuristic-guidance method for DLAgent design action selection.

### 2.1. Visual deep learning framework

The deep learning framework is kept identical to the original formulation (Raina et al., 2019). An overview of the framework is provided here, but readers are referred to (Raina et al., 2019) for details. The framework takes in input of 5 previous design states and predicts what the next design state should look like. The two layered transition network is tasked to imitate the humans designing trusses by learning the relationships between sequential design states and, hence, implicitly extracting design knowledge and strategies from the data. The changes to the current design state are generated due to the matrix operations that are executed in the transition network fully-connected layers. Given that the latent layer representation can capture a wide variety of design states, the operations in the transition network can potentially model any feasible action and produce an arbitrary new latent representation. This representation is passed to the decoder network, which then maps it back to a full-size image that is human interpretable and visually shows the change made in the transition network.

### 2.2. Action inference algorithm

The output from the deep learning framework is an image representing the predicted next state; an image processing-based algorithm similar to the original (Raina et al., 2019), is used to infer suggested actions. The algorithm takes the suggestion heatmap (as shown in Figure 4), which has pink and green highlighted pixels representing suggestions to add and delete material from those regions, respectively. It passes the heatmap through multiple steps of pre-processing before a rule-based classifier is used to classify if a pixel region appears like a node or a member connection between two nodes. As discussed by (Raina et al., 2019), the suggestion heatmap often produces noisy output that may correspond to multiple design actions possibly due to an averaging effect of multiple design strategies, and hence, a candidate list of actions is developed (as shown in Figure 4) to preserve all information. The final step in the algorithm compares the structural similarity between the predicted future state and all the states reached by applying each of the candidate actions individually. This produces a ranked list of the candidate actions based on which action produces a state that is closest to the predicted state. This, in essence, is similar to how humans make decisions; they choose the action that maximally enables them to realise their visualized goal. The final step where the action with the highest similarity is probabilistically picked from the list, is modified in this new framework. The ranked list is exported to

the heuristic guidance algorithm to make the final decision. Details of how the heuristic guidance algorithm affects this final selection step is explained in the next sub-section.

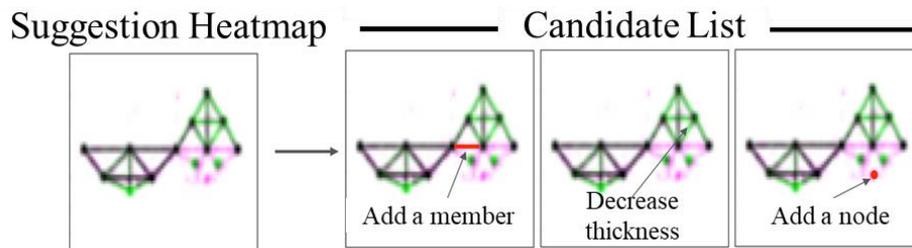


Figure 4. An example of a part of candidate list generated by the action-inference algorithm

### 2.3. Multi-step heuristic-guidance

In order to provide heuristic-guidance to DLAagents, a designer must first establish a set of heuristics applicable to the given design domain. These identified heuristics may come from a designer’s prior experience in that domain or may be influenced by more general design heuristic concepts. As design heuristics are abstract concepts rather than definitively concrete sequential actions, executing a single heuristic may be done through a variety of different action sequences. Thus, for each desired heuristic, the designer must define a set of design change sequences which may be taken to achieve that heuristic action. Across different design domains, the interpretation of a heuristic and its translation to action sequences is likely to be different. However, the underlying abstract design principle represented by that heuristic will remain the same.

The ordered list of candidate design changes that are extracted from an agent’s predictive heatmaps is classified as one of the established heuristic concepts. Because this list is ordered by descending action recommendation weight, it resembles a sequence of actions that a designer may perform during subsequent design iterations. The action list is compared to the predefined heuristic sequences and classified as the heuristic to which it is most similar. This identified heuristic is then used to guide a specific action selection. The flowchart for this heatmap-to-heuristic process can be seen in Figure 5.

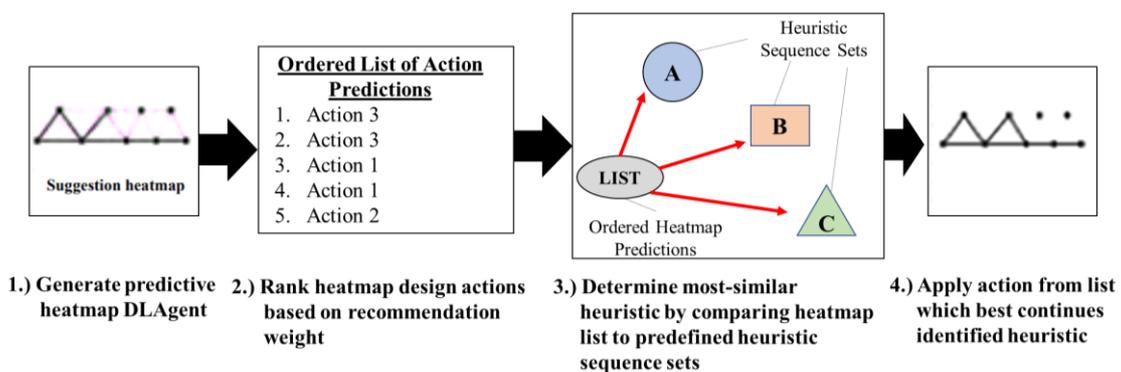


Figure 5. Process for classifying a DLAagent’s predictive heatmap actions as a heuristic sequence and selecting an action

To ensure that the identified heuristic has adequate time to accomplish its short-term design goal, a predefined number of design changes must occur before a reclassification can be done. This predefined number of changes is referred to as the burst length, as previous research has explored “burst algorithms” in which multiple design changes are applied in succession before a design is evaluated (Königseder and Shea, 2016). While the guiding heuristic remains the same across a full burst, the predictive action heatmap is updated after every design iteration. Each action in the heatmap list is tested as an extension of the burst’s current sequence, after which the action that results in the best following of the identified heuristic is selected. In the case that multiple actions may equally best follow the heuristic, the action with the higher recommendation weight is selected. The selected action is then applied to the design and the burst continues. Once a burst has concluded, the sequence is reset.

### 3. Case study and experimental setup

The presented methodology is applied to a 2D truss design problem. The design dataset used to train visual deep learning agents was gathered from a human design study in which students were tasked with constructing truss configurations (McComb et al., 2018). Previous research has shown success in extracting and representing design strategies from this same dataset (McComb et al., 2017a, 2017b; Raina et al., 2019). A detailed explanation of the full experiment can be found in McComb et al. (2015). The subjects were divided into teams to simulate real-life engineering design teams. They were given the task of generating truss designs with minimal mass and a factor-of-safety (FOS) over 1.0, under which they design would collapse. A computer-based truss design interface allowed them to generate their designs using a set of available design actions and permitted interaction among team members. Through this interface, designers took sequential actions to induce incremental changes to the current truss design state, ultimately leading to a complete and feasible truss. A log of all design actions performed by the teams was collected. The design actions for this study allowed for adding, removing, or moving a joint or truss member, or changing the member diameter.

In accordance with the process described in Section 2.3, a set of heuristics applicable to the truss design domain is established. As the prior two-tiered grammar approach research was also applied to a truss design application, a similar set of five design heuristics is selected for use in this paper. Originally, this set was sampled and adapted from an extensive list of generalized design heuristics presented by Daly and Christian (2012). These heuristics allow for large-scale modifications that are typically seen in configuration design problems. Although the heuristics themselves are the same, their sequential action execution is altered to represent the change in design actions available during the human design study. The five heuristics used to guide DLAgents for this design application are as follows: Increase design complexity, decrease design complexity, increase design scale, decrease design scale, and standardize.

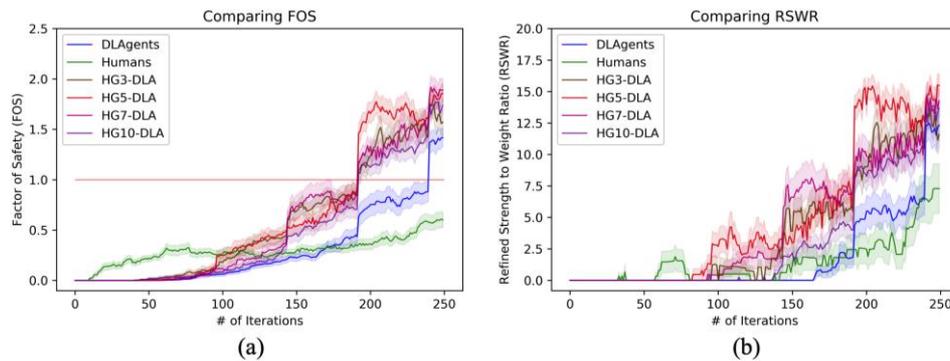
For each heuristic in the set, an extensive set of design change sequences is established to represent the many discrete ways that a heuristic may be implemented in a truss design. As the actual number of sequential variations for the chosen heuristics is large, the process for defining these sequences is automated. Rules are established for the generation process to simulate boundaries on the use of a heuristic. For example, the heuristic for “increase design complexity” would be executed in a design scenario as a sequence of adding truss joints and connecting members. Limits are placed on the maximum length of this heuristic. A similar constraint method is used for all heuristics in the set.

The experimental setup is identical to the setup in the original DLAgent formulation (Raina et al., 2019), except the agent framework is modified as per the new methodology and the new agents are termed as Heuristically Guided Deep Learning Agents (HG-DLAgents). Agent team sizes and the total number of teams are the same as the human study to ensure a fair comparison. Each agent starts from a randomly generated initial state and iteratively takes actions. The agents work on their design independently unless they interact, at which point they share their current design in the team’s common design pool. The agents pick the highest quality of design from the pool and individually continue working on it until the next interaction. The whole process is repeated until the agents reach 250 iterations, which was the mean number of iterations performed by human teams in the study by McComb et al. (2015). For comparison, different versions of the agent framework are setup with varying burst lengths, represented as HGn-DLAgents, where n represents the burst length. It must be noted that a burst length of 1 allows the HG-DLAgent model to collapse to a DLAgent model i.e. no temporal action guidance. Burst lengths from 3 to 10 are evaluated and compared.

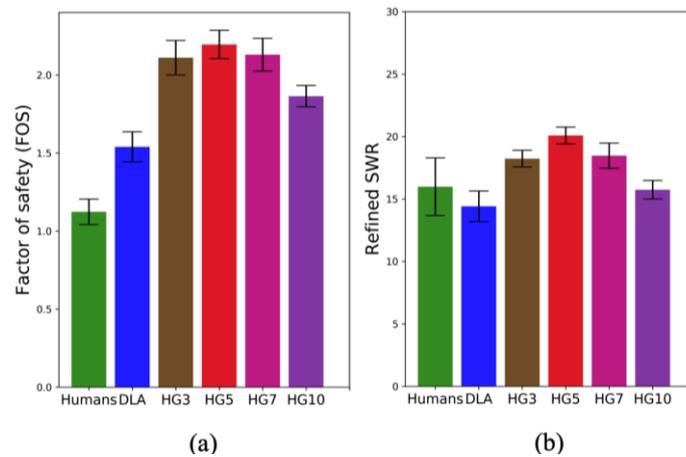
### 4. Results and discussion

This section provides the results and relevant discussion from the design application experiments. Figures 6 and 7 show the comparative plots of the performance of HG-DLAgents to the original DLAgents and humans from the original dataset. For the plots in Figure 6, the solid thick line shows the mean of all agents belonging to the colour coded class at that particular iteration while the shaded area represents the standard errors at that iteration. These plots show the general trend of the agent with increasing iterations. The trend plots in Figure 6 are limited to 250 iterations for the purpose of better readability since the majority of the data can be shown within the range; for calculating the best

performance, all iterations are considered in Figure 7. Figure 6 (a) shows the comparison for the factor of safety (FOS) of the truss, which indicates how much stronger a design is than is required to avoid structural failure. A FOS above 1.0 correlates to a design which meets or exceeds its strength requirement, while a FOS below 1.0 correlates to a design that must be strengthened to be feasible. Since the agents are tasked to produce feasible design (where FOS is greater than 1) with minimal mass, refined strength to weight ratio (RSWR) is used. This metric computes the strength to weight ratio (SWR) of feasible designs and hence only computing the quality of generated feasible designs. The comparison of RSWR is shown in Figure 6 (b). Figure 7, meanwhile, compares the mean across all agents for the best designs reached throughout their process. The final conclusions about best or worst performance are made from these bar-graphs since the line-graphs in Figure 6 only show the trends.



**Figure 6. Average iterative design metrics comparison between human designers, DLAgents, and HGn-DLAagents: (a) factor of safety, and (b) strength-to-weight ratio of only feasible designs**



**Figure 7. Comparison of the average of the best designs between human designers, DLAgents, and HGn-DLAagents: (a) factor of safety, and (b) strength-to-weight ratio of only feasible designs**

Through heuristic guidance, HG-DLAagents produce solutions quicker and of a higher quality than their unguided agent counterparts across almost all burst length variations. Additionally, heuristic-guided agents outperform even the original human design teams, though on average they are slower to reach a feasible solution. By the end of the design process, HG-DLAagents generate truss designs with FOS values which are approximately double that of human designers and about 50% higher than unguided DLAagents. This drastic increase is likely due to underlying methods human designers used for satisfying the design problem. As they are only instructed to create a design which had minimal mass while meeting a minimum FOS of 1.0, humans tended to meet the truss’s strength requirement early and optimize the design further from that point. Likewise, as standard DLAagents have the goal of mimicking human designers, they tend to remain at lower FOS. HG-DLAagents, in contrast, are not as limited by this emphasis on mere imitation. As they attempt to continue applying the same identified design heuristic for multiple iterations, it is possible that they remain stuck in a certain heuristic for longer than is required. In such a case, an HG-DLAagent may continue to make design changes which

add new joints or members to the truss or continuously increase the thickness of members in order to keep following a previously-identified heuristic. The benefit to this occurrence in this application is that HG-DLAgents begin to produce feasible designs significantly sooner than standard DLAgents. In comparison, standard DLAgents require, on average, over 150 design iterations to achieve their first feasible design, while HG-DLAgents are able to accomplish this in around half the time. The increased FOS along with the rapidly-achieved feasible design leads HG-DLAgents to produce solutions which surpass not only standard DLAgents in SWR, but human designers as well.

The effect of heuristic guidance is seen uniformly across the different burst lengths. Four different sequence burst lengths are tested in this application: 3, 5, 7, and 10 design actions per burst. The optimal burst length for this design problem is found to be 5 which dominates all other alternatives uniformly across both metrics. However, all other burst length agents also perform at least as well as the DLAgents or better, illustrating the positive impact of infusing design heuristics in the framework. Further exploration on locating the optimal burst length for a given application may be done in future work.

## 5. Conclusions

This work presents an approach that augments the decision-making of a data-driven visual design agent with a heuristic-guidance method. The predictive heatmap generated by an agent to anticipate a next-iteration design state is classified as a generalized design heuristic from a predefined set. This heuristic classification instructs agents to select the design change which continues the current design sequence in a way most closely resembling that heuristic's underlying principle. This heuristic-guidance method is applied to a truss design problem previously solved by human design teams and unguided deep learning agents. Heuristic-guided agents are able to outperform both of these groups in this design application. The incorporation of heuristics ensures that design changes that are made follow some form of multi-step, general strategic concept rather than greedily making decisions on an iteration-by-iteration basis. This allows agents to take advantage of solution exploration behaviours typical in human designers and illustrate its effectiveness in achieving higher performing designs.

While the goal of this work is to present a method that uses heuristic strategies to guide decision-making, it is important to note that this process does not, however, consider the specific manner in which these changes are applied. Therefore, this method is limited in its ability to distinguish similar sequences between two drastically different heuristic concepts. The deep learning agents themselves do mitigate this issue to an extent, as the actions and decisions are conditional on the current design state and therefore implicitly incorporate the effect of the order. To fully address this, a robust method for classifying the context in which design actions are taken by both agents and the predefined heuristic sequences would be required, which may be explored in future work.

## Acknowledgement

This material is based upon work supported by the Defence Advanced Research Projects Agency through cooperative agreement N66001-17-4064.

## References

- Ahmed, S., Wallace, K.M. and Blessing, L.T.M. (2003), "Understanding the differences between how novice and experienced designers approach design tasks", *Research in Engineering Design*, Vol. 14 No. 1, pp. 1-11. <https://doi.org/10.1007/s00163-002-0023-z>
- Athavankar, U.A. (1997), "Mental Imagery as a Design Tool", *Cybernetics and Systems, Taylor & Francis*, Vol. 28 No. 1, pp. 25-42. <https://doi.org/10.1080/019697297126236>
- Brooks, R.A. (1991), "New Approaches to Robotics", *Science*, Vol. 253 No. 5025, pp. 1227-1232.
- Campbell, M.I., Rai, R. and Kurtoglu, T. (2012), "A Stochastic Tree-Search Algorithm for Generative Grammars", *Journal of Computing and Information Science in Engineering*, Vol. 12, pp. 1-11. <https://doi.org/10.1115/1.4007153>
- Chakrabarti, A. et al. (2011), "Computer-Based Design Synthesis Research: An Overview", *Journal of Computing and Information Science in Engineering*, Vol. 11 No. 2, p. 021003. <https://doi.org/10.1115/1.3593409>
- Daly, S. and Christian, J.L. (2012), "Assessing design heuristics for idea generation in an introductory engineering course", *International Journal of Engineering Education*, Vol. 28 No. 2, pp. 1-11.

- Goldschmidt, G. (1992), "Serial Sketching: Visual Problem Solving in Designing", *Cybernetics and Systems*, Taylor & Francis, Vol. 23 No. 2, pp. 191-219. <https://doi.org/10.1080/01969729208927457>
- Hinton, G.E. and Salakhutdinov, R.R. (2006), "Reducing the Dimensionality of Data with Neural Networks", *Science*, Vol. 313 No. 5786, pp. 504-507. <https://doi.org/10.1126/science.1127647>
- Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983), "Optimization by Simulated Annealing", *Science*, Vol. 220 No. 4598, pp. 671-680. <https://doi.org/10.1126/science.220.4598.671>
- Königseder, C. and Shea, K. (2014), "Systematic rule analysis of generative design grammars", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 28 No. 03, pp. 227-238.
- Königseder, C. and Shea, K. (2016), "Visualizing Relations between Grammar Rules, Objectives, and Search Space Exploration in Grammar-Based Computational Design Synthesis", *Journal of Mechanical Design*, Vol. 138 No. 10, pp. 1-11. <https://doi.org/10.1115/1.4034270>
- Kramer, J. et al. (2014), "A case-study analysis of design heuristics in an upper-level cross-disciplinary design course", *ASEE Annual Conference and Exposition, Conference Proceedings*.
- Masci, J. et al. (2011), "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction", In: Honkela, T., Duch, W., Girolami, M. and Kaski, S. (Eds.), *Artificial Neural Networks and Machine Learning -- ICANN 2011*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 52-59. [https://doi.org/10.1007/978-3-642-21735-7\\_7](https://doi.org/10.1007/978-3-642-21735-7_7)
- 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", *Design Studies*, Vol. 36 No. C, pp. 99-121. <https://doi.org/10.1016/j.destud.2014.10.001>
- McComb, C., Cagan, J. and Kotovsky, K. (2017a), "Capturing Human Sequence-Learning Abilities in Configuration Design Tasks Through Markov Chains", *Journal of Mechanical Design*, ASME, Vol. 139 No. 9, pp. 91101-91112. <https://doi.org/10.1115/1.4037185>
- McComb, C., Cagan, J. and Kotovsky, K. (2017b), "Mining Process Heuristics from Designer Action Data via Hidden Markov Models", *Journal of Mechanical Design*, Vol. 139 No. 11, p. 111412. <https://doi.org/10.1115/1.4037308>
- McComb, C., Cagan, J. and Kotovsky, K. (2017c), "Utilizing Markov Chains to Understand Operation Sequencing in Design Tasks", *Design Computing and Cognition*, Vol. 16, pp. 401-418. [https://doi.org/10.1007/978-3-319-44989-0\\_22](https://doi.org/10.1007/978-3-319-44989-0_22)
- McComb, C., Cagan, J. and Kotovsky, K. (2018), "Data on the design of truss structures by teams of engineering students", *Data in Brief*, Vol. 18, pp. 160-163. <https://doi.org/10.1016/j.dib.2018.02.078>
- Miller, S.W., Simpson, T.W. and Yukish, M.A. (2015), "Design as a Sequential Decision Process: A Method for Reducing Design Set Space Using Models to Bound Objectives". <https://doi.org/10.1007/s00158-017-1756-7>
- Puentes, L., Cagan, J. and McComb, C. (2019), "Heuristic-Guided Solution Search through a Two-Tiered Design Grammar", *Journal of Computing and Information Science in Engineering*, pp. 1-38. <https://doi.org/10.1115/1.4044694>
- Raina, A., Cagan, J. and McComb, C. (2019), "Transferring Design Strategies From Human to Computer and Across Design Problems", *Journal of Mechanical Design*, Vol. 141 No. 11. <https://doi.org/10.1115/1.4044258>
- Raina, A., McComb, C. and Cagan, J. (2019), "Learning to Design From Humans: Imitating Human Designers Through Deep Learning", *Journal of Mechanical Design*, Vol. 141 No. 11, pp. 1-11. <https://doi.org/10.1115/1.4044256>
- Reddy, G. and Cagan, J. (1995), "An Improved Shape Annealing Algorithm For Truss Topology Generation", *Journal of Mechanical Design*, Vol. 117 No. 2A, pp. 315-321. <https://doi.org/10.1115/1.2826141>
- Schmidt, L.C. and Cagan, J. (1997), "GGREADA: A graph grammar-based machine design algorithm", *Research in Engineering Design*, Vol. 9 No. 4, pp. 195-213. <https://doi.org/10.1007/BF01589682>
- Shea, K. and Cagan, J. (1997), "Innovative dome design: Applying geodesic patterns with shape annealing", *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, Vol. 11 No. 05, pp. 379-394. <https://doi.org/10.1017/S0890060400003310>
- Stöckli, F. and Shea, K. (2017), "Automated Synthesis of Passive Dynamic Brachiating Robots Using a Simulation-Driven Graph Grammar Method", *Journal of Mechanical Design*, Vol. 139 No. 9, p. 092301.
- Tai, K. and Akhtar, S. (2005), "Structural topology optimization using a genetic algorithm with a morphological geometric representation scheme", *Structural and Multidisciplinary Optimization*, Vol. 30 No. 2, pp. 113-127. <https://doi.org/10.1007/s00158-004-0504-y>
- Yilmaz, S. and Seifert, C.M. (2011), "Creativity through design heuristics: A case study of expert product design", *Design Studies*, Elsevier Ltd, Vol. 32 No. 4, pp. 384-415. <https://doi.org/10.1016/j.destud.2011.01.003>
- Yukish, M.A., Miller, S.W. and Simpson, T.W. (2015), "A Preliminary Model of Design as a Sequential Decision Process", *Procedia Computer Science*, Elsevier, Vol. 44, pp. 174-183. <https://doi.org/10.1016/j.procs.2015.03.039>