BIGcad: Assisting 3D CAD Modeling with Workflow Graph-Driven Bayesian Command Inferences

Kyung Hoon Hyun[1*], Yugyeong Jang[2]

[1]Department of Interior Architecture Design, Associate Professor, Hanyang University, Seoul, Korea

[2]Department of Interior Architecture Design, Undergraduate Student, Hanyang University, Seoul, Korea

Abstract

Background Recent advancements in 3D generative artificial intelligence (AI) have streamlined design processes by enabling rapid model creation. However, these tools frequently lack the accuracy and comprehensive support needed for intricate real-world applications. Consequently, designers continue to depend on command-based computer-aided (CAD) tools such as Rhino, which provide the necessary accuracy but can impose high cognitive loads. To address these challenges, we introduce BIGcad:a workflow graph-driven system that leverages Bayesian inference to optimize 3D CAD modeling, thereby enhancing both efficiency and precision.

Methods BIGcad encodes 3D modeling sequences using a Workflow graph (W-graph) and integrates a Bayesian information gain (BIG) framework to infer user intentions. By analyzing user interactions and modeling data, the system predicts subsequent steps in the modeling workflow. Implemented as a Rhino plugin, BIGcad captures command logs and snapshots in real time, providing guidance that reduces cognitive load and improves overall design efficiency.

Results The implementation of BIGcad yielded promising results, particularly in improving workflow efficiency and lowering cognitive demands. Participants reported streamlined processes, particularly during complex modeling tasks, while the visualized W-graphs provided valuable insights into alternative workflows. This approach not only reduced errors but also fostered the exploration of creative modeling strategies, underscoring the system's potential to advance design processes.

Conclusions BIGcad introduces a new approach to improving 3D CAD modeling by integrating workflow visualization and command recommendations based on user behavior. The system not only enhances modeling efficiency and accuracy but also provides opportunities for creative exploration and learning. Future efforts will focus on expanding dataset diversity and enhancing personalized features to further optimize their utility in design processes.

Keywords 3D generative AI, Computer-Aided Design, 3D Modeling Workflow, Computational Design, Design Command Inference, Bayesian Information Gain

This work was supported by National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT; RS2023-00208542), and partially supported by Seoul R&BD (ProgramCY230112).

Kyung Hoon Hyun and Yugyeong Jang contributed equally to this work.

*Corresponding author: Kyung Hoon Hyun (hoonhello@hanyang.ac.kr)

Citation: Hyun, K. H., & Jang, Y. (2025). BIGcad: Assisting 3D CAD Modeling with Workflow Graph-Driven Bayesian Command Inferences. *Archives of Design Research, 38*(2), 179-199.

http://dx.doi.org/10.15187/adr.2025.05.38.2.179

Received : Feb. 06. 2025 ; **Reviewed :** Mar. 24. 2025 ; **Accepted :** Apr. 10. 2025
pISSN 1226-8046 **eISSN** 2288-2987

1. Introduction

The approach to design has continuously evolved, reflecting significant changes in the work methods and environments of designers (Faruqi et al., 2023; Liu et al., 2023; Veuskens et al., 2021). This evolution is particularly evident with the advancement of research in 3D Generative AI, which has actively facilitated studies in this area. Such studies have focused on providing direct support for the design process. One of the key innovations in this area is 3DALL-E, a system that generates 2D image inspiration for 3D design (Liu et al., 2023). Designers have recognized the significant potential of incorporating 3DALL-E into their workflow. Specifically, they utilize Text-to-Image AI for creating reference images, renderings, materials, and other design references. This integration facilitates the design process by leveraging the capabilities of 3D generative AI (Liu et al., 2023). Furthermore, SPAGHETTI (Hertz et al., 2022) and SALAD (Koo et al., 2023) have enabled the rapid generation, synthesis, and exploration of 3D mesh models. By enabling part mixing and refinement at the part level, these technologies facilitate shape editing and manipulation. However, 3D models generated by 3D Generative AI are predominantly mesh-based, making it difficult to make detailed design modifications and implement complex surfaces or detailed structures. Such models pose challenges for precise modifications and detailed structural implementations—crucial aspects for real-world applications where exact dimensions and geometric fidelity are non-negotiable. Therefore, transforming mesh-based 3D models with generative AI has been highly valued for providing inspiration in the initial stages of design or creating assets used only in the digital world.

The necessity for precision in manufacturing and fabricating 3D objects underscores the importance of controlling all surfaces to achieve exact dimensions and geometric shapes (Schulz et al., 2017). This affects not just the visual appeal of an object but also its functionality, compatibility with other components, and overall performance. Modeling tasks requiring such complex and precise outcomes involve a significant mental load and time cost. Moreover, there can be various approaches to creating the same 3D model. While some people may go through efficient commands and modeling processes for faster and more accurate modeling, others may undergo less efficient modeling processes. This suggests the existence of an optimal modeling sequence that offers a more efficient alternative to those using less efficient methods, thus reducing the cognitive load and time cost for the designer and enabling them to experience an efficient modeling process. That is, if it is possible to infer what the user wants to create, it is possible to recommend the necessary commands and sequences for the most efficient modeling process tailored to the user's modeling process. Such support for efficient modeling is essential in extending the stage of design support, but research on developing efficient systems that can maximize the modeling experience is still lacking.

To extend support for the design process to the middle and later stages, we propose an inference model based on user behavior. BIGcad provides a workflow along with commands to efficiently proceed with the modeling process to achieve the desired design outcome, through an inference algorithm predicting what the user wants to create. In addition, to support the conventional design process, we have proceeded based on Rhinoceros7 (hereafter Rhino7), a modeling tool that supports precise modeling actively used in the architectural and product modeling industry by compensating for the disadvantages of polygons.

The main challenges of this study are as follows:
1) Developing a Rhino plugin to record user-entered commands and collect 3D CAD modeling sequence data.
2) Application of a W-graph with added command nodes for multiple fixed modeling tasks.
3) Measurement of similarity between 3D models for clustering of nodes in the W-graph.
4) Utilizing the BIG (Bayesian Information Gain) framework to infer 3D modeling processes.
5) Experimenting with a modeling sequence recommendation system.

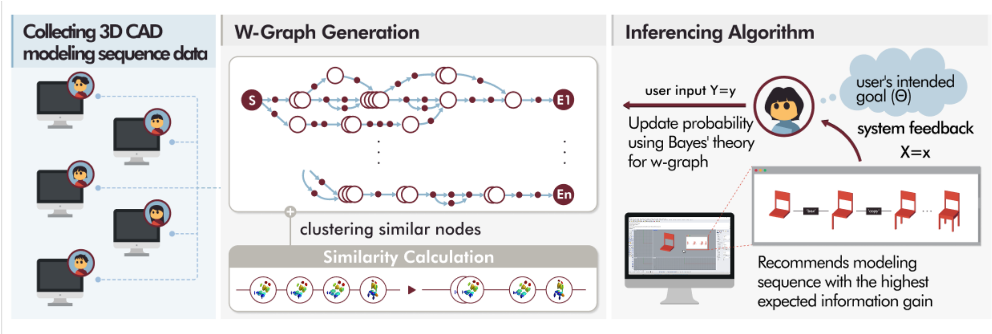


Figure 1 Overview of BIGcad

2. Related Works

2. 1. Design aid for 3D modeling

Research supporting the design process of 3D objects has been continuously and actively conducted. Marzal et al. (2020) proposed a method through Computer-Generated Modeling (CGM) that allows designers to effectively visualize and transform sketches and ideas (Alcaide-Marzal et al., 2020). This approach facilitates the design process by allowing for rapid exploration and experimentation with various design options, thus enabling easy adjustment and improvement of complex structures and forms. As a result, designers are able to create more creative and innovative design outcomes. However, CGM requires understanding and utilizing complex algorithms and calculations, which may necessitate significant time and learning for users to fully grasp. Additionally, this technology sometimes leads to excessive computational load due to its use of Grasshopper to provide design diversity, which can decrease the efficiency of the design process.

The utilization of 3D generative AI in research has notably advanced, seamlessly integrating into the workflows of designers to enhance the design process. Notably, methodologies such as SPAGHETTI (Hertz et al., 2022) and SALAD (Koo et al., 2023) have revolutionized design diversity by facilitating the mixing and interpolation of disparate 3D object segments at a granular part level. SPAGHETTI decomposes shape embedding z into distinct 3D part embeddings corresponding to a Gaussian Mixture Model (GMM) through a Decomposition Network, obtains mixed embeddings via a Mixing Network, and determines the final 3D shape in an Occupancy Network (Hertz et al., 2022). Thanks to the Decomposition Network, it enables part-level control by applying transformations to local areas and mixing and interpolating different shape segments. SALAD's method, providing more refined 3D shapes and adding text-guided part completion features, thus offering more advanced editing capabilities for finely mixing and adjusting parts (Koo et al., 2023). While these innovative studies bolster the initial phases of the design process through the provision of diverse design perspectives via 3D Generative AI, they encounter limitations in aiding the middle and terminal stages of actual design production and realization. Consequently, there is a pressing need for methodologies that augment the efficiency of traditional CAD systems, which are pivotal for the accurate construction of 3D models, to ensure comprehensive support throughout all stages of the design process.

Despite the significant advancements in graphics and machine learning-based tools for 3D modeling, they often present challenges such as high complexity, steep learning curves, and substantial computational demands. Moreover, these tools primarily enhance the initial phases of the design process, leaving a gap in support for the middle and final stages of actual design production. Consequently, there is a need for more comprehensive and user-friendly methodologies that can seamlessly support designers throughout the entire 3D modeling process.

2. 2. Design aid in 3D Modeling within HCI

To enhance the efficiency of 3D modeling tasks, HCI researchers have developed various interfaces and support systems. The Style2Fab system proposes a method for automatically classifying the functional and aesthetic elements of 3D models. By utilizing spectral segmentation and clustering techniques, it segments the model and allows users to modify aesthetic elements through text-based style specifications, effectively preserving the model's functionality while enhancing its aesthetic aspects (Faruqi et al., 2023). Chaudhuri et al. (2013) introduced ATTRIBIT, a system that facilitates the creation of visual content using semantic attributes. By leveraging relative semantic attributes learned from crowdsourced data, ATTRIBIT allows users to interactively explore and modify 3D models in real-time using intuitive linguistic commands. This approach enables designers, especially novices, to achieve high-level design goals more efficiently compared to traditional modeling tools. However, ATTRIBIT's reliance on predefined databases and its focus on relative attributes limits the breadth of design variations and the specificity of user intent that can be captured. Hofmann et al. (2018) developed the PARTs (Parameterized Abstractions of Reusable Things) framework to assist non-experts in clearly expressing and easily reusing design intentions in the 3D modeling process. This framework enhances the accessibility and efficiency of modeling by providing a graphical user interface (GUI), scripting API, and an example library to test and implement design expectations. The usefulness of PARTs was demonstrated through two workshops, showing that even non-experts can easily utilize advanced 3D modeling features. While these modeling support programs are valuable, they do not fully support the entire modeling process from inception to completion and often require users to input and concretize 3D models. To seamlessly support the user's modeling environment, there is a fundamental need for systems that comprehensively understand the modeling process and provide direct assistance to users throughout.

History in Motion leverages the modeling history of users to uncover modeling functions related to geometric elements chosen by designers and introduces a feature that employs new 3D interactive animations (Veuskens et al., 2021). These animations visualize how these modeling functions interact and are implemented. Such support is instrumental in deepening designers' understanding of modeling functionalities and facilitating their modification. Nonetheless, this focus is primarily on comprehending and navigating the design history of existing models rather than offering a real-time interactive experience. Essentially, the system contributes to supporting the understanding of the modeling sequence by identifying which modeling commands input by the user had the most significant impact on creating a particular 3D shape. CODA enhances the modeling process by automatically identifying potential constraints that could be applied during the creation or modification of a model (e.g., constraining two points to be in the same location), thus providing feedback and optimization to minimize errors (Veuskens et al., 2021). This mechanism enables users to improve the accuracy and consistency of their models. However, while such systems concentrate on assessing the history or constraints of modeling states, they do not support comprehensive strategic planning for accurate 3D modeling and exploring 3D shape variations during model creation.

Schulz et al. (2017) conducted research that helps not only experts efficiently search the design space but also lowers the design barrier for general users by proposing a system that uses parametric CAD data to explore and optimize design shapes in real time. By employing pre-calculated data and a new interpolation method, they developed a tool that allows for quick adjustment of various parameters and real-time model optimization. This tool is designed to make the design process more efficient and user-friendly for mechanical engineers and designers dealing with complex parametric forms by enabling easy adjustment and improvement of objects with various physical properties. Furthermore, Chang et al. (2020) introduced the concept of W-graph, which encodes the similarities and differences in their approach to performing defined 3D design tasks into a graph. This research aims to support the 3D modeling learning process for designers and suggest new modeling methods to experienced users through a system. The structure of the W-graph consists of Graph Vertices and Graph Edges. Graph Vertices represent the state capturing the moment after the user has completed a specific subtask but before starting the next subtask, like partial goals of a task, including metadata (command log data, screen recording, scene graph snapshot; Jang & Hyun, 2024). Graph Edges represent the transition between two states, and multiple directional edges may exist if the same state transition is included in the modeling processes of various users. Each edge includes event data (including timestamped command invocations). In essence, the W-graph represents potential workflows from the start node to the end node through each directional edge, and multiple edges between any two states suggest alternative approaches for that task segment. Such W-graphs can serve as the backend of applications that provide users with the foundation to explore alternative workflows by representing the structure of a workflow. Consequently, potential applications that can be implemented through W-graph, such as W-Suggest, W-Guide, W-Instruct systems, are presented, demonstrating the possibility of supporting the design process through W-graph.

In this study, we go beyond simply detecting commands from usage history and present a system that understands user behavior and infers user actions in real-time to recommend the user's next steps. While previous systems have primarily focused on analyzing modeling history, attribute-based modification, or parameter control, BIGcad advances this lineage by providing behavior-driven, real-time modeling support. We leveraged the structure of the W-graph, recognizing its strength in capturing the entirety of the modeling process, and used it as the back-end for our recommendation system.

2. 3. Bayesian Information Gain Framework

The BIG (Bayesian Information Gain) framework aims to model the acquisition of information between user input Y, system feedback X, and the user's intended goals Θ, with the goal of optimizing information acquisition through interactions between the user and the computer. When users wish to select a goal from among potential targets Θ, each target Θ is associated with a probability $P(\Theta=\theta)$ representing the computer's knowledge about the user's goal. The system provides system feedback X to the user and receives user input Y. It then updates the user behavior function $P(Y=y|\Theta=\theta,X=x)$ for each Θ and provides system feedback X that maximizes the expected information gain. In essence, this framework is designed to adjust system feedback to maximize expected information and make interactions with the user more efficient. The BIG framework has been utilized in numerous studies to support the search process by inferring targets of interest based on user behavior (Son et al., 2022; Liu et al., 2017, 2018; Son et al., 2022; Son et al., 2022). For instance, BIGnav interprets user inputs to update the view that maximizes expected information gain, thereby assisting user navigation (Liu et al., 2017).

Liu et al. (2018) designed BIGFile using the Bayesian Information Gain (BIG) framework to enhance the efficiency of file search by providing shortcuts that reduce the number of steps (user inputs) to reach the target file or folder, thereby improving the efficiency of navigation-based file search. After receiving user input, the BIGFile algorithm updates $P(\Theta=\theta)$ using Bayes' rule and calculates and selects the target with the highest possible information gain from all combinations of potential target nodes N. This maximizes the expected information gain for the user's next input and optimizes the file search path. BIGFile updates probabilities based on a binary tree structure and recommends several options from the adaptive area A that provide the maximum expected information gain due to the user's subsequent actions, making the navigation process more efficient and faster.

In this study, we applied the structure of the BIG (Bayesian Information Gain) framework to our system by designing a recommendation mechanism that calculates the expected information gain based on the user's current modeling state and suggests the next command sequence accordingly. Drawing inspiration from the hierarchical structure of the BIGFile algorithm, we partially converted the complex node structure of the W-graph into a tree-like form and configured the system to explore the path with the highest expected information gain among multiple possible goal states.

Bayesian inference provides a theoretically robust foundation for modeling user intentions and uncertainty, and it is particularly effective in environments with limited data. However, recent advances in learning-based approaches have led to the application of alternative methods in CAD modeling support systems(DiPrete et al., 2023; Mandelli & Berretti, 2022; Quan et al., 2024). For instance, Graph Neural Networks (GNNs) are well-suited to learning the relational structure of W-graphs and predicting future modeling states, while Reinforcement Learning (RL) is effective in learning optimal command sequences based on user feedback.

Although these learning-based approaches offer advantages in capturing complex user behaviors, we adopted the BIG framework for the following reasons:

First, it provides an interpretable and transparent recommendation logic that clearly reflects the system's design intent.

Second, it can operate without the need for large-scale training, making it suitable for early-stage system environments.

Third, it is well-suited for command-based CAD tasks where data is inherently limited.

3. Proposed Method

3. 1. Overview

BIGcad aims to support an efficient modeling process by inferring the user's intended target and visualizing the modeling process for the stage with the highest expected information gain at the current modeling step. It uses a W-graph (workflow graph) based on 3D modeling data as the system's backend and generates the system through an inference algorithm based on this structure. The system undergoes the following process:

(1) W-graph generation: Collects modeling sequence data from multiple users performing a total of three fixed tasks. This data is encoded into a W-graph, consisting of nodes representing 3D model snapshots and directed edges containing command information.

(2) 3D model similarity calculation: Performs part-level similarity analysis between 3D models to merge identical nodes from a single user's sequences and create equivalent-intermediate nodes observed in the modeling sequences of multiple users during the W-graph generation phase.

(3) Inferencing Method: Based on the W-graph, when users model a 3D object, it predicts the design outcomes with high expected information gain from the current state and supports the modeling process by suggesting the modeling sequence to reach that stage.

(4) Interface: Visualizes the modeling sequence from the current stage to the proposed modeling stage through the W-graph and provides it as a plugin for Rhino 7.

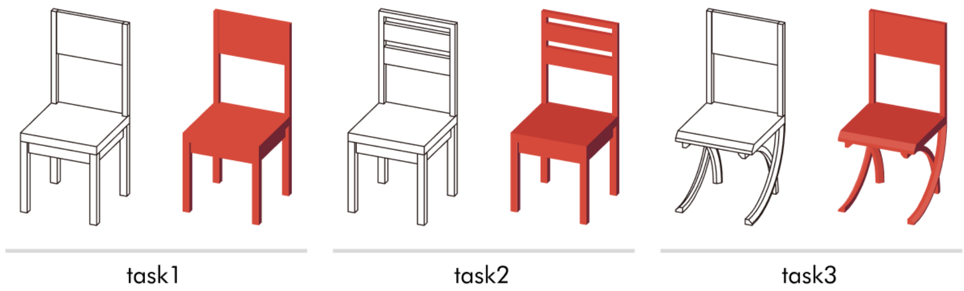


Figure 2 3D chair model of the designed tasks for the data collection experiment

3. 2. W-graph

3. 2. 1. Components of W-graph.

The W-graph consists of Graph Vertices (V) and Graph Edges (A). Graph Vertices represent the modeling state immediately after the user has completed a command but before entering the next one. This includes a snapshot of the 3D object at that state. Graph Edges signify the workflows used by users to connect semantically-similar states. When transitions from V_i to V_j are included in multiple modeling sequences, multiple directed edges can exist between the two nodes. Each directed edge contains information about the specific command used by the user.

3. 2. 2. W-graph Generation.

To collect modeling sequence data from multiple users for fixed tasks, three 3D objects were designated as specified tasks, as shown in Figure 2. To identify similarities and differences in modeling sequences within the graph, each task was chosen to involve a chair that includes some common design elements across all tasks. For example, task 1 and task 2 share the same design elements except for the shape of the chair backrest, implying that the common modeling steps are shared for elements other than the backrest in both design outcomes. By encoding the modeling process for each task into a W-graph, it is possible not only to observe various modeling processes within a limited data collection but also to easily identify the equivalent intermediate nodes, which are an advantage of the W-graph. For the experiment, design students proficient in 3D modeling with Rhino7 were recruited. For each task, five experimenters, totaling 15 participants (8 male, 7 female, ages 20–24, average Rhino usage duration = 21.2 months), conducted the modeling. Each experimenter freely conducted the modeling after checking the exact dimensions of the 3D model provided in the instructions through iso view and three views (top, front, left view).

To collect the modeling sequence data from each experimenter, a plugin for Rhino7 was developed. The plugin records the command, timestamp, and snapshot each time the user enters a command. The recorded snapshots and commands were used to encode the W-graph, referencing “Workflow to Graph Construction” (Chang et al., 2020).

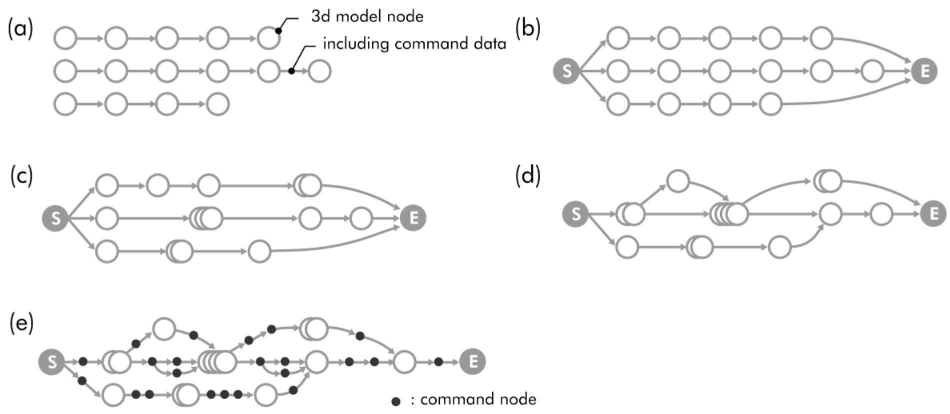


Figure 3 Illustration of how sequences get compressed and merged into a W–graph

Step 1. Preprocessing The state capturing the moment right after a user completes a specific subtask but before starting the next subtask is considered as an event node, and the event sequence of each user is regarded as a set of nodes (one node per event). There are directed edges connecting each event in timestamp order (Figure 2-a). Since each demonstration starts from a blank document, the first node of each demo is merged into a START node (Figure 2-b). Each demo represents a distinct directional path from the START node to the END node.

Step 2. Collapsing Node Sequences Nodes with similarity below a threshold in each event sequence are merged through the calculation of similarity between nodes (Figure 3). This process is explained in '3.3.Generative AI-Driven 3D Model Similarity Calculation Method'. For example, if two nodes in a single user's sequence represent geometrically similar states, they are merged and treated as equivalent. Also, command data associated with the edges connected to the merging nodes are added to the incoming directed edge of the node before merging.

Step 3. Sequence Merging In this step, an "equivalent-intermediate" node representing the middle stages of the task is created, which is useful for representing a common intermediate state appearing in the work of different users (Figure 3-d). To achieve this, the similarity among nodes from different users is compared, similarly utilizing the '3.3. Generative AI-Driven 3D Model Similarity Calculation Method'. For instance, if multiple users create a similar 3D model midway through their modeling, those nodes are merged.

Step 4. Adding Command Nodes The workflow graph, which includes the command corresponding from a specific node to the next node and incorporates the number of demonstrations for each path in the directional edge, replaces the edge (Figure 3-e). At this point, the last node of each event node, i.e., the last nodes in the graph, are designated as end nodes (task1, task2, task3).

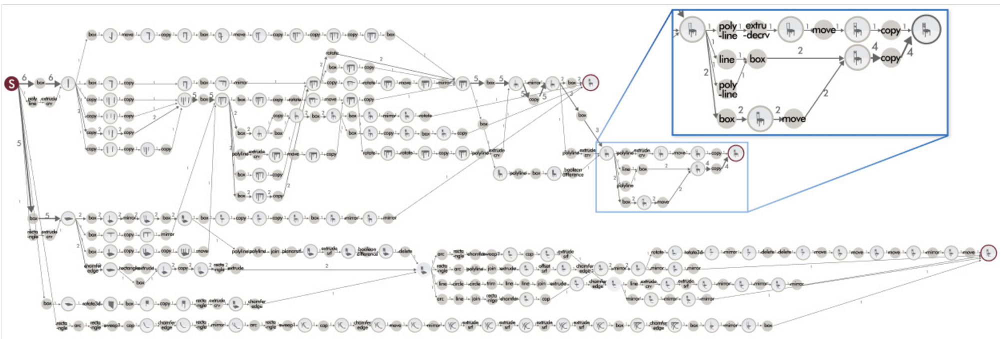


Figure 4 W-graph encoding the modeling process of Tasks 1, 2, and 3

Upon completing these steps, the modeling sequence data from multiple users for the specified tasks (tasks1,2,3) derived from the data collection can be encoded into a W-graph (Figure 3). The W-graph generation method described earlier can be applied to other 3D environments besides Rhino. Since the creation of a W-graph is dependent on the modeling sequence data collected, it can be infinitely expanded based on the collected data and is not restricted to any specific 3D modeling environment.

3. 3. Generative AI-Driven 3D Model Similarity Calculation Method

During the W-graph Generation phase, part-level similarity comparison was used as a method to cluster similar nodes by verifying the similarity between sets of nodes. Through the Decomposition Network stage of SPAGHETTI, each node's 3D model was decomposed into part-level, and the similarity of each node was determined by calculating the Euclidean distance between each part (Hertz et al., 2022). The Decomposition Network stage involves decomposing the shape embedding z into 16 part-levels of z_{id}. Through this process, as shown in Figure 5-a, the 3D model of a snapshot node is decomposed into 16 Gaussian Mixture Models (GMMs). Each part's z_{id} is represented by extrinsic geometric information (g_i), indicating the location information of each Gaussian blob, and intrinsic geometric information(s_i), indicating the shape's characteristics. The set of extrinsic parameters is expressed as $e_i = \{c , \lambda_{i1}, \lambda_{i2}, \lambda_{i3}, u_{i1}, u_{i2}, u_{i3}, w_i\}$, where each parameter consists of the blending weight $w_i \in R^1$, the mean of GMM $c_i \in R^3$, eigenvalues and eigenvectors of covariance matrix $u_{ij} \in R^3$, $\lambda_{ij} \in R^1$. Since similar intra-category parts in the dataset are consistently represented using the same Gaussian blob, it is suitable as a representation for calculating the similarity between nodes. The part-level similarity was determined using one of g_i's parameters, c_i, which indicates the center location of each Gaussian blob. Specifically, by calculating the Euclidean distance in a one-to-one correspondence with the Gaussian blobs of another node for the aligned 16 Gaussian blobs, and averaging the Euclidean distances for the 16 GMMs, nodes with an average distance below a manually specified threshold were considered as the same 3D model and merged (Figure 5-b).

This methodology precisely calculates the similarity between 3D models, providing essential data for clustering nodes and creating equivalent-intermediate nodes in the W-graph. This allows the system to make more accurate recommendations for various 3D designs and to more finely reflect user requirements. Additionally, by effectively analyzing and distinguishing between different types of 3D models, the method enables the system to expand into a broader 3D model database. This allows the system to efficiently respond to a larger number of users and diverse 3D modeling needs, lowering technological barriers and fostering creative design. This similarity calculation method facilitates continuous improvement and expansion of the system, contributing to providing users with continuously new and diverse modeling options.

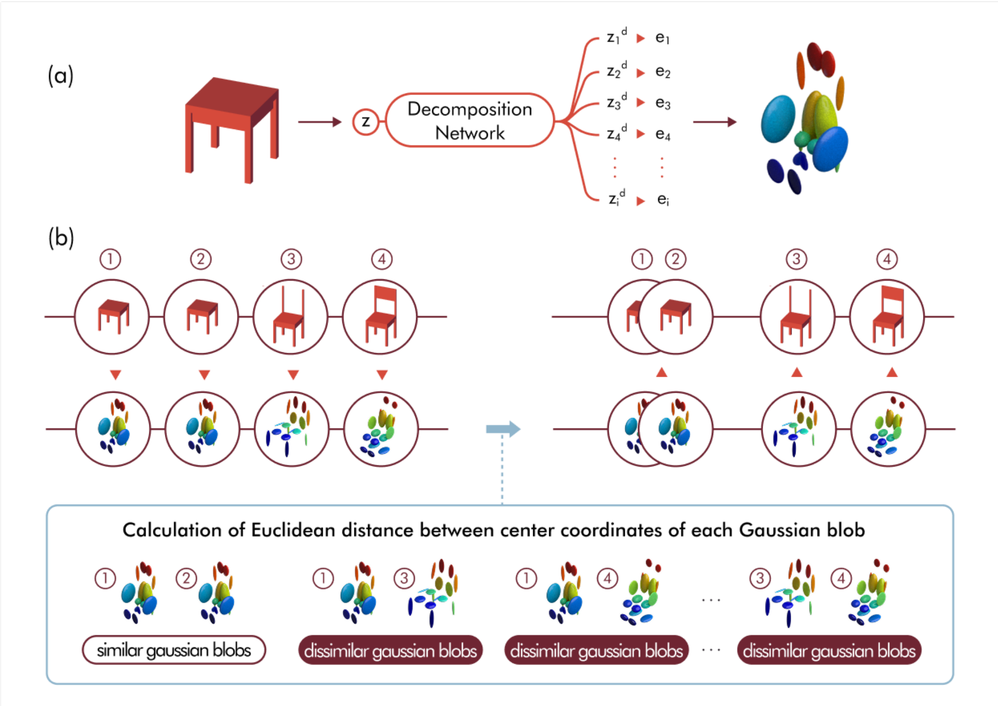


Figure 5 3D model similarity calculation process

3. 4. BIGcad Algorithm

User intention inference is based on the BIG algorithm, specifically utilizing the BIGFile algorithm, which employs a tree structure representing hierarchical structures. Directly applying the BIGFile algorithm to BIGcad is problematic due to the diverging and converging structure of the W-graph, which does not match the tree structure used by the BIGFile algorithm. To address this, after updating the probabilities, we reinterpret the portion of the W-graph between the current node and its next converging node as a tree-like substructure, denoted as G'. The expected information gain is then calculated over the nodes within G' to provide system feedback. This approach allows us to retain the theoretical strength of the BIGFile method while adapting it to the non-hierarchical structure of real modeling workflows.

Θ represents all possible target nodes,aiming to identify the node corresponding to the user's intended design targe. The probability $P(\Theta=\theta)$ for all possible target nodes $\Theta=\theta$ represents the probability when the actual target desired by the user is Θ, calculated and updated based on user behavior and interaction history. With each user input, this probability is updated according to the Bayes rule.

X represents the set of all modeling sequences up to the recommended modeling step presented to the user (x represents the modeling sequence for an individual node), including the current node C and the intermediate stage M with the highest expected information gain to reach N recommended modeling steps selected by the BIGFile algorithm.

At each step, the user provides user input $Y=y$, which signifies the action of entering a command to proceed to the next mode. Fitchett and Cockburn (2015) describe the user's uncertainty when moving files based on the Folder Uncertainty Ratio (FUR) presented by Elsweiler et al. (2011). They set the correct user input rate at 94%, distributing the remaining 6% among other inputs. Fitchett and Cockburn (2015) show that users are accurate about 94% of the time, while in the remaining 6%, they may click on the wrong folder. Accordingly, the probability $P(Y=y|\Theta=\theta,X=x)$ of the user seeing $X=x$ and providing the correct input $Y=y$ is set at 0.94, and otherwise at 0.06. C represents the node with the lowest Euclidean distance when the user has successfully entered a command and all nodes are calculated using the 3D Model Similarity Calculation method. After each user input, C is updated. Then, from the set of N' nodes in G', the node with the highest expected information gain to reach the N probable nodes selected by the BIGFile algorithm is calculated and recommended, supporting the user in the modeling sequence to reach the target node.

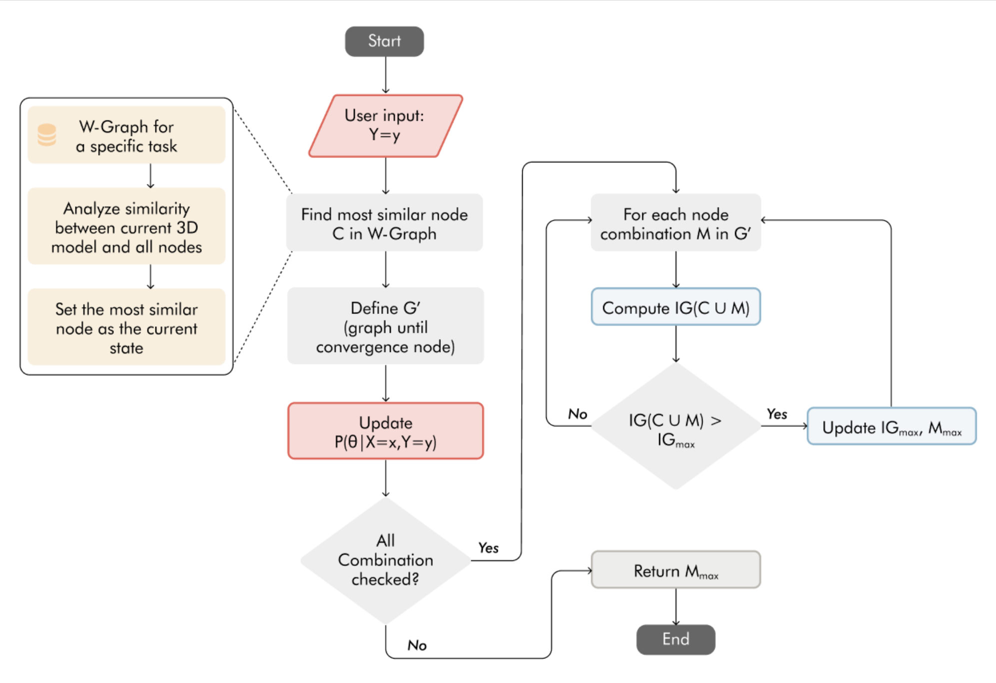


Figure 6 Flowchart of the BIGcad algorithm

Algorithm 1: BIGcad Algorithm

Data: $\Theta, X, Y, P(Y = y|\Theta, X = x), IG_{\max} = 0$

Result: Return set M that, together with set C, has the maximal expected information gain (IG).

Receive user input $Y = y$

Find most similar node (within W-graph) C

Update the probability distribution of Θ (Bayes rule): $P(\Theta|X = x, Y = y) = \frac{P(Y=y|\Theta,X=x)P(\Theta)}{P(Y=y|X=x)}$

for *all combinations of nodes in the node set* N' *of* G' **do**

  Compute $IG(C \cup M) = I(\Theta; Y|X(C \cup M)) = H(\Theta) - H(\Theta|X(C \cup M), Y)$

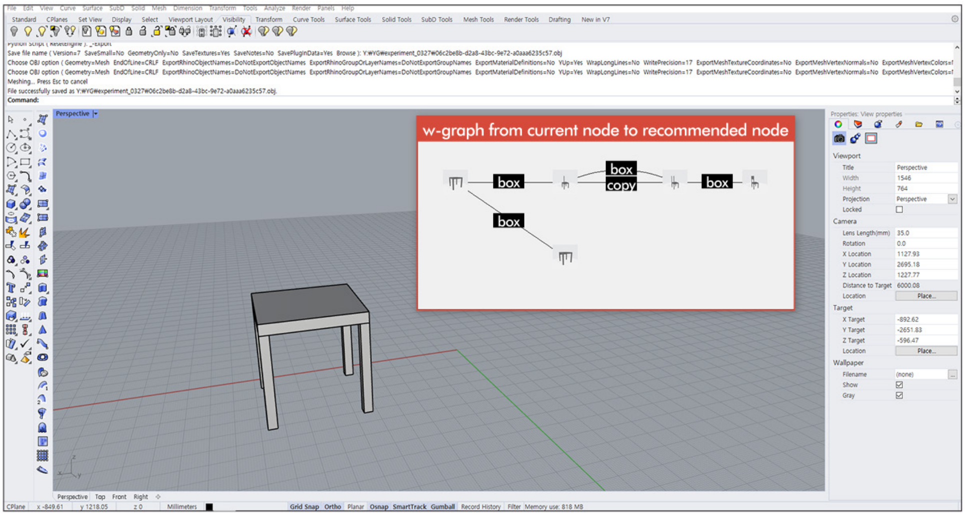
  if $IG(C \cup M) > IG_{max}$ **then**

    $IG_{\max} = IG(C \cup M)$

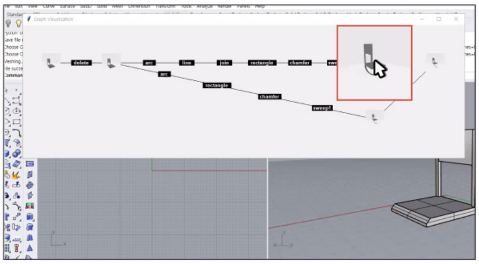
    $M_{\max} = M$

return M_{max}

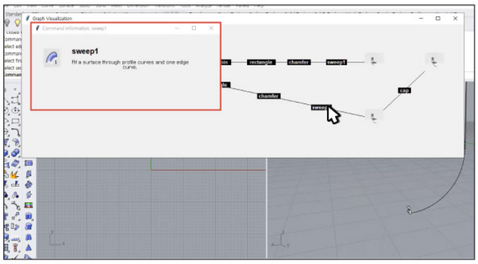
Figure 7 BIGcad algorithm



(a) Within the interface, you can check the recommended design target and recommended modeling sequence for the current stage.



(b) Hovering over a 'node' image enlarges the image.



(c) Clicking on a command allows you to check the description of the command.

Figure 8 The system interface

3. 5. Interface

This system employs the 'Generative AI-Driven 3D Model Similarity Calculation Method', analyzing the user's current 3D model state to compare similarities and identify the current modeling stage. Based on the user's actions (command input), it provides a modeling sequence and commands to reach the recommended design outcome, supporting the modeling process. To avoid confusion that might arise from the recommended stage being too far from the current stage, which could make it difficult for users to follow the modeling process, the system visualizes only up to three steps ahead from the current stage. This not only allows users to see the upcoming progression but also ensures clear communication.

The system interface, utilizing the W-graph, displays the path from the current to the recommended stage, thereby offering users various modeling process options. The equivalent-intermediate nodes in the W-graph are useful for identifying common modeling stages that many people go through. Additionally, by checking the number of commands added to the command nodes on the edges, users can choose a more simplified method to reach a specific stage. By clicking on a command node, users can view a description of its function, learning about efficient yet frequently used commands(Figure 6-b). This interface design enables users to see alternative modeling approaches through multiple edges connected to the current state and helps in understanding the modeling sequence and command functions (Figure 8-a).

4. Implementation and Results

4. 1. Experimental Design

To assess BIGcad's efficacy in inferring the 3D object modeling process and recommending modeling sequences that align with the designer's workflow, we engaged four participants skilled in Rhino usage (1 male, 3 female, ages 21-22, average Rhino usage duration = 36.5 months).The experimental procedure was structured as follows: introduction, consent, and tutorial (lasting approximately 10 minutes) → Modeling 1 (about 15 minutes) →Modeling 2 (about 15 minutes) → Modeling 3 (about 15 minutes) → In-depth interview (20 minutes). Each experimenter was assigned the same tasks to create the 3D objects from the previously collected data for task1, task2, and task3, and they were able to check BIGcad's recommendations during the modeling process. During each experiment, the experimenter's command inputs and 3D snapshots were recorded, and a recommendation window for the recommended modeling sequence was shown each time the user successfully executed a command input. After completing the three modeling tasks with BIGcad, an interview was conducted to collect feedback on the overall experience of modeling with this system. Specifically, feedback was gathered on the effectiveness of BIGcad and the accuracy of its inferences and recommendations, and the advantages and disadvantages of the system were compared to the baseline of using Rhino alone.

4. 2. Results and Discussions

After completing the modeling tasks, we conducted in-depth interviews with each participant to explore their modeling experience using BIGcad. The interviews followed a semi-structured format, allowing participants to freely express their thoughts while addressing key questions regarding system usability, recommendation accuracy, and the perceived advantages and limitations of the system. It should be noted, however, that the number of participants was limited to four, and thus the generalizability of the results may be limited.

4. 2. 1. Evaluation of User Support.

All experimenters responded that the system recommends appropriate modeling processes based on accurate inference of modeling intentions. They particularly mentioned that the system's efficiency is maximized for complex modelings, emphasizing the simplification of the modeling process. P2 and P4 stated that task 3 is more efficient than tasks 1 and 2, indicating that as the complexity of commands increases, they rely more on the system, which ultimately enables more efficient modeling. Indeed, compared to the time taken for modeling to collect modeling data (baseline), the use of BIGcad resulted in a decrease in time by 43.9%, 54.7%, and 43.6%, respectively. It appears that by following the system's recommendations, users experience reduced mental load to progress to the next step and spend less time modifying specific tasks.

- P1: "I think it can reduce mistakes that occur during modeling. Also, in terms of reducing the time a user spends thinking, I felt it could be used more efficiently while modeling."
- P4: "I felt that the system streamlined the design process. I received more help when modeling the more complex task3. For example, while I might follow my usual habits for simple steps like 'copy'-'rotate' or 'mirror', for more complex modeling processes like the leg design in task3 (especially when the use of commands becomes complex), I found efficiency and reduced mistakes by following the system's recommendations."

Figure 9 Representative user quotes on modeling efficiency, creativity, and error reduction

During the 12 modeling sessions conducted by the four experimenters, each involving three tasks, a reduction in uncertainty was observed in all cases. (BIGcad achieved an average information gain of 0.52, 0.56, 0.60, and 0.53 bits per user input for participants P1, P2, P3, and P4, respectively. Moreover, all commands contributed to reducing uncertainty by an average of 0.60, 0.61, 0.66, and 0.57 bits, respectively.) Notably, an increase in Information Gain was observed when selecting one of multiple edges at a step in the W-graph. Figure 8 depicts the Uncertainty and Information Gain graph for Tasks 1, 2, and 3 executed by Participant 4. It is evident from the graph that during certain stages of Task 3, the Uncertainty diminishes to a significantly low value(Figure 11-(c)). This is because the current node corresponding to the user input utilized a command that did not affect the 3D object, resulting in no change to the object and thus being considered the same node as the previous one. In such cases, Information Gain was also found to be close to 0.

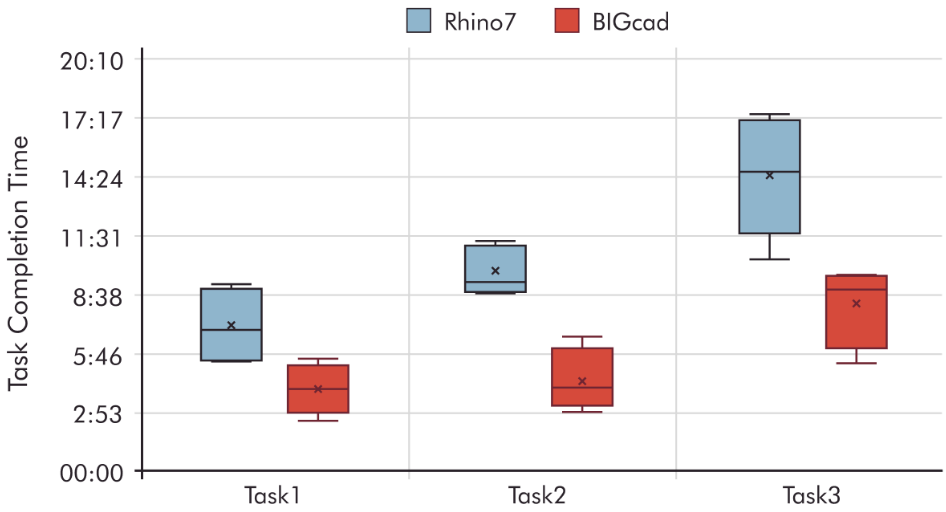


Figure 10 Task Completion Time

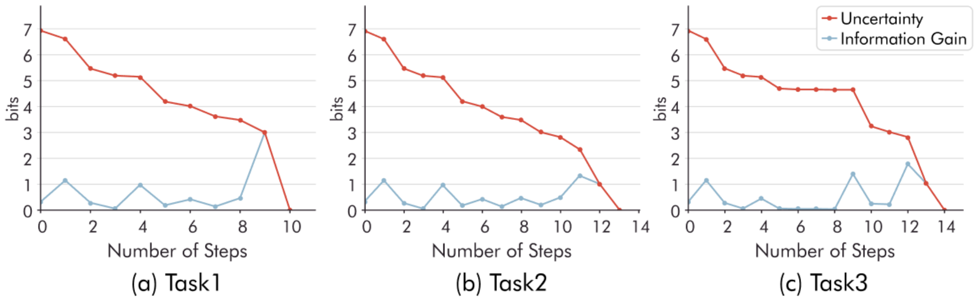


Figure 11 Uncertainty and information gain after each step

4. 2. 2. Evaluation of Visualization Using the Workflow Graph.

All participants responded positively to the visualization provided by the W-graph.They noted that being able to reference various modeling methods available from the current step, as depicted in the W-graph, enhanced the efficiency of the modeling process. P1 specifically mentioned that consulting the graph helped them identify the most efficient modeling process and avoid unnecessary steps. Furthermore, 1, 2, and 3 stated that the graph enabled them to explore and experience creative modeling methods they had not previously considered. Such experiences can ensure efficiency in the user's modeling process and improve the user's chronic and unnecessary modeling habits by discovering new modeling methods.

- P1: "I liked being able to choose steps that could be done faster by looking at the various modeling methods shown in the graph, which allowed me to avoid inefficient steps."
- P2: "The more complex the modeling, the more I felt the efficiency. Additionally, it was beneficial to see more than one option (node)."
- P3: "By providing multiple options, the system allowed me to work creatively without being limited to my usual modeling habits."

Figure 12 Representative user quotes on modeling efficiency and creative exploration

4. 2. 3. Proposal for novel 3D Modeling Interface.

Building on the advantages of BIGcad discussed in sections 4.2.1 and 4.2.2, some experimenters expressed a desire for BIGcad's features to be integrated into the Rhino interface. Specifically, P1 commented, "This system would be helpful when the user knows what design outcome they want but is unsure how to achieve it or has forgotten the commands needed. If implemented within the Rhino interface, I believe it would make the modeling process more efficient." This comment points to the potential for changes within the Rhino interface. Such a demand signifies a shift towards a new paradigm in 3D modeling interfaces, potentially offering a more user-friendly design environment that enhances system usability, especially for beginners. From this perspective, BIGcad is expected to become a benchmark for novel modeling interfaces.

4. 2. 4. Applicability of the System.

Users proposed two main aspects of utilization for the system: educational enhancement and support in the design process. Firstly, for education, the system's alternative workflows and command explanations aim to help beginners grasp modeling sequences and learn commands. It also promises to introduce creative workflows and unfamiliar commands to those proficient in Rhino. For example, P1 found the 'ChamferEdge' command—previously not frequently used—beneficial, simplifying an otherwise complex modeling process.

Moreover, the system's role as a reference tool for design development was highlighted. P1, P2, and P4 responded that the system could contribute to design development by providing new design ideas through recommended nodes and expected outcomes. Specifically, P1 responded that it could also serve to show examples of creative outcomes when developing designs during modeling. P4 reviewed the processes of both task 1 and task 3 while performing task 1 and showed a preference for the leg structure suggested in task 3. Therefore, P4 mentioned that if they were in the chair design process, they would have deviated from the initial design of task 1 and changed to a design with more curves and chamfered edges, influenced by task 3 (Figure 11).

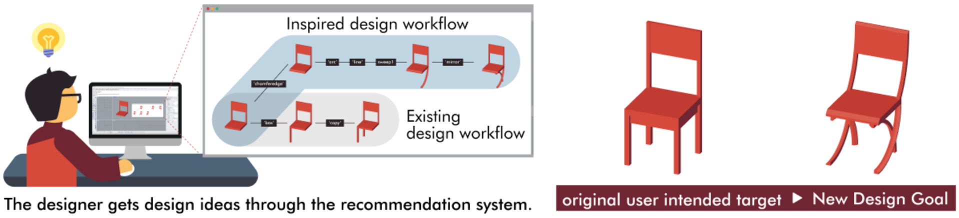


Figure 13 Design Evolution: Task 3 inspired changes in chair design, showcasing creative evolution

5. Limitations and Future Work

In this section, we identify three limitations of the study as follows.

Data Collection Limitation: The modeling sequences provided by BIGcad are based on actual modeling data collected from users. Consequently, the system faces limitations in offering recommendations when specific 3D modeling data is lacking.

Moreover, because the current dataset primarily reflects the modeling habits and preferences of a specific user group (e.g., design students), there is a potential for bias. This may lead the recommendation system to favor particular modeling patterns or commands, which could constrain its applicability across more diverse user populations.

Although collecting 3D modeling data is crucial for generating the W-graph and identifying meaningful workflows, it requires considerable effort. However, the challenge lies not in encoding the data into a W-graph but in the time and effort required to collect this data due to the lack of previously recorded modeling data (e.g., 3D model snapshots, commands). The experiments conducted in this study have confirmed the usability and accuracy of the system's recommendations. As more data is collected, the system's contributions and accuracy will become even more evident, thereby allowing for continuous improvement of the system. Future work aims to expand the dataset by incorporating a more diverse range of users and modeling contexts, thereby improving the system's generalizability and reliability over time.

Enhancing Personalized Recommendations: Users mentioned the need for additional information for the system to more quickly identify modeling goals in the early stages. P3 suggested that receiving workpiece references from designers before supporting the modeling process could recommend modeling steps for similar designs. P4 proposed the possibility of entering information about the modeling goal in advance to more clearly understand the user's modeling path from the beginning.

Addition of a 'Generate' Function: The current system is limited in effectively simplifying the modeling steps. While users can identify alternative and effective modeling sequences through the W-graph, they cannot directly reach the desired step and reduce the number of steps like with BIGFile. In the actual experiment, users requested a generate function to reach the target step in one go. If such a feature were implemented, it could significantly reduce the number of steps executed in the modeling process.

6. Conclusions

The BIGcad system has demonstrated a significant capacity to enhance the 3D CAD modeling process through the utilization of a Bayesian inference model combined with a workflow graph. This system effectively predicts modeling goals based on user behavior, offering tailored command recommendations and modeling sequences. The outcome is a more precise and efficient modeling process, tailored to the specific needs and intentions of the user. This innovation represents a substantial leap forward in supporting complex 3D modeling tasks, especially in applications requiring high precision and detail, such as architectural and product design.

From our user studies, participants reported a notable reduction in mental load and modeling time, along with improvements in modeling accuracy. This feedback underscores BIGcad's potential not only to streamline the design process but also to elevate the quality of the final models. Capitalizing on these advantages, BIGcad debuts a novel 3D modeling interface that fosters a more user-friendly and efficient modeling experience for everyone from novices to experts. Additionally, the system's potential to inspire design ideas and support design development was highlighted, reflecting its value as both a practical tool and a source of creative stimulation.

The educational potential of BIGcad also emerged as a significant benefit, with the system's ability to introduce learners to efficient modeling sequences and commands.

To enhance the system for showcasing a broader range of modeling processes in the initial stages, it's crucial to develop more precise methods for inferring user intentions and knowledge graph for 3D modeling parts. While BIGcad is currently implemented as a plugin for Rhino, its pipeline has the potential to be extended to other CAD platforms that support command tracking and model state capture. Future work may explore the integration of BIGcad with generative AI-based design systems to construct a hybrid modeling environment that supports both exploratory ideation and precise editing. For instance, systems such as SPAGHETTI or SALAD—capable of generating draft models via text-based 3D generation or shape interpolation—can be used to create initial designs, which are then imported into a CAD environment. BIGcad can subsequently provide command-level recommendations to refine these generative outputs into manufacturable models. Such integration bridges generative ideation with actionable design workflows, enabling designers to reach their intended outcomes more efficiently and accurately. This hybrid approach has the potential to offer design guidance for novice users while enhancing productivity for experts, ultimately contributing to a more intuitive, creative, and comprehensive modeling experience.

References

1. Alcaide-Marzal, J., Diego-Mas, J. A., & Acosta-Zazueta, G. (2020). A 3D shape generative method for aesthetic product design. *Design studies, 66*, 144-176.
2. Chang, M., Lafreniere, B., Kim, J., Fitzmaurice, G., & Grossman, T. (2020). Workflow graphs: A computational model of collective task strategies for 3D design software. In *Graphics Interface 2020*.
3. Chaudhuri, S., Kalogerakis, E., Giguere, S., & Funkhouser, T. (2013, October). Attribit: content creation with semantic attributes. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (pp. 193-202).
4. DiPrete, B. C., Garimella, R. V., Cardona, C. G., & Ray, N. (2023). Reinforcement learning for block decomposition of CAD models. *arXiv preprint arXiv:2302.11066*.
5. Deng, Y., Mueller, M., & Shields, M. (2023, August). CAD Challenges App: An Informatics Framework for Parametric Modeling Practice and Research Data Collection in Computer-Aided Design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 87295, p. V002T02A046). American Society of Mechanical Engineers.
6. Elsweiler, D., Harvey, M., & Hacker, M. (2011, July). Understanding re-finding behavior in naturalistic email interaction logs. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 35-44).
7. Faruqi, F., Katary, A., Hasic, T., Abdel-Rahman, A., Rahman, N., Tejedor, L., ... & Mueller, S. (2023, October). Style2Fab: Functionality-Aware Segmentation for Fabricating Personalized 3D Models with Generative AI. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (pp. 1-13).
8. Fitchett, S., & Cockburn, A. (2015). An empirical characterisation of file retrieval. *International Journal of Human-Computer Studies, 74*, 1-13.
9. Hertz, A., Perel, O., Giryes, R., Sorkine-Hornung, O., & Cohen-Or, D. (2022). Spaghetti: Editing implicit shapes through part aware generation. *ACM Transactions on Graphics (TOG), 41*(4), 1-20.
10. Hofmann, M., Hann, G., Hudson, S. E., & Mankoff, J. (2018, April). Greater than the sum of its PARTs: Expressing and reusing design intent in 3D models. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1-12).
11. Hui, K. H., Li, R., Hu, J., & Fu, C. W. (2022, November). Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia 2022 Conference Papers* (pp. 1-9).
12. Jang, Y., & Hyun, K. H. (2024, May). Advancing 3D CAD with Workflow Graph-Driven Bayesian Command Inferences. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (pp. 1-6).
13. Koo, J., Yoo, S., Nguyen, M. H., & Sung, M. (2023). Salad: Part-level latent diffusion for 3d shape generation and manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 14441-14451).
14. Lee, S. W., Kim, H., Yi, T., & Hyun, K. H. (2023). BIGaze: An eye-gaze action-guided Bayesian information gain framework for information exploration. *Advanced Engineering Informatics, 58*, 102159.
15. Liu, V., Vermeulen, J., Fitzmaurice, G., & Matejka, J. (2023, July). 3DALL-E: Integrating text-to-image AI in 3D design workflows. In *Proceedings of the 2023 ACM designing interactive systems conference* (pp. 1955-1977).
16. Liu, W., d'Oliveira, R. L., Beaudouin-Lafon, M., & Rioul, O. (2017, May). Bignav: Bayesian information gain for guiding multiscale navigation. In *Proceedings of the 2017 CHI conference on human factors in computing systems* (pp. 5869-5880).
17. Liu, W., Rioul, O., Mcgrenere, J., Mackay, W. E., & Beaudouin-Lafon, M. (2018, April). BIGFile: Bayesian information gain for fast file retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1-13).
18. Mandelli, L., & Berretti, S. (2022). CAD 3D Model classification by Graph Neural Networks: A new approach based on STEP format. *arXiv preprint arXiv:2210.16815*.

19. Quan, Y., Zhao, H., Yi, J., & Chen, Y. (2024). Self-supervised Graph Neural Network for Mechanical CAD Retrieval. *arXiv preprint arXiv:2406.08863*.
20. Schulz, A., Xu, J., Zhu, B., Zheng, C., Grinspun, E., & Matusik, W. (2017). Interactive design space exploration and optimization for CAD models. *ACM Transactions on Graphics (TOG), 36*(4), 1-14.
21. Son, K., Kim, K., & Hyun, K. H. (2022, April). BIGexplore: Bayesian information gain framework for information exploration. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (pp. 1-16).
22. Son, K., Lee, S. W., Yoon, W., & Hyun, K. H. (2022). CreativeSearch: Proactive design exploration system with Bayesian information gain and information entropy. *Automation in Construction, 142*, 104502.
23. Veuskens, T., Heller, F., & Ramakers, R. (2021). CODA: A Design Assistant to Facilitate Specifying Constraints and Parametric Behavior in CAD Models.